Survey: Six Years of Neural Differential Cryptanalysis

David Gerault[1], Anna Hambitzer[2], Moritz Huppert[3] and Stjepan Picek[4]

[1] Luxembourg Institute of Science and Technology, Luxembourg
[2] Technology Innovation Institute, UAE
[3] Technical University of Darmstadt, Germany
[4] Radboud University, The Netherlands

Abstract. At CRYPTO 2019, A. Gohr introduced Neural Differential Cryptanalysis and used deep learning to improve the state of the art in cryptanalysis of 11-round SPECK32. As of February 2025, Gohr's article has been cited 238 times on Google Scholar. The variety of targeted cryptographic primitives, techniques, settings, and evaluation methodologies that appear in these follow-up works provides a basis for a careful survey, which we provide in this paper. More specifically, we propose a taxonomy of these 238 publications and systematically review the 71 papers focusing on neural differential distinguishers, pointing out promising directions and recent advances in explainability and key-recovery attacks. We then highlight future challenges in the field, particularly the need for improved comparability of neural distinguishers and advancements in scaling. This holistic survey helps researchers and engineers to identify the leading neural differential attacks, compare their performance, and highlight the outstanding open problems in AI-assisted cryptanalysis.

Keywords: Neural Differential Cryptanalysis · Survey · Key Recovery

1 Introduction

The security of most digital applications relies on cryptography, the science of protecting the integrity, authenticity, and confidentiality of data, among other security goals. Confidentiality ensures that only the intended parties can read the exchanged data. Typically, an encryption key is used with a secure cipher algorithm to transform plaintext into ciphertext. The recipient, who knows the decryption key, can easily recover the plaintext from the ciphertext. In contrast, it is computationally infeasible for an adversary who does not know the key to do so.

The cornerstone of symmetric cryptography, in which the encryption and decryption keys are identical, is the block cipher. A block cipher encrypts fixed-size inputs, typically by applying a simple round function iteratively. Beyond their central role in ensuring confidentiality, block ciphers also serve as fundamental building blocks for other cryptographic primitives [KL07], such as hash functions and message authentication code (MAC) schemes. Therefore, the security analysis of block ciphers (or cryptanalysis) is a crucially important field.

In the classical notion of pseudorandom permutation (PRP) security, an adversary is given black-box access to an oracle implementing either (A) the block cipher under study with a hidden random key or (B) a random permutation. The block cipher is considered

E-mail: dagerault@gmail.com (David Gerault), anna.hambitzer@tii.ae (Anna Hambitzer), moritz.huppert@tu-darmstadt.de (Moritz Huppert), stjepan.picek@ru.nl (Stjepan Picek)

This work is licensed under a "CC BY 4.0" license.

secure if no adversary can distinguish between these two cases significantly faster than by exhaustive key search. The quality of such a distinguisher is typically measured by its time, data, and memory complexity.

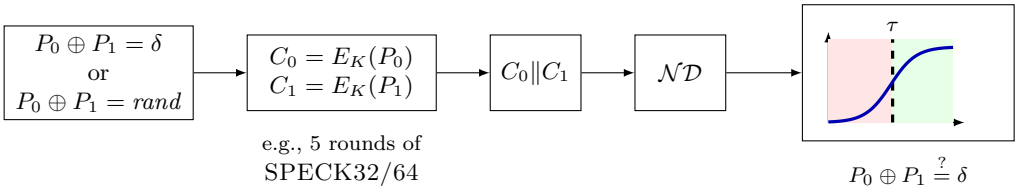


Figure 1: Neural Differential Distinguisher: Basic Training Pipeline. Given plaintexts P_0, P_1 with $P_0 \oplus P_1 \in \{\delta, \text{rand}\}$, encrypt under key K to obtain $C_i = E_K(P_i)$. The concatenation $C_0 \| C_1$ is fed to $\mathcal{ND}$, which outputs a sigmoid score $\hat{Y} \in [0, 1]$; thresholding at τ yields a binary decision for $P_0 \oplus P_1 = \delta$ (yes if $\hat{Y} \geq \tau$). In the plot, the constant baseline for the negative class is shown in red and the sigmoid for the positive class in green; the background halves visualize the threshold.

The main goal of block-cipher cryptanalysis is to estimate how many applications of the round function, or rounds, are required to achieve security, that is, until such a distinction becomes infeasible. This is an iterative process, and new cryptanalytic results often continue to appear many years after a cipher's release. As a result, cryptographers are keen to develop and refine tools that can assist with this tedious task.

Differential cryptanalysis, introduced by Biham and Shamir in 1991 [BS90], studies how input differences (δ) propagate through a block cipher, seeking high-probability differentials for which specific plaintext differences yield predictable ciphertext differences (Δ). Traditional approaches rely on mathematical analysis of the propagation of differences across rounds, evaluating the probabilities that given input differences produce specific output differences to build distinguishers. Although this technique proved highly effective against early ciphers, modern designs include dedicated countermeasures that make conventional differential analysis increasingly difficult and computationally expensive. In particular, it often requires substantial manual analysis of the cipher structure and may still yield suboptimal distinguishers for complex round functions. Owing to their ability to detect and exploit subtle patterns, deep learning methods have long been regarded as a promising tool for cryptanalysis.

Deep learning has made significant advances in recent years, leading to remarkable achievements across various domains. Initially, Frank Rosenblatt introduced Multi-Layer Perceptrons (MLPs) in his 1958 book Perceptron and laid the foundation for modern neural networks. The introduction of Convolutional Neural Networks (CNNs) in the 1980s [Fuk80] led to a breakthrough in computer vision in the form of LeNet, which achieved human-level performance in digit recognition in 1998 [LBBH98]. Through advancements in Monte Carlo Tree Search (MCTS) and reinforcement learning, further leaps were enabled, such as Google's AlphaGo surpassing human capabilities [SHM+16, SHS+18, SAH+20]. More recently, transformer-based Large Language Models (LLMs) [VSP+17], such as GPT, have revolutionized natural language processing, demonstrating near-human capabilities in tasks like machine translation and language generation.

Despite the long-standing recognition of the intersection between cryptography and machine learning [Wea47, Val84, Riv91], the use of machine learning in cryptanalytic tasks has remained limited. Early approaches typically relied on extensive precomputation [PPS14], exploited implementation flaws (e.g., side-channel attacks) [HGG20], targeted inherently weak cryptographic schemes [Gre17, GHZ+18], or generally proved ineffective [CLC12]. It was not until Gohr's work [Goh19], presented at CRYPTO 2019, that deep learning was successfully combined with traditional cryptanalytic techniques, leading to improved

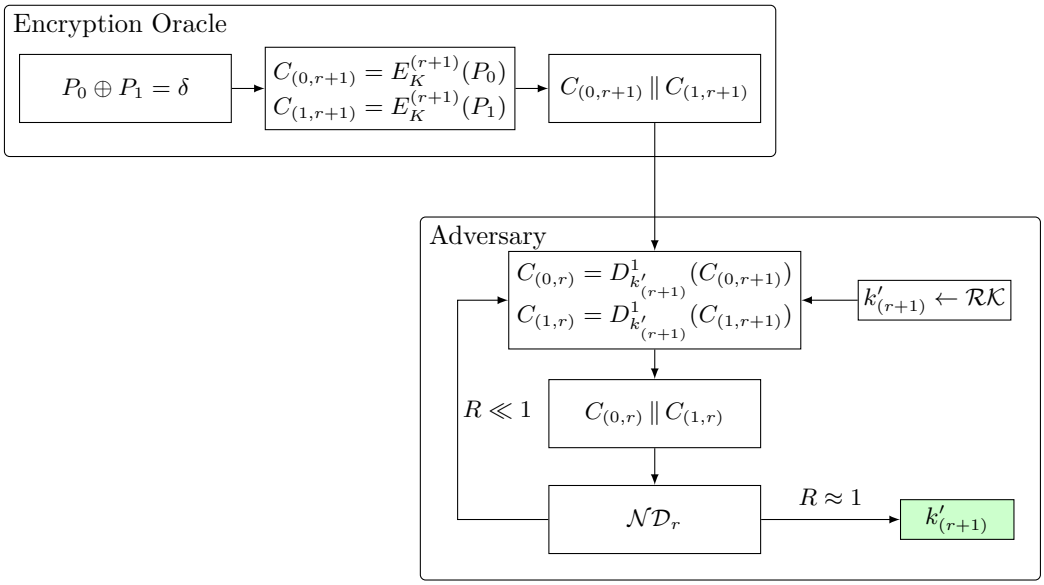


Figure 2: Neural Differential Distinguisher: Basic Key Recovery Pipeline. Let $\mathcal{RK}$ denote the round-key space of the block cipher under scrutiny. An encryption oracle (with a secret master key K) generates an $(r+1)$-round ciphertext pair with input difference δ. The adversary decrypts round $(r+1)$ using a key guess $k'_{(r+1)}$ and feeds the result to a neural distinguisher $\mathcal{ND}_r$, which has been trained on the r-round ciphertext pair distribution. Key guesses $k'_{(r+1)}$ producing distinguisher scores near 1 are output as candidates, following the wrong-key randomization hypothesis [Goh19].

attacks. In particular, Gohr showed that neural networks can be effectively applied to cryptanalysis and achieved results that improved on previous methods for a round-reduced variant of a modern block cipher.

Gohr's work [Goh19] marked an important step in this line of research by showing that deep neural networks can be used as distinguishers in what is now commonly called neural differential cryptanalysis.[1] These networks are trained to detect statistical differences between ciphertext pairs arising from structured input differences and those arising from random differences, as illustrated in Figure 1. In particular, for the NSA-designed SPECK32 [BTCS+15] cipher, Gohr showed that neural distinguishers achieve strong performance on variants of up to 8 rounds and, once trained, can be incorporated into practical key-recovery attacks that improve on the state of the art at the time for variants of up to 11 rounds, as illustrated in Figure 2. For completeness, we note that Gohr's key-recovery attack was later surpassed in time complexity by a subsequent classical attack [BdST+23].

Since Gohr's seminal paper, researchers have explored every part of the basic Neural Differential Cryptanalysis pipeline. A majority of the works citing Gohr that focus on neural differential distinguishers have attempted to apply the scheme to **24 different symmetric-key primitives** (with the SPECK32 and SIMON32 variants receiving the most attention at 32 and 17 experiments respectively), improve the distinguishing advantage by **developing 15+ distinct network architectures** (from the dominant $\mathcal{ND}_{\text{Gohr}}$ to specialized approaches like DBitNet, SE-ResNet, and even quantum neural

[1] We use the terms machine-learning-aided cryptanalysis and neural cryptanalysis interchangeably. Although non-neural machine-learning methods are sometimes used to train distinguishers, they are typically less successful, as discussed in our comparative review (cf. Section 8.1). We adopt this terminology because neural methods are by far the most common approach in this area.

networks), **scaling training data from originally 20 million to 32 billion samples** (though 20M remains the de facto standard), or **innovating sample formats** (including advanced feature engineering techniques that have been shown to improve distinguishing accuracy). Finally, we also observe emerging research directions aimed at enhancing neural distinguishers across multiple dimensions: increasing **automation** of the attack pipeline (as indicated by AutoND flags in a total of 35 experiments), improving **transparency** through explainability techniques, and boosting **effectiveness** of the obtained neural distinguishers in practical **key recovery** attacks.

The intersection of machine learning and cryptography has grown rapidly in recent years, as evidenced by the increasing presence of ML-related work at leading cryptography conferences. This development includes both **cryptography for machine learning** [LSE+26, QLW+26] and **machine learning for cryptography** [GHRS26], and is particularly visible in cryptanalytic research that employs learning-based methods [HBLC25, SMR+24, YWZW26, ZYS+26]. Given the pace of progress, it is plausible that the full potential of machine learning for cryptanalysis has yet to be realized. At the same time, the rapid growth of the literature on Neural Differential Cryptanalysis has led to a fragmented research landscape, making systematic synthesis increasingly difficult. Recent developments have outpaced efforts to establish coherent theoretical foundations and consistent methodological standards, posing particular challenges for researchers operating at the intersection of cryptography and machine learning. As a consequence, the field has exhibited several recurring issues, including duplicate lines of inquiry accompanied by erroneous novelty claims, such as the erroneous assertion by [Seo24] that it introduced the first truncated neural distinguisher, unresolved methodological disagreements, for example the conflicting conclusions of [BR21], [SSL+22], and [BBCD21] on the suitability of different neural architectures, and persistent misunderstandings of foundational concepts. A notable example is the aggregation of predictions across multiple ciphertext pairs, an issue explicitly clarified by Gohr in [GLN22] but still misinterpreted in subsequent work. Lastly, the literature exhibits a concerning pattern of unreliable result reporting, including claims of full-round attacks on modern block ciphers and evaluations based on validation datasets that are too small to support robust conclusions. Such experimental setups can enable cherry-picking of unusually high-performing runs and may even lead to random-guessing distinguishers being reported as having a meaningful advantage over random guessing. This makes it considerably more difficult to distinguish robust, reproducible findings from overstated or methodologically fragile results.

These issues highlight the need for a comprehensive survey and critical analysis of the literature on neural (differential) cryptanalysis. In addition to consolidating the existing body of work, such a survey can clarify foundational concepts, contextualize competing methodological perspectives, and make the field more accessible to researchers from both cryptography and machine learning. As the area continues to expand, developing a coherent theoretical and methodological framework for Neural Differential Cryptanalysis becomes increasingly necessary.

Our Contributions. In our survey, we have achieved the following:

1. **Accessible Core Concepts:** In the preliminaries (Section 4), we present the *fundamental concepts* of Neural Differential Cryptanalysis in a simple, precise, and expressive manner.
2. **Comprehensive Field Review:** We conduct an exhaustive survey of the literature building on Gohr's seminal work [Goh19] (Section 5). Based on this survey, we identify the *full body of research* in Neural Differential Cryptanalysis, characterize its *principal research directions*, develop a *detailed taxonomy* of the field, and compile an *exhaustive list of open-source frameworks* accompanying the peer-reviewed literature.
3. **Explainability and Key-Recovery Overview:** We *survey recent advances* in

explainability techniques for neural differential distinguishers (Section 6) and in *neural-aided key recovery* (Section 7). While these sections are not the central contribution of this work, they represent two major research directions identified in our field review. Their inclusion is therefore intended to provide a comprehensive view of Neural Differential Cryptanalysis and to contextualize our main contributions within the broader literature.

4. **Rigorous Classification and Comparison:** We provide a *systematic classification and comparison of peer-reviewed research* on neural differential distinguishers (Section 8) across diverse techniques, architectures, and cryptographic primitives. On this basis, we identify promising directions for future research, expose substantial methodological weaknesses in parts of the literature, and critically examine the validity of selected published results.
5. **Best-Practice Recommendations:** Because evaluating research on neural network training is challenging, we propose a *comprehensive set of best-practice guidelines* tailored (but not limited) to the assessment of Neural Differential Cryptanalysis research (Section 9).
6. **Future Challenges:** Finally, we identify and discuss *three major challenges* set to shape the next six years of neural cryptanalysis (Section 10).

We emphasize that one of the aims of this survey is to serve as a central reference point for situating and comparing future work within the field. For this reason, we explicitly reference all relevant works throughout the paper and include dedicated overview boxes for individual categories. These are intended to help readers quickly identify, access, and navigate the subset of papers most relevant to a particular topic, without having to extract them manually from the broader discussion.

2 Related Work

Related Surveys While a substantial body of literature has been devoted to the development and analysis of effective neural differential distinguishers, this paper is, to the best of our knowledge, the first to systematically organize, compare, and critically examine a large body of work (238 papers), while also identifying promising research directions and key challenges in the area. Recent related surveys [AEA17, BHR+22, NR23, MLR+24, SST24, TDD22] do not claim a systematic approach, cover a significantly smaller body of work, and most lack a specific focus on machine learning-based cryptanalysis.

Bellini et al. in [BHR+22] examine machine learning-based black-box and white-box cryptanalysis. Regarding white-box cryptanalysis, they reference Gohr's work [Goh19] along with 15 related follow-up studies, though these were not selected systematically and include preprints. Similarly, Tolba et al. [TDD22] reference Gohr's work and a selection of subsequent studies, but their discussion is primarily oriented toward the potential of quantum neural networks. Nitaj and Richidi, in [NR23], explore various cryptographic areas that could benefit from the application of artificial intelligence (AI). While they briefly mention the potential of machine learning to enhance side-channel and cryptanalytic attacks on symmetric block ciphers, they neither provide a systematic analysis of existing work in this area nor delve into the specific methodologies involved. Awad and El-Alfy, in [AEA17], conduct a survey of computational intelligence applications in cryptography, focusing on the automated design and cryptanalysis of ciphers. However, their work predates Gohr's introduction of differential machine learning-based cryptanalysis in [Goh19], and as a result, it does not include a comprehensive review of neural distinguishers found in more recent literature. Singh et al. [SST24] investigate various machine learning and optimization techniques, including Hill Climbing and Particle Swarm Optimization, applied to cryptanalysis. They also reference Gohr's research [Goh19] along with 12 subsequent studies that build upon it. However, the selection of these papers is not based on a

systematic methodology. The work by Martinez et al. [MLR+24] is the most comparable to ours. Although it does not follow a systematic paper selection process, it aims to capture the state of the art and categorizes 10 works, including Gohr's, based on their architectures and the cryptographic schemes they target.

Neural Cryptanalysis of Historical Ciphers A related, but largely distinct, line of research has emerged within the *HistoCrypt* community. In contrast to Gohr's work and most of the subsequent literature, which focus primarily on neural differential distinguishers and neural-aided key recovery for modern symmetric primitives, these works typically apply machine learning to problems in historical cryptanalysis, i.e., the automation, classification, and analysis of ciphers that are already broken by classical cryptanalytic methods. Examples include ciphertext-only cipher classification [DS25, Kop20, LKE+21] and attacks on historical cipher systems [Gre17].

This line of work is important not only because historical cipher systems are valuable cryptanalytic objects of study in their own right, but also because it may eventually produce insights that extend beyond historical cryptology. As in the broader history of cryptanalysis, methods developed and tested on simpler or more structured systems can provide useful foundations for approaches to more complex modern primitives. At the same time, the corresponding research landscape is currently methodologically and empirically distinct from the one examined in this survey; accordingly, we do not include these works in our comparative analysis.

3 AI and Cryptography in the Beginnings

The popularity and widespread adoption of neural differential distinguishers (more precisely, deep learning-based cryptanalysis) can be credited to the seminal work of A. Gohr [Goh19]. However, even in that work, the author cited several related works at the intersection of cryptanalysis and AI. What distinguishes Gohr's work from previous ones is that it considers relevant (modern) ciphers and manages to obtain results that surpass state-of-the-art conventional cryptanalysis techniques. The following section is not meant to provide an exhaustive list of works connecting AI and cryptology, but rather to provide a brief historical overview of various approaches.

Already in 1947, researchers started considering connections between cryptography and artificial intelligence [Wea47]. While this attempt lacked technical details, it still demonstrates the scientific community's interest in combining these two domains. In 1984, L. Valiant discussed learnable Boolean functions and cited cryptographic evidence that the whole class of functions computable by polynomial-size circuits is not learnable [Val84]. Shortly after, in 1988, Minsky and Papert showed that every Boolean function can be realized by an MLP neural network [MP87]. In 1991, R. Rivest wrote a paper on connections between cryptography and machine learning [Riv91]. Already there, he mentioned the possibility of using machine learning for cryptanalysis.

In 2002, Klimov et al. analyzed the security of a key exchange protocol based on mutually learning neural networks [KMS02]. While the authors experimentally verified that it is unlikely for a particular attacker using a similar neural network to converge to the same key, they still break the protocol using more advanced cryptanalytic techniques. Similarly, in 2016, Abadi and Andersen [AA16] employed neural networks in a framework inspired by generative adversarial networks (GANs) to develop an encryption scheme. Although this early research did not show any formal security, Coutinho et al. demonstrated in 2018 [CdOAB+18] that, with certain architectural modifications, the network could be trained to learn the One-Time Pad [Sha49].

In 2002, Castro et al. used evolutionary algorithms to construct a cryptanalytic tool that can distinguish between the two-round TEA algorithm and random permutations [CSIR02].

In 2007, Laskari et al. considered the application of diverse computational intelligence techniques to the cryptanalysis of known cryptosystems, including public key cryptosystems and Feistel ciphers [LMSV07]. In the same year, Tapiador et al. used heuristics for nonlinear cryptanalysis and applied them to the MARS S-box [TCHC07]. In 2012, Chou et al. experimented with machine learning techniques to mount distinguishing attacks and concluded it is not possible to extract useful information from ciphertexts produced by modern ciphers operating in secure modes, nor to distinguish them from random data [CLC12]. On the other hand, Svenda et al. in 2014 used evolutionary algorithms to construct empirical tests for randomness [SSUM14].

4 Preliminaries

This section introduces key concepts in machine learning-assisted differential cryptanalysis: conventional differential cryptanalysis (Subsection 4.1), deep learning applications (Subsection 4.2), and neural network-aided key recovery (Subsection 4.3).

4.1 Differential Cryptanalysis

Differential cryptanalysis [BS90] is a chosen-plaintext attack analyzing how plaintext perturbations propagate through block ciphers. In principle, it can be applied to any symmetric primitive with a deterministic internal structure. In practice, it has proven most effective for block ciphers and permutation-based designs. While bitwise XOR is typically used, some approaches employ modular addition or rotations. For a map $F : \{0,1\}^b \to \{0,1\}^b$ a *differential transition* is a pair $(\delta, \Delta) \in \{0,1\}^b \times \{0,1\}^b$ with probability:

$$P(\delta \to \Delta) = \frac{\left|(\{x \in \{0,1\}^b : F(x) \oplus F(x \oplus \delta) = \Delta\})\right|}{2^b}.$$

In the differential cryptanalysis of block ciphers, F denotes the (possibly round-reduced) encryption function $E_K : \{0,1\}^b \to \{0,1\}^b$ instantiated with a fixed but unknown key K. The analysis typically considers the expected differential probability $\mathbb{E}_K[P_K(\delta \to \Delta)]$ over uniformly random keys, reflecting the average behavior of the cipher family.

4.2 Training Neural Differential Distinguishers

Let $P_0, P_1 \in \{0,1\}^b$ with associated ciphertexts $C_i = E_K(P_i)$. For a fixed difference $\delta \in \{0,1\}^b$ and over all possible master keys $K \leftarrow \mathcal{K}$, a neural distinguisher $\mathcal{ND}$ is a learned classifier

$$\mathcal{ND} : \{0,1\}^{2b} \to \{0,1\}, \qquad (C_0 \| C_1) \mapsto \hat{Y},$$

trained to approximate the decision rule

$$Y(C_0 \| C_1) = \begin{cases} 1, & \text{if } P_0 \oplus P_1 = \delta, \\ 0, & \text{otherwise.} \end{cases}$$

For high prediction accuracy, $\mathcal{ND}$ exploits non-uniformities in the ciphertext-pair distribution induced by the input difference δ across the keyspace $\mathcal{K}$.

Training is supervised, using balanced datasets with equal proportions of positive and negative examples. Positive samples $(C_0, C_1, 1)$ correspond to conforming plaintext pairs satisfying $P_1 = P_0 \oplus \delta$, while negative samples $(C_0, C_1, 0)$ are generated from random independent plaintexts. Neural networks are optimized using stochastic gradient descent (SGD) [RM51], which updates parameters based on randomly sampled *batches* rather than the full dataset. An *epoch* denotes one complete pass over the training data. The loss

function (typically mean squared error or cross-entropy) quantifies the discrepancy between the predicted outputs $\hat{Y}$ and the true labels Y.

Typical training regimes, following Gohr [Goh19], use approximately 10^7 training samples, 10^6 test samples, batch sizes of about 5,000, and up to 200 epochs. Performance and convergence are enhanced by optimization techniques such as the Adam optimizer [KB15], which adaptively tunes learning rates per parameter, and by regularization methods like L2 weight decay [HK00] or batch normalization. Learning-rate scheduling and data shuffling further stabilize convergence and improve generalization.

The choice of the neural architecture strongly influences the distinguisher's ability to capture cipher-specific structure and feature dependencies. Common design elements include *multilayer perceptrons (MLPs)*, which consist of fully connected layers offering universal approximation capabilities at the cost of large parameter counts and limited inductive bias; *convolutional layers (CNNs)*, which apply local filters with parameter sharing to detect spatial or word-level differential patterns; and *Inception modules*, which combine parallel convolutions with multiple kernel sizes to extract features at different spatial scales. Deeper networks often incorporate *residual connections (ResNets)* to alleviate gradient vanishing and facilitate the training of deep architectures. When cipher rounds are modeled as sequential transformations, *recurrent units* such as LSTMs can capture temporal dependencies across rounds. More recently, *attention mechanisms* enable dynamic, data-dependent weighting of input components, allowing the model to learn long-range dependencies. Architectural design is typically cipher-specific: for ARX ciphers, convolutional or masked layers exploit bit- or word-level locality, whereas more generic architectures such as deep ResNets or transformers are used to capture higher-order dependencies at increased sample complexity and computational cost [GLN22, BGHR23].

4.3 Neural-aided Key Recovery

Once a neural distinguisher $\mathcal{ND}_r$ that sufficiently approximates $Y(C_{(0,r)}||C_{(1,r)})$[2] has been obtained, inference enables practical key recovery attacks on block ciphers, demonstrating their concrete cryptanalytic value. This section outlines the attack methodology based on Gohr's seminal approach [Goh19], which has become the standard framework in subsequent research.

The Basic Attack Let $\mathcal{ND}_r$ be a neural distinguisher trained to approximate the binary decision

$$Y(C_{(0,r)}||C_{(1,r)}) = \begin{cases} 1, & \text{if } P_0 \oplus P_1 = \delta, \\ 0, & \text{otherwise,} \end{cases}$$

where $C_{i,r} = E_K(P_i)$ is the r-round encryption of plaintext P_i, $\delta \in \{0,1\}^b$ is a fixed input difference, and training is performed over pairs (P_0, P_1) (either satisfying $P_0 \oplus P_1 = \delta$ or sampled at random) and master keys $K \leftarrow \mathcal{K}$. Here E denotes the r-round reduced encryption function of the block cipher, and $\mathcal{ND}_r$ outputs a score $R \in [0,1]$ estimating the likelihood that the input pair has difference δ.

Assume the cipher is round-based with round keys $k_1, \ldots, k_{r+1}$ derived from the master key K. We use $\mathcal{ND}_r$ to recover the last round key k_{r+1}. The attack proceeds as follows: The adversary queries the $(r+1)$-round encryption oracle with a conforming plaintext pair $(P_0, P_1 = P_0 \oplus \delta)$ and obtains the ciphertext pair $(C_{(0,r+1)}, C_{(1,r+1)})$. For a candidate last-round key k'_{r+1} (or trial key), the adversary computes the one-round decryptions

$$C_{(i,r)} \;=\; D^1_{k'_{r+1}}(C_{(i,r+1)}) \qquad (i = 0, 1),$$

[2]Here, r denotes the r-round reduced version of the block cipher under consideration, for example $r = 7$-round SPECK32 as compared to the full 22-round version.

where $D^1_{k'_{r+1}}$ denotes the 1-round block cipher decryption function with round key candidate k'_{r+1}. The candidate is then scored by the distinguisher: $R_{k'_{r+1}} = \mathcal{ND}_r(C_{(0,r)} \| C_{(1,r)})$. Under the wrong-key randomization hypothesis [Goh19], correct key guesses produce scores $R_{k'_{r+1}} \approx 1$ (the distribution matches the distinguisher's positive class), while incorrect guesses yield scores close to the random baseline. Keys are therefore ranked by $R_{k'_{r+1}}$, and the top candidates are tested or expanded in a subsequent recovery phase.

When round keys are short (for example, SPECK32 uses 16-bit round keys while the master key is 64 bits), it is feasible to enumerate all 2^{16} candidates k'_{r+1}. After identifying the last round key, the same procedure can be iterated to recover preceding round keys until the full key schedule is reconstructed.[3]

Prediction scores exhibit sampling variance, so practical attacks aggregate responses across multiple ciphertext pairs to increase reliability. Given m independent ciphertext pairs $C_{(0,r,0)} \| C_{(1,r,0)}, \ldots, C_{(0,r,m-1)} \| C_{(1,r,m-1)}$ and distinguisher responses $R^i_{k'_{r+1}} = \mathcal{ND}_r(C_{(0,r,i)} \| C_{(1,r,i)})$ for each candidate round key k'_{r+1} and pair $C_{(0,r,i)} \| C_{(1,r,i)}$, we form a log-likelihood aggregate score

$$A_{k'_{r+1}} = \sum_{i=0}^{m-1} \log\left(\frac{R^i_{k'_{r+1}}}{1 - R^i_{k'_{r+1}}}\right),$$

which improves separation between correct and incorrect keys and is used to rank candidates in practice.

Extending the Rounds Covered by the Distinguisher. Neural distinguishers can sometimes be applied beyond the number of rounds on which they were explicitly trained by exploiting structural properties of the target cipher. In ARX designs such as SPECK and SIMON, the first round key is added only after the initial nonlinear operation. This allows one to gain an additional round of coverage without retraining the distinguisher. Concretely, one chooses plaintext pairs with difference δ' such that, after one round of encryption, their difference becomes the input difference δ expected by the neural distinguisher. In the notation of differential cryptanalysis, this means selecting δ' so that the one-round differential transition $\delta' \to \delta$ holds with probability 1. The distinguisher can then be applied exactly as before, but effectively one round later in the cipher.

More generally, neural distinguishers can be combined with classical differential cryptanalysis by prepending an s-round differential transition to an r-round neural distinguisher, thereby obtaining an $(s + r)$-round distinguisher. Suppose the neural distinguisher is trained to distinguish ciphertext pairs arising from input difference δ after r rounds. One then chooses an additional difference δ' such that the s-round transition $\delta' \to \delta$ holds with probability $P(\delta' \to \delta)$. Plaintext pairs are generated with difference δ'; with probability $P(\delta' \to \delta)$, they reach the difference δ after the first s rounds and can therefore be processed by the neural distinguisher over the remaining r rounds. This hybrid construction extends the round coverage, but it also inherits the standard trade-off from differential cryptanalysis: if $P(\delta' \to \delta)$ is small, then many more plaintext pairs are required to obtain a sufficient number of useful samples. In Gohr's implementation [Goh19], for example, a two-round differential transition of probability 2^{-6} was prepended to a 7-round neural distinguisher, resulting in a 9-round distinguisher at the cost of increasing the data complexity by roughly a factor of 64.

This combination of classical and neural components also raises an important scoring issue. Because the prepended differential transition $\delta' \to \delta$ holds only probabilistically, many plaintext pairs with input difference δ' will not arrive at difference δ after the first s rounds. These pairs, therefore, do not match the input distribution expected by the neural

[3] Despite not being able to recover the master key, this allows the adversary to encrypt plaintexts and decrypt ciphertexts, respectively

distinguisher and act as noise, significantly reducing the reliability of the resulting scores. One way to mitigate this problem is to use *(probabilistic) neutral bits* (PNBs). Informally, a neutral bit is a bit that can be flipped in both texts of a pair without destroying the relevant differential behavior, either with probability 1 or with high probability. More precisely, if a pair (P_0, P_1) satisfies a differential transition $\delta \to \Delta$, then a neutral bit i has the property that $(P_0 \oplus (1 \ll i), P_1 \oplus (1 \ll i))$ also satisfies $\delta \to \Delta$, either always or with high probability. Starting from one suitable pair, j, such bits generate a structure of size 2^j. The key advantage is that all pairs in such a structure tend to behave coherently: they either all follow the prepended differential transition or all fail together. This reduces inconsistency in the aggregated scores and improves the overall signal-to-noise ratio. Later work generalized this idea through *generalized neutral bits*, which support the chaining of adjacent differential transitions [BGL+22].

Reducing Computational Cost. Gohr [Goh19] proposes a statistical method based on Bayesian optimization that reduces the computational cost of exhaustive round-key search in the key-recovery attack by shifting part of the effort into a precomputation phase. Indeed, the author pre-computes a *Wrong-Key Response Profile* (WKRP) to guide key recovery. Let k_{r+1} be the true last-round key and $k'_{r+1} = k_{r+1} \oplus \gamma$ a trial key with difference γ. For a fixed key difference γ, let X_γ denote the random variable obtained by evaluating $\mathcal{ND}_r$ on a randomly sampled ciphertext pair $(C_{(0,r+1)}, C_{(1,r+1)})$ after one-round partial decryption under $k'_{r+1} = k_{r+1} \oplus \gamma$:

$$X_\gamma = \mathcal{ND}_r\Big(D^1_{k'_{r+1}}(C_{(0,r+1)}) \parallel D^1_{k'_{r+1}}(C_{(1,r+1)})\Big).$$

For each round key difference γ, the WKRP stores the empirical mean μ_γ and standard deviation σ_γ of the response X_γ.

In practice, responses are averaged over a ciphertext structure of many pairs. If a structure contains m pairs, then the average response under trial key k'_{r+1} is modeled as approximately Gaussian with mean μ_γ and standard deviation $\sigma_\gamma/\sqrt{m}$.

Given a set of trial keys $k^{(0)}_{r+1}, \ldots, k^{(\ell-1)}_{r+1}$, let $\overline{X}^{(0)}, \ldots, \overline{X}^{(\ell-1)}$ denote the corresponding average distinguisher responses over the ciphertext structure. Key recovery is then formulated as finding the true key k_{r+1} whose induced WKRP best matches the observations, namely by minimizing

$$\sum_{i=0}^{\ell-1} \frac{\left(\overline{X}^{(i)} - \mu_{k_{r+1} \oplus k^{(i)}_{r+1}}\right)^2}{\sigma^2_{k_{r+1} \oplus k^{(i)}_{r+1}}}.$$

Hence, the attack does not rely on a single high distinguisher score but instead matches the entire observed response pattern to the precomputed WKRP.

The search is initialized with a random subset of trial keys, which are evaluated on a ciphertext structure by averaging their distinguisher responses. These observations are then used to evaluate the WKRP objective for every possible true-key hypothesis, and the next ℓ trial keys are chosen as those ℓ hypotheses that minimize this objective. Repeating this procedure iteratively refines the search toward increasingly plausible keys.

When multiple ciphertext structures are available, Gohr allocates effort using an *Upper Confidence Bound* (UCB) rule:

$$s_i = w^{(i)}_{\max} + \alpha\sqrt{\frac{\log_2(j)}{n_i}},$$

where $w^{(i)}_{\max}$ is the best score obtained so far for structure i, n_i is the number of iterations spent on that structure, and j is the global iteration counter. This balances exploration of underused structures and exploitation of promising ones.

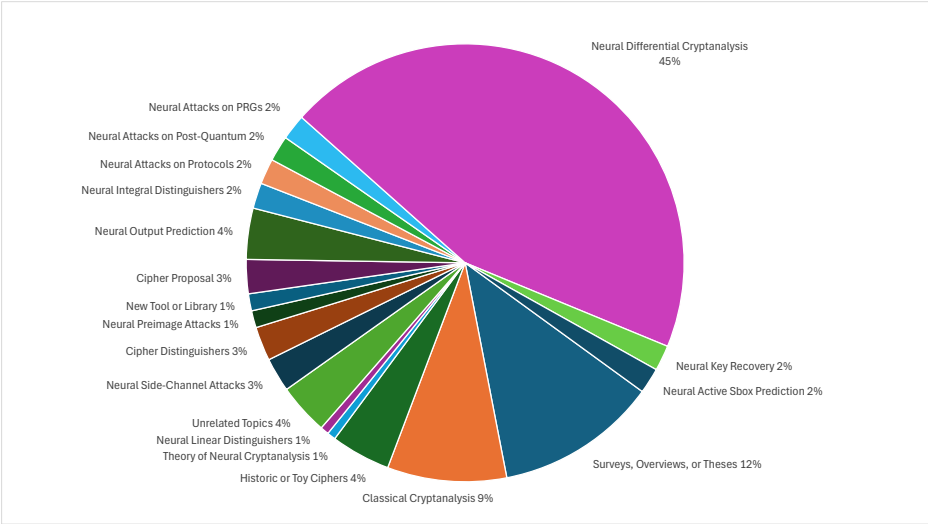


Figure 3: Our taxonomy of the peer-reviewed English-language publications that cite Gohr's seminal work [Goh19].

Additional Verification. Gohr initially observed that near-correct key candidates can also yield high distinguisher scores [Goh19]. In particular, candidates that differ from the true key in only a few bits may still produce elevated responses. Gohr therefore performs an additional local search within the Hamming neighborhood of promising candidates to distinguish the true key from nearby false positives.

A second and conceptually separate verification step is based on cascaded recovery across adjacent rounds. To this end, Gohr combines the distinguishers $\mathcal{ND}_r$ and $\mathcal{ND}_{r-1}$ in a joint recovery procedure for the $(r+1)$- and r-round keys. Whenever the $(r+1)$-round search guided by $\mathcal{ND}_r$ produces a candidate whose score exceeds a threshold t_1, an auxiliary r-round key search is initiated using $\mathcal{ND}_{r-1}$. The candidate is accepted only if the corresponding r-round score also exceeds a second threshold t_2. This cascaded verification filters false positives, since incorrect $(r+1)$-round key guesses rarely induce consistently high scores under the associated r-round distinguisher.

5 Neural Differential Cryptanalysis: A Taxonomy of Research Directions

5.1 Selected Literature

As of February 03, 2025, a total of 238 works cite Gohr's work [Goh19] on Google Scholar. Among these, we exclude 19 redundant entries, 2 faulty entries, 36 entries not available in English, and our own work. Additionally, 21 references have not been peer-reviewed [BBC+26, DNS24, Ess23, GJS20a, GLN22, Goh22, Hal22, HRC21a, HRC21b, HRC21d, JKM21, KJL+22, KVD+25, LRC22, PMC+22, PMK20, SS23, SWL+25, YW24, ZZW24, ZDW+23]. After excluding these, we are left with 159 peer-reviewed references, which we systematically categorize as shown in Figure 3.

We exclude references that are outside the field of Neural Differential Cryptanalysis. This includes works whose primary focus is one of the following:

- broad surveys, overviews, and theses on "ML in cryptography" [Bak21, BHR+22, Bru21, CDS22, ElS21, MJBH22, MLR+24, Moh23, NMN24, NR23, PJ21, PJ22, Ros24, Som23, Tan23, TDD22, Tu22, YLFP24, ZG24]

- classical cryptanalysis [BdST+23, DDK+23, ELR20, FLW+23, GPT21, ITY25, KLJW23, KY21, LJSC24b, PLH+24, SLL24, WW22, YK22, ZYSH24],
- theory of neural differential cryptanalysis [SMR+24],
- historic or toy ciphers [GZAL22, KSJS21, KLK+23, LF24, LMK+21, PKM23, So20],
- neural active S-box prediction [HLG+23, ITYY21, LTJ+21],
- neural side-channel attacks [GJS20b, WNB+26, YBBP23, ZZC+22],
- distinguishers between different ciphers [BP22, DM23, MPM+21, XLC+22],
- neural preimage attacks [JM24, LLL+21],
- neural key recovery attacks [CS21, DZF+22, ZLF+24],
- introduction of a new tool or library [BGG+23, PVM24],
- proposal of a new cipher [AVS+24, BSL21, DCW23, FAAQ24],
- neural output prediction attacks [KEI+22, KEI+23, MKMP21, PTD22, TDF+22, WIO24],
- neural integral distinguishers [HLLL24, WG24, ZL22],
- neural linear distinguishers [HLZW20],
- neural attacks on protocols [MLYW22, TD21, ZKL20],
- post-quantum schemes [AAEK23, LWAZ+23, WCCL22],
- pseudorandom number generators [Boa24a, Boa24b, EAZD23], and
- other unrelated topics [ABA22, KJK+24, MGMP21, PYW24, Sug24, ZZS21].

This leaves us with a total of 71 peer-reviewed publications in the field of Neural Differential Cryptanalysis.

The Body of Peer-Reviewed Research in Neural Differential Cryptanalysis

The full body of peer-reviewed publications that focus specifically on advancing research of Neural Differential Cryptanalysis is [BBDH21, BR21, BGPT21, BBCD21, BGL+22, BBP22, BBD+23, BLYZ23, BGHR23, BPC24a, BFG+24, BPC24b, CSY23, CSYY23, DCC23, ERP22, EGP23, GLP+24, HRC21c, HGH+23, HRC23, HLF+24, HLZH25, KS22, KJL+23, KKJ+24, Li24, LCLH22, LTZ22a, LTZ22b, LLHC23, LRC23, LRCL23, LLS+24, LRC24, LJSC24a, MPKM+22, PMDC22, PPWR23, PSM24, PCDC24, RRSM22, RKS23, SZM20, SSL+22, SM24, SL25, SSL+24, SBG+25, Seo24, TH21, TTJ23, TSL23, WW21, WWH21, WTZ+22, WQW+24, WW24b, WWS24b, WW24a, WWS24a, YK21, YW23, YK24, ZLHH25, ZZY+21, ZZ21, ZLWL23, ZWC23, ZWW24, ZWL24]

5.2 Taxonomy of Research Directions

We identified three main research directions within the 71 peer-reviewed publications on Neural Differential Cryptanalysis: understanding the inner workings of neural distinguishers, developing neural-aided (round) key-recovery attacks, and improving the state of the art

Table 1: Summary of the main research directions covered in the Neural Differential Cryptanalysis literature and the corresponding sections of this survey.

Research direction	# Works	Section
Neural distinguisher explainability	20	Section 6
Neural-aided key recovery	22	Section 7
Comparative review of distinguishers	59	Section 8

Table 2: Publicly available repositories for Neural Differential Cryptanalysis. For the topic, we consistently use the notation from our classification scheme (cf. Subsection 8.2). Further abbreviations: BS = Bit Sensitivity, Cond = Conditional, GNB = Generalized Neutral Bits, HPO = Hyperparameter Optimization, PCA = Principal Component Analysis, Quant = Quantum, WKRP = Wrong Key Response Profile.

Publication	Source	Framework	Topic	Publication	Source	Framework	Topic
[BR21]	[Cry21]	Tensorflow	$E = \mathrm{R}^+$	[BGPT21]	[Ano21]	PyTorch	Explain
[BGL+22]	[Dif22b]	Tensorflow	GNB	[BBP22]	[Nor22]	Tensorflow	Prune
[BLYZ23]	[BLY23]	Tensorflow	RK	[BFG+24]	[Pea23]	Tensorflow	Auto
[CSY23]	[AI-23a]	Tensorflow	BS	[CSYY23]	[AI-23b]	Tensorflow	Reuse
[HGH+23]	[Cry23c]	Tensorflow	BS	[HRC23]	[Mon23]	Unavailable	WKRP
[HLF+24]	[AI-24]	Tensorflow	BS	[KJL+23]	[Khj23]	Unavailable	Quant
[LLHC23]	[Ali23]	Tensorflow	Cond	[LRCL23]	[Can23]	Tensorflow	$T = A$
[LLS+24]	[JIN24]	Tensorflow	RK	[RRSM22]	[DRD22]	PyTorch	$E = D$
[SSL+22]	[ASC22]	PyTorch	LSTM	[SL25]	[BJS24]	Tensorflow	PCA
[TTJ23]	[Cry23e]	Tensorflow	LBCIoT	[WW24b]	[Dif24b]	Incomplete	Hybrid
[WWS24b]	[Dif24a]	Tensorflow	$E = D$	[WW24a]	[Dif24c]	Tensorflow	RK
[WWS24a]	[Dif23]	Tensorflow	BS	[YK21]	[Tar21]	Tensorflow	Hybrid
[YK24]	[Tar24]	Tensorflow	Amplify	[ZLWL23]	[Cry23a]	Tensorflow	$T = A$
[ZWC23]	[Cry23b]	Tensorflow	Inception	[ZWW24]	[Cry24b]	Tensorflow	$T = A$
[ZWL24]	[Cry24a]	Tensorflow	BS				

Additional related repositories.

Publication	Source	Framework	Topic	Publication	Source	Framework	Topic
[BGG+23]	[Cry23d]	Tensorflow	Library	[Goh19]	[Ago19]	Tensorflow	Seminal
[GLN22]	[Dif22a]	Tensorflow	HPO				

in neural distinguishers themselves. For ease of navigation, Table 1 outlines these research directions and indicates where they are discussed in this survey. We exclude three papers that were inaccessible at the time of writing [BPC24a, PCDC24, KS22].

We found contributions to the explainability (or interpretability) of neural distinguishers in the following 20 works [BGPT21, BBP22, BLYZ23, CSY23, DCC23, ERP22, Goh19, GLP+24, HGH+23, HLF+24, LRC23, LRC24, LJSC24a, SL25, Seo24, WWS24a, YW23, ZWW24, ZWC23, ZWL24], and will discuss their respective contributions in Section 6.

We found contributions to neural-aided key recovery in the following 22 works [BGL+22, BLYZ23, CSY23, CSYY23, Goh19, HRC23, HLF+24, LTZ22a, LCLH22, LLHC23, LRC24, LJSC24a, SZM20, Seo24, TH21, TTJ23, WQW+24, YW23, ZWW24, ZLWL23, ZLHH25, ZWL24], and will give an overview of these works in Section 7.

Lastly, most (66/68) accessible, peer-reviewed research on Neural Differential Cryptanalysis involves training neural differential distinguishers [BBDH21, BGPT21, BR21, BBCD21, BGL+22, BBP22, BGHR23, BBD+23, BLYZ23, BFG+24, BPC24b, CSY23, CSYY23, DCC23, ERP22, EGP23, HRC21c, HRC23, HGH+23, HLF+24, HLZH25, KJL+23, KKJ+24, LTZ22a, LCLH22, LTZ22b, LRC23, LRCL23, LLHC23, LLS+24, LRC24, Li24, LJSC24a, MPKM+22, PMDC22, PSM24, RRSM22, RKS23, SZM20, SSL+22, SM24, SSL+24, SL25, Seo24, SBG+25, TH21, TTJ23, TSL23, WW21, WWH21, WTZ+22, WQW+24, WW24b, WWS24b, WWS24a, WW24a, YK21, YW23, YK24, ZZY+21, ZZ21, ZLWL23, ZWW24, ZWC23, ZLHH25, ZWL24].

To assess which approaches have been most effective for constructing neural differential distinguishers, we present a comparative review of all reported distinguishers. To ensure a meaningful comparison, however, we exclude works that are not directly comparable. This includes papers whose primary objective is not to obtain the strongest possible distinguisher,

but rather to address other aspects such as explainability or compatibility with classical differentials [SL25, YK21, WW24b]; papers that provide insufficient detail on neural network training or do not report concrete accuracy results [HRC23, LRC24]; and papers that rely on leakage or adversary models outside standard security assumptions [PPWR23, SBG+25, HLZH25]. Consequently, in Section 8 we present a comparative review of 59 differential distinguishers, including Gohr's seminal work [Goh19].

In addition, in Table 2, we provide an exhaustive list of 29 associated repositories with the peer-reviewed works considered in this study and highlight three related frameworks: [BGG+23], a general and user-friendly framework that supports neural distinguishers; [Goh19], the seminal framework in this area; and [GLN22], a framework that supports systematic hyperparameter optimization. Two repositories [HRC23, KJL+23] are no longer publicly available since their publication, and in one case [WW24b], the source code itself has not been released; only experimental artifacts have been made available.

6 Overview: Neural Differential Distinguisher Explainability

The lack of explainability is the "machine's inability to explain its decisions and actions to human users" [GVWT21]. One of the major efforts in research on explainability was the 4-year program (2017-2021)" XAI" by the Defense Advanced Research Projects Agency (DARPA) of the United States Department of Defense *"DARPA's Explainable Artificial Intelligence (XAI) Program"* [GA19]. A more recent review of the research in XAI is given in *"Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence"* [HCM+24]. To this day, explainability remains an active research field in AI and has led to various approaches to add explainability to neural networks, e.g., pruning, ablation studies, and visualization techniques.

Gohr [Goh19] first showed that neural distinguishers can outperform classical differential distinguishers in practice. As a baseline, he considered distinguishers derived from the full difference distribution table (DDT), which records for each input difference δ the distribution of output differences Δ, that is, the probabilities $P(\delta \to \Delta)$. For round-reduced SPECK32/64, the neural distinguishers achieved better overall classification accuracy than these DDT-based baselines. This observation motivated 19 subsequent, peer-reviewed studies on the explainability of neural distinguishers, aiming to identify the source of their superior performance.

We identify five broad directions in explainability research. For each direction, with the exception of "ciphertext bits", we conduct a comparative analysis of the existing literature and conclude the corresponding subsection with a synthesis of the key findings. The "ciphertext bits" subsection is treated as an overview, as the relationships among the various bit-importance tests remain insufficiently understood, and a comprehensive comparative assessment is therefore beyond the scope of this work. We end this section with a concise paragraph on the key findings in neural differential distinguisher explainability.

Purely Differential Features Gohr et al. [GLN22] show that the success of a purely differential distinguisher (based on the DDT) is determined, up to constant factors, by the deviation of the full ciphertext-difference distribution from uniform. This differs from the classical view of differential cryptanalysis, which typically emphasizes a small number of high-probability transitions. Here, the distinguishing advantage instead arises from the aggregate non-uniformity of the entire output-difference distribution. The authors emphasize that this makes neural distinguishers a useful (approximate) cryptanalytic tool: for large block sizes, computing the full ciphertext-difference distribution, i.e., probabilities $P(\delta \to \Delta)$ for each input difference δ and output difference Δ, is often computationally expensive or infeasible, whereas neural networks can learn useful approximations within hours on consumer hardware. This also helps explain why Gohr's original attack [Goh19]

is most effective for an input difference that does not correspond to the single highest-probability differential transition in SPECK32.

While, in principle, a neural distinguisher should *at least* be able to approximate the deviation of the full ciphertext-difference distribution from the uniform distribution, doing so remains difficult in practice. For example, although 8-round reduced SPECK32 exhibits such a deviation, the neural network fails to learn it without a more elaborate staged training approach [Goh19]. A plausible explanation is that the deviation is too small to be reliably captured by the training data or to yield gradients strong enough for stochastic optimization to exploit. Overcoming this limitation remains one of the central open problems in the area. The situation is further complicated by the fact that, even for instances that are solvable in practice, such as 9-round SIMON32, the quality of the neural network's approximation can depend strongly on hyperparameter choices [GLN22].

Features Beyond Purely Differential Statistics Gohr investigates the capabilities of the provided neural networks by introducing a differential cryptanalytic task called the real differences experiment [Goh19]. In this task, half of the ciphertext pairs are randomized by XORing the same uniformly random mask M to both ciphertexts, yielding $(C_0 \oplus M, C_1 \oplus M)$. This preserves the ciphertext XOR difference exactly, so the randomized and non-randomized samples have the same ciphertext difference distribution. Consequently, any distinguisher based solely on differential information, such as the ciphertext difference distribution captured by the DDT, cannot separate the two classes better than random guessing. Gohr nevertheless shows that the neural distinguishers for SPECK32 still achieve noticeably above-chance accuracy in this setting, indicating that they learn and exploit structural features of the distribution of ciphertext pairs *beyond purely differential statistics*. In many subsequent works, the neural network input is preprocessed by XORing the ciphertext pairs, primarily to reduce complexity and speed up training. This transformation, however, necessarily eliminates the type of advantage observed by Gohr in the real differences experiment: once only the XOR difference is provided as input, the network can no longer learn or exploit features arising from the full ciphertext-pair distribution. We note that features beyond purely differential information generally yield only a modest increase in accuracy. For example, on 7-round SPECK-32, the accuracy increases from the purely differential baseline of 59.1% to 61.6% in [Goh19].

In a direct follow-up work, Gohr et al. [GLN22] show that, for ciphers admitting a key-independent representation of intermediate ciphertext values of the form $c' \oplus k'$, where k' consists of independent key bits, the full ciphertext-pair distribution contains no more information than the ciphertext-difference distribution. Hence, they show an inherent limitation: neural distinguishers cannot learn features beyond purely differential statistics for a class of ciphers. The authors show that this framework applies, under the assumption of independent round keys, to important cipher classes, including SPNs such as PRESENT and Feistel ciphers such as SIMON. Empirically, this limitation is confirmed for SIMON: its neural distinguisher achieves essentially the same accuracy as a DDT-based distinguisher, and in the real-differences experiment, its accuracy drops to 0.5, indicating exclusive reliance on differential information. By contrast, SPECK does not satisfy these conditions due to its use of modular addition, and its neural distinguishers continue to outperform purely differential baselines and remain above random guessing in the real-differences setting. The authors also find no evidence that the neural distinguisher exploits weaknesses in the key schedule, since replacing the derived round keys with independent random round keys does not affect the performance. This is further confirmed by Chen et al. [CSYY23], who explore two settings for multi-pair distinguishers: one in which all ciphertext pairs in a sample are generated under the same encryption key, and another in which each pair uses an independent key. They observe that the key-reuse pattern has no substantial effect on distinguishing performance.

In [BGPT21], Benamira et al. investigate which features are exploited by Gohr's neural distinguishers. By analyzing ciphertext pairs for which the neural distinguisher shows high confidence, they observe that the distinguisher relies not only on ciphertext-level differential information, but also on differential patterns in the penultimate and antepenultimate rounds. In particular, for 5-round SPECK32, they identify truncated differential masks at rounds 3 and 4, and show experimentally that these patterns can be recognized with very high accuracy; for 6 rounds, they report a similar, though less pronounced, trend. To further interpret the learned features, they also replace the first ReLU activation with a Heaviside activation, thereby binarizing the intermediate values and making the first convolutional layer more explicitly analyzable. They show that the modified network preserves essentially the same accuracy and that the first layer computes simple Boolean relations between quantities derived from the ciphertext halves. More broadly, their analysis suggests that Gohr's network behaves like a compact approximation of a richer masked output-distribution statistic, rather than a classical DDT alone.

Building on the study by Benamira et al. [BGPT21], Wang et al. [WWS24a] further extend the analysis beyond Gohr's original single-pair neural distinguisher to include distinguishers trained on ciphertext differences and multiple ciphertext pairs. In particular, they show that the ciphertext-difference-based distinguisher exhibits behavior closely paralleling that observed by Benamira et al., including sensitivity to high-probability or high-bias patterns in the penultimate and antepenultimate rounds, while a direct comparison between ciphertext-pair and ciphertext-difference inputs suggests that the former gains its advantage from exploiting additional relations between certain ciphertext bits. They also extend the analysis to multi-pair neural distinguishers, arguing that their improved accuracy is driven mainly by the average differential probability across the input pairs rather than by strong inter-pair dependencies. This interpretation is consistent with the assessment of Gohr et al. [GLN22], who argue that the gains claimed for multi-pair neural distinguishers are often overstated: after a fair comparison based on combining the scores of single-pair distinguishers, the advantage of multi-pair models is at most marginal.

Extending the line of analysis initiated by Benamira et al. [BGPT21], Bao et al. [BLYZ23] identify *explicit* features that can be incorporated into a purely differential, DDT-based distinguisher to improve its performance and help explain the gap to Gohr's neural distinguisher for SPECK32. These features are based on strong correlations between bit values in "right" pairs for XOR-differential propagation through addition modulo 2^n. The authors further show that these rules are closely related to earlier studies on multi-bit constraints and fixed-key differential probability and demonstrate that, once conditional differential information derived from partially known ciphertext values is incorporated into a classical distinguisher, its accuracy can match and even slightly exceed Gohr's neural distinguisher on their SPECK32 benchmarks. In contrast, they argue that neural differential distinguishers already exploit these rules, or equivalent forms of them, inherently. The authors further note that neural distinguishers can be more efficient while achieving accuracy comparable to sophisticated hand-crafted methods, and therefore conclude that neural distinguishers offer a good trade-off between efficiency and accuracy.

Overall, these results suggest that the question is close to being resolved. For ciphers admitting a representation of intermediate values of the form $c' \oplus k'$, with k' consisting of independent key bits, neural distinguishers appear to be fundamentally limited to purely differential information. This still makes neural distinguishers a useful cryptanalytical tool for approximating the deviation of the full ciphertext-difference distribution from the uniform distribution. For ciphers that do not admit such a representation, the remaining advantage can, at least to a large extent, be attributed to conditional, value-dependent differential information contained in the full ciphertext-pair distribution, especially in the presence of modular addition. There is, at present, no convincing evidence that neural distinguishers exploit the key schedule itself, nor that dedicated multi-pair variants derive

a substantial advantage from learning strong dependencies across ciphertext pairs.

The Architecture Benamira et al. [BGPT21] observe that replacing Gohr's neural architecture with standard non-neural classifiers generally leads to slightly lower accuracy. They further show that Gohr's distinguisher can be reinterpreted as a three-stage pipeline: first, the ciphertext pair (C_l, C_r, C'_l, C'_r) is transformed into a derived representation $(\Delta L, \Delta V, V_0, V_1)$ and linear combinations thereof. Second, a compressed output-distribution statistic is computed on this representation through a masked output distribution table, and third, a final classifier operates on the resulting probability-based feature vector. This reformulation achieves accuracy very close to that of Gohr's original distinguisher.

In [BBP22], Bacuieti et al. apply the *lottery ticket hypothesis*[4] to prune Gohr's convolutional networks. They show that the depth-1 distinguisher for SPECK32/64 can be reduced to just 10% of its original weights without, on average, any loss in performance relative to the unpruned network. Moreover, they demonstrate that residual connections are unnecessary and that a substantial portion of the network can be removed based on its activation pattern, while maintaining performance within one percentage point of the original. These results suggest that shallow, sparse architectures are sufficient to obtain meaningful distinguishers in this setting.

Taken together, these findings suggest that the initial nonlinear input transformation, along with the induced output distribution, is a central component of the neural distinguisher. More generally, learning Boolean functions is known to be a difficult problem for standard neural-network training in the worst case [MS20]. The present setting appears to exhibit a similar phenomenon: architectures such as convolutional neural networks operating on bit-level representations can learn useful Boolean-style features, whereas more generic architectures may fail when applied directly to the raw input. Subsequent evidence from lottery-ticket-style pruning and activation-pattern analysis further suggests that a substantial part of Gohr's neural distinguisher is redundant or only sparsely activated. This supports the view that the effective computation performed by the distinguisher may be much smaller than the original residual architecture, and that a shallow, sparse architecture with an appropriate bit-level preprocessing stage may be sufficient.

The Input Data Seok et al. [SL25] investigate the use of Principal Component Analysis (PCA) and K-means clustering to define metrics for evaluating the quality of datasets in differential-neural cryptanalysis. Their findings reveal that datasets associated with input differences that lead to successful distinguishers tend to have more axes that effectively represent the data than other datasets. Similarly, these datasets form multiple high-density clusters rather than a single spherical cluster. They introduce an input difference search method based on PCA and K-means clustering that surpasses the efficiency and effectiveness of the greedy approach proposed in [Goh19].

In [YW23], Yue et al. perform a comparative experiment to investigate the trade-off between improved accuracy and overfitting when using multiple ciphertext pairs per sample for neural differential distinguishers. Their experiments show that, when the total number of ciphertext pairs is held constant, increasing the number of pairs per sample improves accuracy but also increases the risk of overfitting, since it reduces the total number of available training samples. This finding is further supported by related experiments by Zhang et al. [ZWW24, ZWC23].

Overall, these results indicate that the effectiveness of a neural distinguisher is strongly shaped by the structure and size of its input data. Input differences that yield successful distinguishers appear to induce ciphertext-pair distributions with richer structure, such as informative principal directions and multiple dense clusters. However, learning to exploit

[4]The hypothesis that a subnetwork of the original network can be trained to achieve the same test accuracy as the full network within at most the same number of training iterations.

this structure reliably requires sufficiently large training sets. This seems to be particularly important in the multi-pair setting.

Ciphertext Bits Recent advances in neural distinguishers [CSY23, DCC23, ERP22, GLP+24, HGH+23, HLF+24, LRC23, LRC24, Seo24, WWS24a, ZWL24] demonstrate remarkable efficiency by operating on partial ciphertext information rather than complete outputs. These approaches advance interpretability in cryptanalysis by systematically identifying the most influential ciphertext bits across a range of analytical techniques and already impact neural-aided key recovery for large-state block ciphers (Section 7). An important open question is how these influential bits are related to one another and whether they reflect a common underlying structure. This subsection provides a complete list of the main works relevant to this question.

Chen et al. [CSY23] introduce bit sensitivity testing as an ablation test on individual ciphertext bit positions to identify informative bits, i.e., ciphertext bits that help distinguish between a cipher and a pseudo-random permutation. They successfully maintain a high distinguisher performance for SPECK32/64 while using only 16 of 32 ciphertext bits, as identified through their novel testing methodology.

Deng et al. [DCC23] introduce an attention mechanism [VSP+17] into deep-learning-based differential cryptanalysis of SPECK and use Grad-CAM visualizations to analyze which input-bit positions are most salient for the distinguisher. Concretely, they average heatmaps over 1000 ciphertext samples, identify the most salient bit positions by thresholding the heatmap values, and compare these positions between a baseline ResNet and their attention-augmented model for several input differentials and round numbers.

Ebrahimi et al. [ERP22] introduce a partial differential distinguisher for SPECK32/64 and show that, for six rounds, a distinguisher trained on only 8 carefully selected output-difference bits attains nearly the same accuracy as one trained on the full 32-bit difference. To identify these bits, they sample many 16-bit subsets, train a neural distinguisher on each subset, and assign each bit an effectiveness score.

Goi et al. [GLP+24] employ explainable AI techniques (LIME and SHAP, model-agnostic methods for explaining model predictions) to examine Gohr's neural distinguisher against SPECK32, revealing significant methodological differences: LIME effectively captures the significance of individual bits, while SHAP uniquely identifies important bit pairings in the ciphertext. LIME is also applied by Bacuieti et al. [BBP22], who concluded that LIME cannot identify the important features.

Hambitzer et al.'s [HGH+23] deep learning ensemble, NNBits, is specifically designed for bit profiling of cryptographic and pseudorandom bit sequences. Applied to Gohr's round-6 SPECK32/64 distinguisher, NNBits trains separate predictors for individual ciphertext bits and measures their predictability from the remaining bits, thereby yielding a detailed bitwise profile. Using only the correctness pattern of these single-bit predictors, the authors construct an ensemble distinguisher that reaches 78.1% accuracy, closely matching the 78.3% accuracy of Gohr's depth-1 network.

Huang et al. [HLF+24] train partial neural distinguishers for SPECK by decrypting a ciphertext pair with a pseudo-round key in which only the subkey bits associated with the selected ciphertext-bit range are kept, while all other subkey bits are set to zero. This training strategy enables partial distinguishers (operating on only a subset of ciphertext bits) to recover their corresponding key bits independently of the remaining subkey bits, enabling parallelizable key recovery.

Liu et al. [LRC23] present a systematic analysis of advantage bits, that is, ciphertext bits that carry particularly useful distinguishing information. Building on this idea, they propose an advantage-bit search algorithm that heuristically identifies a small subset of bits whose XOR differences preserve most of the cryptanalytic signal of the full ciphertext, thereby reducing the representation to only 8 bits for SPECK32.

Li et al. [LRC24] select sensitive bits using a key bit sensitivity test (KBST). Starting from teacher neural distinguishers trained on full ciphertext pairs for different single-bit input differences, they measure which last-round key bits most strongly affect distinguisher accuracy. They then map these sensitive key bits, via the cipher round function and key schedule, to the corresponding partial ciphertext-state bits used to train the student distinguishers for SPECK and SIMON.

Seok [Seo24] develops a truncated neural distinguisher for HIGHT that focuses exclusively on a subset of ciphertext bits. Unlike bit-selection methods based on sensitivity tests or feature scoring, the selected words are manually selected based on a structural analysis of HIGHT's round function.

Wang et al. analyze bit importance [WWS24a] using permutation feature importance (PFI). Concretely, they randomly permute individual ciphertext bits for SPECK32 and measure the resulting reduction in distinguisher accuracy. By comparing permutations applied to one ciphertext, the other ciphertext, or both simultaneously, they distinguish between information coming from ciphertext differences and additional information arising from relations between corresponding ciphertext bits.

Zhang et al. [ZWL24] extend neural cryptanalysis to AES128, training distinguishers for the 2-round reduced cipher and additionally examining specific intermediate states between rounds 2 and 3. Their approach replaces full 16-byte state processing with specialized networks operating on just manually selected 2-byte segments while maintaining nearly identical accuracy.

Key Insights In summary, existing explainability studies show that neural distinguishers primarily exploit deviations of the ciphertext-difference distribution from uniformity, while their additional advantage, when present, often stems from value-dependent differential information in the full ciphertext-pair distribution. This advantage appears strongly cipher-dependent: for some cipher classes, neural distinguishers are fundamentally limited to purely differential information, whereas operations such as modular addition, as in SPECK32, can expose richer exploitable structure. Across the literature, effective distinguishers depend on suitable bit-level architectures, informative input data, and carefully selected ciphertext bits. Overall, neural distinguishers are best understood not as opaque replacements for classical cryptanalysis, but as efficient tools for approximating and exploiting complex differential statistics that are difficult to compute explicitly.

7 Overview: Neural Aided Key Recovery Attacks

Gohr's work [Goh19] represents a significant advance in machine-learning-based cryptanalysis, as it produced state-of-the-art statistical distinguishers for 7-round SPECK32/64 and led to key-recovery attacks on 11- and 12-round variants whose performance matched or exceeded that of state-of-the-art key recovery methods. This observation motivated 21 subsequent, peer-reviewed studies on *key recovery* using neural distinguishers.

We present a comparative review of 15 works on neural-aided key recovery, including the original work by Gohr, in Table 3 and briefly discuss those with particularly noteworthy findings, without aiming for an in-depth treatment. For brevity, we omit attacks that are not meaningfully comparable, since they are the only results reported for their respective cipher categories: 8-round LBC-IoT [TTJ23], 8-round PRESENT [ZLHH25], 15-round HIGHT [Seo24], 10-round DES [CSY23], 12-round SPECK128, 14-round SIMON96, 19-round SIMON128 [LRC24], 14-round SIMON48/96 [HRC23], and 2-round AES [ZWL24]. Similarly, we exclude works in alternative adversarial settings due to limited comparability, such as the (full) key recovery attack on 14-round SPECK32/64 in the related-key setting [BLYZ23] and the (full) key recovery attacks on various KATAN variants in the related-key setting [LCLH22] and conditional setting (additional key-dependent conditional

Table 3: Comparison of 15 neural-aided key-recovery attacks. Some works are excluded due to limited comparability, as discussed in the main text. *Full* indicates implementation of a full Bayesian key-recovery procedure (cf. Section 4); *Hybrid* indicates use of a classical prepended differential transition, supported by (probabilistic) neutral bits (NB), generalized neutral bits (GNB), or filtering (F). *Discussed* marks results elaborated in the main body. Best attacks per cipher category are highlighted.

Work	Cipher	Rounds	Full	Hybrid	Discussed
[YW23]	SPECK32/64	8	-	-	-
[CSYY23]	SPECK32/64	11	-	NB	✓
[Goh19]	SPECK32/64	12	✓	NB	-
[CSY23]	SPECK32/64	13	✓	NB	✓
[BGL+22]	SPECK32/64	13	✓	GNB	✓
[ZWW24]	SPECK32/64	14	✓	GNB	✓
[LJSC24a]	SPECK32/64	14	✓	GNB	✓
[LRC24]	SPECK64	9	-	NB	✓
[HLF+24]	SPECK64/96	10	-	NB	✓
[LRC24]	SPECK96	10	-	NB	✓
[HLF+24]	SPECK96/96	10	-	NB	✓
[CSY23]	SPECK96/96	14	✓	NB	✓
[SZM20]	SIMON32/64	11	-	F	✓
[WQW+24]	SIMON32/64	12	✓	NB	-
[HRC23]	SIMON32/64	13	✓	NB	✓
[TH21]	SIMON32/64	15	✓	NB	-
[BGL+22]	SIMON32/64	16	✓	GNB	✓
[ZWW24]	SIMON32/64	17	✓	GNB	✓
[HRC23]	SIMON64/128	13	✓	NB	✓
[LRC24]	SIMON64	14	-	NB	✓
[LTZ22a]	SIMECK32/64	15	✓	NB	✓
[ZLWL23]	SIMECK32/64	17	✓	GNB	-

assumptions) [LCLH22, LLHC23]. We end this section with a concise paragraph on the key findings in neural-aided key recovery attacks.

We stress that key recovery is not a central contribution of this survey. Instead, this section provides a concise overview of how neural distinguishers are used in practice, while also situating prior results to help position future work. For a detailed discussion of the neural distinguishers employed in key-recovery attacks, including their feature engineering and architectural choices, we refer the reader to our comparative review in Section 8.

Small-State Ciphers In [CSYY23], the authors propose a *data-reuse strategy* for key recovery using multi-pair neural distinguishers, allowing ciphertext groups to be formed from a shared pool while limiting pair reuse and inter-group overlap. This reduces the data cost of distinguishers that process $k > 1$ ciphertext pairs. They apply the resulting distinguishers within existing key-recovery frameworks, demonstrate reduced-complexity 10-round Speck32/64 recovery with NASA [CSY23], and reproduce Gohr's 11-round Speck32/64 attack with an improvement in success rate. Indeed, their posterior-probability analysis suggests that multi-pair distinguishers improve the separation between the true key and wrong keys in Gohr's rank-score framework.

Bao et al. [BGL+22] extend classical neutral-bit methods through what they describe

as *generalized neutral bits* (GNB), in particular simultaneous-neutral bit-sets (SNBSs) and conditional simultaneous-neutral bit-sets (CSNBSs), while also introducing switching bits for adjoining differentials (SBfADs). These mechanisms enlarge the structures of conforming pairs available to the attacker, which is crucial for boosting neural distinguishers through combined responses and thereby enabling improved 12-round and practical 13-round attacks on SPECK32/64, as well as a practical 16-round attack on SIMON32/64.

Zhang et al. [ZWW24] present key-recovery attacks on 13- and 14-round SPECK32/64, with the 14-round attack derived by augmenting the 13-round attack core with an exhaustive search over one additional round subkey, thereby adding a multiplicative complexity factor of 2^{16}. They further report the first 17-round key-recovery attack on SIMON32/64. In addition, they use *knowledge distillation* to derive compact student distinguishers, substantially lowering the computational cost of inference during key recovery while retaining strong practical effectiveness.

Lv et al. [LJSC24a] replace the neural distinguishers in the standard differential-neural SPECK32 key-recovery pipeline with *simplified differential-linear distinguishers* (MDLDs). Concretely, they map each ciphertext pair to a 20-bit feature vector via inner products with 20 differential-linear masks, then use a precomputed lookup table of size 2^{20} to score candidates, avoiding online neural network inference and reducing time complexity. With this substitution, they obtain a faster practical 13-round single-key attack on SPECK32, while their 14-round result follows the same pattern as prior extensions [ZWW24]: it is derived by exhaustively guessing an additional 16 bits of the next-round subkey.

Su et al. [SZM20] propose a two-stage key-recovery strategy based on their polytopic neural distinguishers. First, an $(r+1)$-round distinguisher is used to *filter* a large pool of candidate data generated from a prepended differential path. Then, an r-round neural distinguisher is applied to the filtered data to rank the final-round subkey candidates. They also introduce a *Bayesian Key Research with Error* variant designed to tolerate residual "bad points" in the wrong-key response table, although in practice they report that roundkey enumeration performs better.

Lyu et al. [LTZ22a] stand out in that they do not optimize neural distinguishers for accuracy alone. Instead, for SIMECK32/64, they choose distinguishers based on their *usefulness in Bayesian key search*, emphasizing regular wrong-key response profiles and compatible prepended differentials with sufficient neutral bits. This attack-oriented criterion leads to 8-, 9-, and 10-round neural distinguishers and successful recovery of the penultimate and last round subkeys for 13-, 14-, and 15-round SIMECK32/64, while showing that the most accurate distinguisher need not be the best for key recovery.

Large-State Ciphers Chen et al. [CSY23] propose a *Neural-Aided Statistical Attack* (NASA) for reduced-round SPECK32/64, DES, and Speck96/96. In contrast to Gohr's key-recovery framework [Goh19], which ranks key guesses directly from accumulated neural-distinguisher scores and relies on sufficiently many neutral bits in the prepended differential, NASA recasts key recovery as a statistical hypothesis test: for each key guess, the distinguisher outputs are thresholded and aggregated into a statistic whose right-key and wrong-key distributions can be modeled, enabling theoretical estimates of data complexity. The authors further reduce complexity through three optimizations: *student distinguishers* trained on selected subsets of ciphertext bits, a *Bit Sensitivity Test* to identify informative bits, and a *selective Bayesian key-search procedure* inspired by Gohr's wrong-key response strategy. We note that the authors present (only) theoretical complexity estimates for their key-recovery attacks on 13-round SPECK32/64, 14-round SPECK96/96, and 10-round DES.

Li et al. [LRC24] propose an improved multi-stage key-recovery framework based on combinations of *student distinguishers*. Each student distinguisher is trained for a selected input difference and a selected subset of ciphertext bits, so that it targets only

a corresponding subset of last-round key bits. Their key-bit sensitivity test identifies which key bits most affect each distinguisher, allowing the last-round subkey space to be partitioned into smaller components that can be recovered stage by stage. Using this framework, they report key-recovery attacks on large-state variants of SIMON and SPECK.

Huang et al. [HLF+24] propose a parallelizable multi-stage differential-neural key-recovery framework for large-state SPECK. Their method uses independently trained *partial neural distinguishers* (PNDs) to recover subsets of last-round key bits, followed by a full neural distinguisher (FND) to select the final key candidate. A pseudo-round-key training strategy reduces dependencies between key-bit subsets, enabling parallel recovery. They reduce the attack time for 9-round SPECK64/96 and 10-round SPECK96/96, and extend the SPECK64/96 attack to 10 rounds using 7-round PNDs trained with neutral-bit structures, additional round-state information, and staged training.

Hou et al. [HRC23] observe that the wrong-key response profile exhibits a *pseudo-periodic structure*, meaning that the neural distinguisher's mean and variance under wrong-key decryption follow similar trends across different key intervals. Exploiting this property, they approximate the full wrong-key response profile from only a partial profile and use the resulting pseudo-profile for Bayesian key search. This substantially reduces the otherwise prohibitive precomputation cost for ciphers with larger round-key spaces, such as SIMON64/128, where generating the complete profile would be impractical.

Constraints on Neural-Aided Key Recovery Hybrid distinguishers are central to Gohr's key-recovery framework, and extending the prepended classical differential while preserving sufficiently large plaintext structures has become a major route to stronger neural-aided attacks [BGL+22]. Several works [YK21, WW24b] build longer hybrids by training an s-round neural distinguisher on the output difference of a long r-round classical trail. Although such $(r+s)$-round hybrids can reduce distinguishing data complexity, their use in Gohr-style key recovery is not immediate. Indeed, long classical prefixes may not provide enough neutral bits, or generalized neutral bits, to form the plaintext structures required for score aggregation. Instead, Yadav et al. [YK21] suggest a filtering-based approach (as in [SZM20]), but do not validate it experimentally.

A related limitation arises when the neural distinguisher itself has low accuracy. Yadav et al. [YK24] amplify weak neural distinguishers by aggregating predictions over many ciphertext pairs and applying a cutoff rule, which is similar in spirit to Gohr's original score aggregation over plaintext structures [Goh19, GLN22]. However, integrating such amplification into key recovery is nontrivial: the aggregated samples cannot be arbitrary, but must jointly pass the prepended classical differential so that they induce the same input distribution for the neural distinguisher. This requires sufficiently large plaintext structures generated from neutral bits or related mechanisms. Bao et al. [BGL+22] show that this can be achieved for the weak 8-round Speck distinguisher by using generalized neutral-bit techniques, which provide large enough structures to make 13-round key recovery practical.

Key Insights Overall, the best neural-aided key-recovery results are still shaped by Gohr's original framework [Goh19], including its selective Bayesian key-search strategy. The main extension to small-state ciphers comes from Bao et al.'s generalized-neutral-bit framework [BGL+22], which enables longer prepended differential transitions and, consequently, longer hybrid key-recovery attacks. For large-state block ciphers, by contrast, the dominant approach is to use partial distinguishers that recover subsets of the last-round key independently. Most large-state results have not yet incorporated generalized neutral bits, suggesting a route for improvement. There is also preliminary evidence [CSY23, HLF+24] that multi-pair distinguishers can improve the complexity of the key recovery.

8 Comparative Review: Neural Differential Distinguishers

In the following, we provide a comparative review of all trained neural differential distinguishers reported up to February 2025. To support a systematic comparison, we first introduce the neural network architectures proposed in the literature (Subsection 8.1) and present a classification scheme (Subsection 8.2) that captures the main settings considered across prior works. We then comparatively review the leading neural differential distinguishers for the most relevant symmetric primitives (Subsection 8.3), reserving the discussion of cross-primitive performance trends and key findings for Subsection 8.4.

8.1 Architectures

In this section, we review the neural network architectures employed in Neural Differential Cryptanalysis. Rather than treating architectures as isolated implementation choices, we compare them based on their intended functionality.

$\mathcal{ND}_{\text{Gohr}}$ $\mathcal{ND}_{\text{Gohr}}$ is the neural network architecture first proposed by Gohr in his seminal work on Neural Differential Cryptanalysis [Goh19]. It consists of four main components: (1) an input reshaping step that represents a ciphertext pair as a word-oriented matrix, thereby reflecting the internal structure of the cipher; (2) an initial bit-sliced convolution with kernel size one, which acts independently on corresponding bit positions and learns simple Boolean relations between the input words [BGPT21]; (3) a residual convolutional tower of varying depth, typically depth 1, 5, or 10, which extracts higher-level local dependencies and is shown to only be sparsely activated once trained [BBP22]; and (4) a fully connected prediction head that performs the final binary classification. The architecture has been shown to generalize well across a range of symmetric primitives, requiring only minor hyperparameter adjustments [GLN22]. However, training $\mathcal{ND}_{\text{Gohr}}$ on challenging ciphertext-pair distributions, such as 8-round SPECK32/64, requires designing an elaborate, staged training using differential-cryptanalytic expertise and large amounts of data ($\approx 2 \cdot 10^9$) [Goh19]. Subsequent work shows that training a distinguisher for fewer rounds and then freezing all layers except the prediction head can effectively reproduce the performance of the staged training pipeline [BLYZ23].

Several variants extend the original design: $\mathcal{ND}^{\text{pruned}}_{\text{Gohr}}$ removes the residual connection from the depth-1 variant of $\mathcal{ND}_{\text{Gohr}}$ and further prunes convolutional channels and dense neurons with low activation, yielding a more compact architecture [BBP22]; $\mathcal{ND}^{\text{attntn.}}_{\text{Gohr}}$ augments the residual-network design of $\mathcal{ND}_{\text{Gohr}}$ with a multi-head attention block placed after the residual tower, allowing the model to capture longer-range dependencies and focus on particularly informative ciphertext bits [DCC23]; $\mathcal{ND}^{\text{ensmbl.}}_{\text{Gohr}}$ uses an ensemble of Gohr-style neural networks to perform bit-level profiling of ciphertext-derived sequences, yielding an ensemble-based distinguisher [HGH+23]; and $\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$ replaces the standard convolutions in the residual tower of $\mathcal{ND}_{\text{Gohr}}$ with depthwise separable convolutions, thereby reducing the number of trainable parameters and the computational cost [PMDC22, LRC23]. $\mathcal{ND}_{\text{Gohr}}$ and variants thereof have been applied to 15 symmetric primitives.

Classical ML Classical ML refers to non-neural machine learning methods, such as AdaBoost, Random Forests, and Decision Trees. These approaches typically achieve substantially lower accuracy than CNN-based distinguishers, often by at least 20 percentage points [ZZ21, BGPT21]. Nevertheless, some methods, such as SVMs, can be competitive in specific settings and are therefore included where relevant [BBD+23].

DBitNet DBitNet was designed as a "cipher-agnostic" alternative to $\mathcal{ND}_{\text{Gohr}}$ [BGHR23]. Built on dilated convolutional layers that learn dependencies between distant rather than

neighboring neurons, DBitNet eliminates the need for input reshaping and bit-slicing convolution. This design enables automatic distinguisher generation across multiple primitives through a substantially simpler staged training strategy. The architecture successfully generates distinguishers for eight primitives: SIMON (Table 4), SPECK (Table 6), GIMLI (Table 14), HIGHT (Table 16), KATAN (Table 17), LEA (Table 20), PRESENT (Table 21), and TEA (Table 26).

DenseNet. DenseNet replaces the standard convolutional structure of $\mathcal{ND}_{\text{Gohr}}$ with dense connectivity: each layer receives the concatenation of all previous feature maps, i.e., $\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}])$. This encourages feature reuse and helps gradients flow through the network, but also increases the amount of intermediate feature data passed between layers. Bao et al. investigate DenseNet as an alternative to Gohr's $\mathcal{ND}_{\text{Gohr}}$ when building neural distinguishers for SIMON32/64 [BGL+22]. In their experiments, however, DenseNet is generally outperformed by SE-ResNet.

Inception The INC architecture adapts GoogLeNet's Inception module to neural differential distinguishers by applying parallel convolutional layers with different kernel sizes, enabling multi-scale feature extraction and a larger receptive field [ZWW24], though at increased computational cost compared to manual kernel size selection in $\mathcal{ND}_{\text{Gohr}}$ [GLN22]. In its original form, the Inception module is placed before the residual tower. INC has been applied to SIMON (Table 4), SIMECK (Table 5), SPECK (Table 6), CHASKEY (Table 10), DES (Table 11), and PRESENT (Table 21).

Quantum Neural Networks Quantum Neural Networks (QNNs) use parameterized quantum circuits as trainable components, typically within hybrid quantum-classical models. Kim et al. [KJL+23] propose a QNN-based neural distinguisher for 5-round SPECK-32/64 (Table 6). Since no large fault-tolerant quantum computer was available, their experiments were run on a classical simulator of quantum circuits. The result is best viewed as a feasibility study rather than a cryptanalytic improvement: the quantum distinguisher performs far worse than the baseline $\mathcal{ND}_{\text{Gohr}}$. Thus, QNNs currently remain an experimental architecture for neural distinguishers, with no demonstrated advantage over classical networks.

SE-Networks. Compared to Gohr's original ResNet-based distinguisher, SE-ResNet (Squeeze-and-Excitation Residual Network) adds a lightweight mechanism that learns which feature channels are most useful [BGL+22, LLS+24, WW24a]. A squeeze-and-excitation block first summarizes each channel by global average pooling, then uses a small fully connected network to assign channel weights, and finally rescales the feature maps with these weights. This helps the model strengthen useful differential signals and weaken noisy ones, with little extra cost. SE-ResNet has been applied on SIMON (Table 4) and SIMECK (Table 5). The authors report that this architecture outperforms $\mathcal{ND}_{\text{Gohr}}$ on SIMON32/64. However, Gohr et al. observe that the $\mathcal{ND}_{\text{Gohr}}$ baseline used in that comparison had unfavorable hyperparameter choices [GLN22]. This makes the claimed architectural advantage less conclusive.

Sequence Models LSTMs are recurrent models that process inputs as sequences. In neural cryptanalysis, they have been tested when differences or features of ciphertext pairs are arranged sequentially. Baksi et al. find that LSTMs learn some signal, but are slower and less accurate than tuned MLPs [BBCD21]. Later works [SSL+22] nevertheless use LSTM for GIFT (Table 13) with some success. Transformer architectures use attention to model dependencies across ciphertext sequences. Bose et al. propose a Transformer-based encoder-only classifier (TbEC), where ciphertext pairs and their XOR difference are embedded

as token sequences and classified by a Transformer encoder [BPC24b]. Both sequential models are applied to HIGHT (Table 16), LEA (Table 20), and PRESENT (Table 21).

MLP MLP denotes a multi-layer perceptron, a fully connected feed-forward architecture in which each neuron is connected to all neurons in the subsequent layer. Owing to their simplicity and computational efficiency compared to training $\mathcal{ND}_{\text{Gohr}}$, MLP has been investigated for 12 of the 24 primitives considered in this survey. However, MLP has been shown to be unable to replicate the distinguishing performance of $\mathcal{ND}_{\text{Gohr}}$ [BGPT21].

UNet UNet is an encoder-decoder architecture originally introduced for image segmentation and later adapted in cryptanalytic neural distinguishers. Its encoder progressively compresses and abstracts the input representation, while the decoder reconstructs task-relevant features. Skip connections transfer fine-grained information from the encoder to the decoder, helping preserve spatial or positional dependencies that may otherwise be lost during downsampling. In the BCS model of Zhu et al. [ZLHH25], UNet with long-range skip connections is applied to GIFT (Table 13) and PRESENT (Table 21).

8.2 Classification Scheme n-m-T-E for Neural Distinguishers

The proliferation of diverse training configurations for neural distinguishers often complicates the comparison of results across studies. Bellini et al. [BGHR23] addressed this challenge by proposing a systematic classification framework based on four key parameters: n, m, T, and E. We adopt this classification scheme throughout our review due to its demonstrated robustness in organizing the extensive cryptographic literature, extending it with additional taxonomic symbols to enable more nuanced categorization. We note that [BGHR23] is a technical contribution paper focused on developing automated neural cryptanalysis tools, not a survey paper, and makes no claims regarding a systematic or comprehensive collection of research literature.

To illustrate this framework, consider Gohr's foundational approach, which exemplifies the 2-1-CT-R configuration: the distinguisher processes pairs of ciphertexts ($n = 2$), considers only a single input difference ($m = 1$), receives raw ciphertext pairs without preprocessing ($T = \text{CT}$), and performs binary classification to detect the usage of the input difference ($E = \text{R}$).

Parameter Overview of n-m-T-E Classification

- n: Number of ciphertexts per training sample
- m: Number of input differences per training sample
- T: Feature engineering on training sample
- E: Type of distinguishing experiment performed

8.2.1 Number of Ciphertexts per Sample: n

The foundational work of Gohr [Goh19] already showed that the outputs of a single-pair neural distinguisher can be aggregated over multiple ciphertext pairs during key recovery. This idea was later revisited and made explicit by Gohr, Leander, and Neumann [GLN22], who observed that several subsequent works on multi-pair distinguishers effectively rediscovered the same statistical effect, such as [HRC21b, SZM20, ZWW22, BGPT21, LLS+24].

This aggregation has a natural statistical interpretation. A neural distinguisher may be viewed as a learned approximation to a binary hypothesis test for distinguishing random data from cipher-generated data. Let X_i denote a ciphertext pair generated either

from plaintexts with a fixed input difference or from a random reference distribution. If $X_1, \ldots, X_\ell$ are modeled as independent samples drawn from either f_0 or f_1, then the Neyman–Pearson lemma [NP33] states that, for simple hypotheses, the most powerful test among all tests with Type I error at most α is obtained by thresholding the likelihood ratio, or equivalently, the log-likelihood ratio,

$$\log \Lambda(X_1, \ldots, X_\ell) = \sum_{i=1}^{\ell} \log \frac{f_1(X_i)}{f_0(X_i)}.$$

In Gohr-style key recovery (cf. Section 4), the true distributions f_0 and f_1 are not known explicitly; instead, the neural distinguisher provides scores $R^i_{k'} \in [0, 1]$ after partially decrypting the i-th pair under a candidate subkey k'. If these scores are interpreted as probabilities for the encryption class, then their odds can be multiplied under an independence assumption. Equivalently, for key ranking, one may sum the log-odds,

$$A_{k'} = \sum_{i=1}^{\ell} \log\left(\frac{R^i_{k'}}{1 - R^i_{k'}}\right).$$

Similarly, in [GLN22], Gohr et al. aggregate the scores of a single-pair distinguisher $R_1, \ldots, R_\ell$ to ℓ ciphertext pairs, where R_i is interpreted as the probability that the i-th pair belongs to the encryption class: Under an independence assumption, the combined encryption probability is

$$\frac{1}{1 + \prod_{i=1}^{\ell} \frac{1-R_i}{R_i}}.$$

A multi-pair neural distinguisher could, in principle, do more than this independent aggregation: by receiving several ciphertext pairs jointly, it might exploit dependencies between pairs. However, in the absence of evidence that such additional structure is being learned (cf. Section 6), existing multi-pair distinguishers should be interpreted with caution: unless they outperform the independent log-odds aggregation baseline, their gain is best explained as standard statistical accumulation rather than as evidence of learned inter-pair structure.

This also affects how results in the literature should be compared. A reported single-pair accuracy alone is insufficient to reproduce the independent-aggregation baseline, because aggregation requires access to the distinguisher's soft outputs R_i, not merely its thresholded accuracy. Consequently, without access to the trained models or sufficiently detailed score distributions, it is generally not possible to fairly compare a multi-pair distinguisher against the corresponding non-aggregated single-pair baseline. For this reason, we report the best distinguishers in the standard single-pair setting ($n = 2$) and treat multi-pair results separately unless such an aggregation baseline is available. We believe that the burden of proof lies with authors who claim superiority for a multi-pair distinguisher: they should demonstrate that its gain cannot be explained by independent aggregation.

8.2.2 Number of Input Differences: m

Traditional differential cryptanalysis often focuses on a single input difference, whereas multiple differential cryptanalysis generalizes this setting by exploiting a set of differentials, potentially involving several input and output differences [BG11]. Blondeau and Gérard provide a theoretical framework for analyzing the data and time complexities, as well as the success probability, of such attacks. The benefit comes from aggregating statistical information from several differential events rather than relying on a single trail or differential. In favorable cases, when several input differences contribute comparable information, the data complexity can decrease roughly proportionally to the number of useful differences; however, the exact gain depends on the associated differential probabilities.

Mirroring this trend in Neural Differential Cryptanalysis, recent work on neural differential cryptanalysis has explored several ways of incorporating multiple input differences into the distinguisher input. Although these approaches are motivated by different classical frameworks, they share the same basic idea: instead of training on a single plaintext pair $(P, P \oplus \delta)$, the neural network is given ciphertext data derived from several related plaintexts or several differential pairs.

Two works [SZM20, SSL+22] adapt ideas from *polytopic cryptanalysis* to construct neural distinguishers with multiple differences relative to a common plaintext anchor. Conceptually, these methods use plaintext tuples of the form $(P, P \oplus \delta_1, P \oplus \delta_2, \ldots)$, which yield $n-1$ anchor-based differential pairs from n related plaintexts. These works are therefore best understood as polytopic-inspired neural input formats rather than full polytopic cryptanalytic attacks: the anchor-based structure provides richer inputs to the neural network, but the attacks do not explicitly exploit polytopic transition constraints.

Wu et al. [WQW+24] propose an approach that uses the same anchor-based principle, specialized to plaintext quadruples. The difference is mainly in the interpretation and use of the structure: Wu et al. formulate the construction in the language of *mixture differential cryptanalysis* and combine the resulting neural distinguisher with a classical mixture-differential transition in a key-recovery attack.

Wang et al. [WTZ+22] investigate the same general multiple-input-difference idea for SIMECK32/64 through two variants. In ND_{rm}, a sample is formed by concatenating independent ciphertext pairs, one for each input difference Δ_i; thus, m input differences give $n = 2m$ ciphertexts. In ND_{am}, the plaintexts are associated sequentially: starting from a random plaintext P_0, each subsequent plaintext is generated as $P_i = P_{i-1} \oplus \Delta_{i-1}$, so that m input differences give $n = m + 1$ related plaintexts and ciphertexts. This is not identical to the fixed-anchor form above, since the later plaintexts are related to P_0 through cumulative differences, but it implements the same overarching idea of enriching the neural input by combining several input differences in one sample.

Multiple works [BBDH21, BBCD21, BBD+23, RRSM22, WWH21, WW24a, WWS24b, WWS24a] train neural distinguishers to differentiate between $m > 1$ different output difference distributions (cf. E =D in Subsubsection 8.2.4).

8.2.3 Feature Engineering Type: T

Feature Engineering T Overview

- CT: Unprocessed ciphertext pair handed to $\mathcal{ND}$
- δ: Ciphertext XOR handed to $\mathcal{ND}$
- $\mathrm{CT}_{\mathbf{tr}}/\delta_{\mathbf{tr}}$: Reduced-bit ciphertext pair or XOR difference handed to $\mathcal{ND}$
- A: Advanced processing, such as partial decryption

Raw Ciphertext Processing ($T = \mathrm{CT}$) The original feature engineering type [Goh19] simply passes the raw ciphertext pairs directly to the neural distinguisher. While this preserves maximum information content, it requires networks to learn differential patterns implicitly, potentially increasing training and architectural complexity [BGPT21].

XOR Difference Features ($T = \delta$) Several works, including [BBCD21, HRC21c, YK21], demonstrate that replacing ciphertext pairs with their XOR differences can speed up training while maintaining comparable performance. This transformation reduces input dimensionality and provides direct access to the output difference at the cost of preventing the distinguisher from learning anything beyond differential features (cf. Section 6).

Truncated Variants ($T = \mathrm{CT}_{\mathrm{tr}}/\delta_{\mathrm{tr}}$) Some works [CSY23, DCC23, ERP22, GLP+24, HGH+23, HLF+24, LRC23, LRC24, Seo24, WWS24a, ZWL24] reduce training requirements by truncating ciphertexts to a few bits before handing them to the distinguisher ($T = \mathrm{CT}_{\mathrm{tr}}$), thereby significantly decreasing sample-size requirements at the cost of information loss. In some cases, the input size can be reduced even further by computing the XOR differences in addition to truncation ($T = \delta_{\mathrm{tr}}$).

Advanced Variants ($T = A$) In two works [BPC24a, PMDC22], the authors explicitly append the XOR difference to the corresponding ciphertext pair. This makes the basic differential information directly available as an additional input feature, rather than requiring the neural network to infer it from the two ciphertexts. However, more complex derived features, such as those studied in [BLYZ23], still have to be learned.

Partial decryption without key knowledge. Several works exploit the fact that, for some cipher structures, parts of the last round can be inverted without knowing the key. For SPECK32, this amounts to applying simple operations, such as XORing cipher halves and applying rotations, to expose intermediate structural information, which can improve distinguisher accuracy [BGPT21, HLF+24, WWS24a, ZWW24]. Similar feature-engineering ideas have been applied to SIMON-like ciphers. Bao et al. [BGL+22] augment the ciphertext-pair representation with the key-independent previous-round right-half difference, thereby giving the network direct access to information that can be obtained by partially inverting the last round. Lu et al. [LLS+24] extend the same principle by additionally including a partial estimate of the right-half difference two rounds earlier. In both cases, the goal is to expose prior-round differential structure explicitly in the input representation, rather than requiring the neural network to infer it from raw ciphertext pairs. The same type of SIMON-oriented feature engineering is also used in [WW24a, ZLWL23].

Gohr et al. [GLN22] show that, for several ciphers, neural distinguishers can often learn such transformations from raw ciphertext pairs, provided the architecture and hyperparameters are well tuned. Thus, these feature transforms should not necessarily be viewed as fundamentally new information, but rather as cipher-specific preprocessing that can make relevant structure easier for the network to access.

Partial Decryption With Key Assumption. Yue et al. [YW23] introduce an augmented SPECK32/64 input format that includes the partially inverted right halves of the ciphertext pair together with an additional feature, which approximates the previous-round left-half difference while ignoring the last-round subkey, i.e., by treating it as the zero key. The MRMSD/MRMSP approaches of Liu et al. [LRCL23] further generalize this feature-engineering direction by combining multi-pair inputs with prior-round information. Their construction first decrypts the ciphertext pairs by one round using randomly generated subkeys, producing false penultimate-round data that is then spliced with the original ciphertext data. In MRMSP (Multiple Rounds Multiple Splicing Pairs), the resulting sample consists of multiple ciphertext pairs together with their randomly decrypted counterparts, whereas MRMSD (Multiple Rounds Multiple Splicing Differences) applies the same idea at the level of output differences. Zhu et al. [ZLHH25] propose the BCS model for PRESENT and GIFT, extending this line of feature engineering from round-level preprocessing to component-level preprocessing. Similar to the MRMSD/MRMSP construction [LRCL23], they use randomly generated round keys to partially decrypt ciphertext pairs and extract intermediate S-box input/output features. Their ASMCP format augments multiple ciphertext pairs with the corresponding S-box input/output pairs, while ASMOD uses the analogous S-box input/output differences.

Differential-Linear Feature Extraction. Lv et al. [LJSC24a] combine differential and linear-neural ideas by first transforming each ciphertext pair (C_0, C_1) into a vector of linear-mask evaluations. For masks (Γ_i, Γ'_i), they compute bits of the form $z_i = \Gamma_i \cdot C_0 \oplus \Gamma'_i \cdot C_1$, yielding an N-bit feature vector $\mathbf{z} = z_1 \| \cdots \| z_N$. The classifier then uses

$\mathbf{z}$, rather than the raw ciphertext pair, to decide whether the underlying plaintext pair satisfies $P_0 \oplus P_1 = \delta$.

8.2.4 Type of Distinguishing Experiment: E

Experiment Type E Overview

- R: $(C_0 \| C_1) = (E_K(P_0) \| E_K(P_1))$, classify whether $P_0 \oplus P_1 = \delta$.
- $\mathrm{R_M}$: $(C_0 \oplus M) \| (C_1 \oplus M)$, where $P_1 = P_0 \oplus \delta$; classify whether $M = 0$.
- R^+: $(C_0 \| C_1) = (E_K(P_0) \| E_K(P_1))$, classify whether $P_1 = P_0 \circledast \delta$, where $\circledast \in \{\boxplus, \gg\}$.
- D: $(C_0 \| C_1) = (E_K(P_0) \| E_K(P_0 \oplus \delta_i))$, identify $i \in \{0, \ldots, m-1\}$.

Standard Differential Detection ($E = \mathrm{R}$). Gohr's foundational experiment [Goh19] uses ciphertext-pair samples $(C_0 \| C_1) = (E_K(P_0) \| E_K(P_1))$, where the network decides whether the corresponding plaintext pair satisfies $P_0 \oplus P_1 = \delta$. This binary real-versus-random classification task is the standard neural analog of differential distinguishing: positive samples are generated from plaintext pairs with the fixed input difference δ, while negative samples are generated from independent random plaintexts.

The standard differential-detection experiment is designed for key recovery: the key recovery procedure relies on the wrong-key randomization hypothesis [Goh19], namely that decryption under a wrong key yields ciphertext-pair decryptions that are close to the random-label setting, which the model was explicitly trained to detect. It is therefore the only experiment considered here with a demonstrated application to key recovery (cf. Subsection 4.3). Although the other experiment types are interesting in their own right, their applicability to key recovery has yet to be demonstrated. For this reason, we report the best distinguishers for the standard differential-detection experiment ($E = \mathrm{R}$), and treat alternative experiments separately. In our view, the utility of distinguishers obtained from alternative experiment types should first be established for key recovery, since this remains the main application of neural differential distinguishers.

Real Ciphertext Masking ($E = \mathrm{R_M}$). Gohr's "real ciphertext" experiment [Goh19] modifies a valid ciphertext pair (C_0, C_1), generated from $P_1 = P_0 \oplus \delta$, by applying a common mask M to both ciphertexts, giving $(C_0 \oplus M) \| (C_1 \oplus M)$. The task is to distinguish the unmasked case $M = 0$ from the case where M is random. Since the XOR difference $C_0 \oplus C_1$ is unchanged by the common mask, success in this experiment indicates that the neural distinguisher exploits information beyond the ciphertext difference alone.

Alternative Difference Operations ($E = \mathrm{R}^+$). Some works replace the XOR relation $P_1 = P_0 \oplus \delta$ with an alternative input relation $P_1 = P_0 \circledast \delta$. Bellini et al. [BR21] adapt the framework to modular-addition differences for ciphers such as TEA and RAIDEN, while Ebrahimi et al. [EGP23] consider rotational-XOR style differences. These variants preserve the same binary distinguishing setup as $E = \mathrm{R}$, but change the algebraic relation used to generate positive plaintext pairs.

Multi-Difference Classification ($E = \mathrm{D}$). Several works [BBDH21, BBCD21, BBD+23, RRSM22, WWH21, WW24a, WWS24b, WWS24a] replace the binary real-versus-random task by a multi-class classification problem. Instead of fixing one input difference δ, they choose a set of differences $\delta_0, \ldots, \delta_{m-1}$. Samples are generated as $(C_0 \| C_1) =$

$(E_K(P_0)\|E_K(P_0 \oplus \delta_i))$, and the network is trained to identify the index i. The random-guessing baseline is therefore $1/m$, rather than $1/2$.

This classification framework provides a systematic foundation for comparing neural distinguisher approaches across the literature. The parameter space defined by *n-m-T-E* enables researchers to precisely characterize their methodologies and identify gaps for future exploration.

8.3 Comparative Review

Based on the whole body of research in Neural Differential Cryptanalysis (Subsection 5.1), this section provides a comparative review of all best published neural distinguishers, classified according to the previously introduced scheme (Subsection 8.2), together with their neural network architectures (Subsection 8.1).

The neural differential distinguishers of each publication were selected as follows: *i)* We present the *best* result of each work, either the standard setting (2-1-⋆-R/R$^+$) or an alternative setting (*n-m-T-E*). If a result in the standard setting is also provided, we will present it as well. *ii)* In most works, no error margins on the results are provided, preventing us from displaying them. Ideally, the accuracies shown should be test accuracies on sets of several fresh samples. However, in many works, only the validation accuracy is reported. *iii)* Note that from a machine learning and statistical perspective, the number of training and validation samples is very important. However, from a cryptographic perspective, the number of needed encryptions, i.e., ciphertexts, is more relevant. Accordingly, the numbers reported in the following under **Trn.** (training data) and **Val.** (validation data) are the number of *ciphertexts*.

We report the best distinguishers for cipher variants with at least two distinguishers in the standard setting, since this is the minimum requirement for a meaningful comparison. As discussed in Subsubsection 8.2.1 and Subsubsection 8.2.4, we exclude multi-pair distinguishers and alternative experimental settings from this comparison. The implications of the best distinguishers are discussed in Subsection 8.4.

Moreover, we either flag or exclude results whose statistical reliability is doubtful. A neural-distinguisher experiment can be viewed as a binary hypothesis test between random data and cipher-generated data, with the reported validation accuracy providing an empirical estimate of the distinguisher's true success probability. Since each validation sample is classified either correctly or incorrectly, the number of correct classifications on v independent validation samples follows a binomial model. If the true accuracy is p, the empirical accuracy $\hat{p}$ has standard error $\sqrt{p(1-p)/v}$. Since $p(1-p) \leq 1/4$, this standard error is at most $1/(2\sqrt{v})$. Using the normal approximation to the binomial distribution, a worst-case two-sided 95% confidence interval is therefore approximately

$$\hat{p} \pm \frac{1.96}{2\sqrt{v}}.$$

This motivates requiring a validation set of size at least 10^5 for robust reporting of distinguisher accuracy. For $v = 10^5$, the worst-case standard error is approximately 0.16%, yielding a 95% confidence interval of about ±0.31%. This precision is sufficient for many comparisons involving sub-percent differences, while still indicating that very small accuracy gaps should be interpreted cautiously. In contrast, for $v = 10^3$, the worst-case standard error is about 1.6%, yielding a 95% confidence interval of about ±3.1%. We therefore regard results based on substantially smaller validation sets as statistically fragile. This is particularly important for accuracies close to random guessing: for example, a reported accuracy of 53% on 10^3 samples has an approximate confidence interval of $53\% \pm 3.1\%$, so the true accuracy may plausibly be close to the random baseline of 50%. Consequently, if the random-guessing baseline lies within the confidence interval, we treat

the claimed distinguishing advantage as statistically inconclusive and exclude the result from quantitative comparison. Similarly, we explicitly flag results that substantially deviate from comparable prior work when no sufficient explanation, validation, or methodological difference is provided.

To date, neural distinguishers have been applied to analyze 24 symmetric primitives. This comparative review provides a foundation for integrating new research into the existing body of work. While comprehensive tables compiling these analyses are provided in the appendices, we focus here on the most extensively studied primitives (namely SIMON, SIMECK, and SPECK), as only these provide sufficient data for meaningful comparative analysis of optimal approaches. The complete list of primitives analyzed to date is as follows: AES (Table 7), ARADI (Table 8), ASCON (Table 9), CHASKEY (Table 10), DES (Table 11), FF (Table 12), GIFT (Table 13), GIMLI (Table 14), GOST (Table 15), HIGHT (Table 16), KATAN (Table 17), KNOT (Table 18), LBCIoT (Table 19), LEA (Table 20), PRESENT (Table 21), PRIDE (Table 22), SHA3 (Table 23), SIMECK (Table 5), SIMON (Table 4), SKINNY (Table 24), SLIM (Table 25), SPECK (Table 6), TEA and XTEA (Table 26), and TinyJAMBU (Table 27).

In addition, Bose et al. [BPC24b] claim statistically significant distinguishers for 6-round SPARX and 8-round PICCOLO-80. Since theirs is the only work targeting these ciphers, and since their reported improvements for other ciphers challenge fundamental cryptographic principles, we omit dedicated tables that would lack meaningful contextual comparison. Instead, we include their distinguishing results for LEA, PRESENT, and HIGHT, together with a critical discussion. Similarly, we do not include the 7-round RC5 distinguisher by Pal et al. [PMDC22].

8.3.1 SIMON

SIMON is a family of AND-RX block ciphers, denoted SIMON-B/K, that encrypt blocks of size B with a key of size K. SIMON-32/64, SIMON-48/96, SIMON-64/128, and SIMON-128/256 have 32, 36, 44, and 72 rounds, respectively. Neural differential distinguishers have been developed for all versions of SIMON.

Table 4 provides an overview of differential neural distinguishers developed for the SIMON family of block ciphers. The most extensively studied variant is SIMON-32, for which multiple works explore different neural architectures and experimental settings. In the standard setting (2-1-CT-R), the best distinguisher reaches 11 rounds using an automated training pipeline and a generic convolutional architecture [BGHR23]. The same holds for SIMON-64. Overall, convolutional architectures consistently achieve the highest accuracy among neural distinguishers reported for SIMON.

Using multiple ciphertext pairs ($n = 16, 32, 64$) together with advanced feature-engineering techniques, as in [LRCL23, LLS+24, ZWW24], further improves distinguisher performance and extends the analysis to 12 rounds [LLS+24]. Similarly, in the related-key setting, distinguishers can be trained for up to 13 rounds [LLS+24]. The classification task becomes easier when distinguishing among ciphertext distributions induced by different input differences ($E = \mathrm{D}$), rather than distinguishing a single cipher-generated distribution from the uniform distribution ($E = \mathrm{R}$). This setting therefore leads to some of the highest-round neural distinguishers reported so far for SIMON-32 [WWS24b, WW24a].

For SIMON, several works use very large training datasets. For example, [BGL+22] obtain an 11-round distinguisher for SIMON32 using a four-stage training procedure with 10^7, 2^{28}, and $2 \cdot 2^{30}$ pairs in the later stages. Bellini et al. [BGHR23] propose a polishing step in which a neural distinguisher initially trained on 10^7 pairs is retrained with an additional 10^9 pairs. This polishing step is also used by Wang et al. [WWS24b]. Similarly, Zhang et al. [ZWW24] use staged training with $4 \cdot 10^7$ samples, each consisting of 16 pairs. Lu et al. [LLS+24] use different training strategies depending on the scenario: in some cases, they employ a staged approach with $8 \cdot (2 \cdot 10^7 + 2 \cdot 2^{25})$ pairs, while in others they

Table 4: Overview of the Neural Differential Distinguishers for SIMON.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SIMON-32/64	$\mathcal{ND}_{\text{Gohr}}$	2-1-A-R	20M	2M	-	8	0.834	[BGPT21]
	$\mathcal{ND}_{\text{Gohr}}$	4-3-CT-R	40M	4M	-	8	0.923	[WQW+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	9	0.5907	[HRC21c]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	9	0.6277	[SZM20]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	-	-	-	9	0.6320	[TH21]†
	$\mathcal{ND}_{\text{Gohr}}$	4-3-CT-R	40M	4M	-	9	0.6373	[SZM20]
	INC	8-2-A-D	80M	8M	-	9	0.886	[WWS24a]
	$\mathcal{ND}_{\text{Gohr}}$	64-1-δ-R	640M	6.4M	-	10	0.6109	[HRC21c]
	SE-ResNet	2-1-A-R	4852M	537M	-	11	0.517	[BGL+22]
	DBitNet	2-1-CT-R	2020M	2M	✓	11	0.518	[BGHR23]
	$\mathcal{ND}_{\text{Gohr}}$	64-1-A-R	640M	64M	-	11	0.6081	[LRCL23]
	DBitNet	2-2-CT-D	2020M	2M	✓	12	0.505	[WWS24b]
	SE-ResNet	16-1-A-R	320M	32M	-	12	0.5152	[LLS+24]
	INC	32-1-A-R	1280M	2M	-	12	0.5218	[ZWW24]†
SIMON-32/64^{RK}	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	11	0.5445	[EGP23]
	SE-ResNet	16-1-A-R	320M	32M	-	13	0.5262	[LLS+24]
	SE-ResNet	16-2-A-D	320M	32M	✓	13	0.567	[WW24a]
SIMON-48/96	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5789	[HRC21c]
	$\mathcal{ND}_{\text{Gohr}}$	96-1-δ-R	960M	9.6M	-	11	0.6143	[HRC21c]
	DBitNet	2-2-CT-D	20M	2M	✓	12	0.515	[WWS24b]
	$\mathcal{ND}_{\text{Gohr}}$	96-1-A-R	960M	96M	-	12	0.6159	[LRCL23]
SIMON-48/96^{RK}	SE-ResNet	16-2-A-D	320M	32M	✓	13	0.696	[WW24a]
SIMON-64/128	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	11	0.5972	[HRC21c]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	12.8M	-	12	0.6957	[HRC21c]
	DBitNet	2-1-CT-R	20M	2M	✓	13	0.518	[BGHR23]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-A-R	1280M	128M	-	13	0.701	[LRCL23]
	DBitNet	2-2-CT-D	20M	2M	✓	14	0.506	[WWS24b]
	SE-ResNet	16-1-A-R	1394M	134M	-	14	0.5185	[LLS+24]
SIMON-64/128^{RK}	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	13	0.5151	[EGP23]
	SE-ResNet	16-1-A-R	320M	32M	-	14	0.5788	[LLS+24]
	SE-ResNet	16-2-A-D	320M	32M	✓	14	0.618	[WW24a]
SIMON-128/256	DBitNet	2-1-CT-R	20M	2M	✓	20	0.507	[BGHR23]
SIMON-128/256^{RK}	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	16	0.5062	[EGP23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-). Best distinguishers for cipher variants with at least two distinguishers in the standard setting are highlighted.

† A critical discussion of these results is provided in the text.

RK Related key setting.

train directly on $2 \cdot 10^7 \cdot 8$ pairs. Thus, virtually all high-performing distinguishers for SIMON-32 rely either on very large datasets or on some form of staged training.

We conclude this section on SIMON with a critical discussion of [TH21, ZWW24]. Tian et al. [TH21] do not specify the size of their training or validation sets, making it impossible to assess the statistical reliability of their reported accuracy. In contrast, Zhang et al. [ZWW24] specify the size of their validation set; however, it contains only 63K samples. For $v = 63$K, the worst-case standard error is approximately 0.20%, giving a 95% confidence interval of roughly $\pm 0.39\%$.

8.3.2 SIMECK

SIMECK is a variant of SIMON using a key schedule similar to that of SPECK. SIMECK-32/64, SIMECK 48/96, and SIMECK-64/128 have 32, 36, and 44 rounds, respectively. Neural differential distinguishers have been developed for all versions of SIMECK.

Table 5 provides an overview of differential neural distinguishers developed for the SIMECK family of block ciphers. The most extensively studied variant is SIMECK-32, for which multiple works explore different neural network architectures. In the standard setting (2-1-CT-R), the best distinguisher reaches 10 rounds using the original convolutional architecture [LTZ22a]. Overall, convolutional architectures consistently achieve the highest accuracy among neural distinguishers reported for SIMECK.

Using multiple ciphertext pairs ($n = 16$) together with advanced feature-engineering techniques, as in [LLS+24, ZLWL23], further improves distinguisher performance and

Table 5: Overview of the Neural Differential Distinguishers for SIMECK.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SIMECK-32	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5407	[LTZ22b]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5438	[LTZ22a]
	$\mathcal{ND}_{\text{Gohr}}$	4-3-CT-R	67M	1M	-	11	0.5042	[WTZ+22]
	DBitNet	2-2-CT-D	2020M	2M	✓	12	0.505	[WWS24b]
	SE-ResNet	16-1-A-R	1394M	134M	-	12	0.5146	[LLS+24]
	INC	16-1-A-R	32320M	32M	-	12	0.5161	[ZLWL23]
SIMECK-32/64$^{\text{RK}}$	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	15	0.5134	[EGP23]
	SE-ResNet	16-1-A-R	320M	32M	-	15	0.5467	[LLS+24]
	SE-ResNet	16-2-A-D	320M	32M	✓	15	0.568	[WW24a]
SIMECK-32$^{\text{Unkeyed}}$	MLP	2-2-δ-D	66K	66K	-	9	0.526	[BBD+23]†
SIMECK-48/96	DBitNet	2-2-CT-D	20M	2M	✓	15	0.505	[WWS24b]
SIMECK-48/96$^{\text{RK}}$	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	17	0.5206	[EGP23]
	SE-ResNet	16-2-A-D	320M	32M	✓	19	0.523	[WW24a]
SIMECK-64/128	DBitNet	2-2-CT-D	20M	2M	✓	18	0.507	[WWS24b]
	SE-ResNet	16-1-A-R	1394M	134M	-	18	0.5218	[LLS+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R$^+$	20M	2M	-	20	0.5212	[EGP23]
SIMECK-64/128$^{\text{RK}}$	SE-ResNet	16-1-A-R	320M	32M	-	22	0.5180	[LLS+24]
	SE-ResNet	16-2-A-D	320M	32M	✓	22	0.526	[WW24a]
SIMECK-64$^{\text{Unkeyed}}$	MLP	2-2-δ-D	66K	66K	-	14	0.55	[BBD+23]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-). Best distinguishers for cipher variants with at least two distinguishers in the standard setting are highlighted.

† A critical discussion of these results is provided in the text.

$^{\text{RK}}$ Related key setting.

extends the analysis to 12 rounds [LLS+24]. In the related-key setting, distinguishers can be trained for up to 15 rounds [EGP23, LLS+24, WW24a]. The classification task becomes easier when distinguishing among ciphertext distributions induced by different input differences ($E = \text{D}$), rather than distinguishing a single cipher-generated distribution from the uniform distribution ($E = \text{R}$). This setting, therefore, leads to the highest-round neural distinguishers reported so far for related-key SIMECK-32 [WW24a].

For SIMECK, several works use very large training datasets. Zhang et al. [ZLWL23] use the staged training approach proposed by Gohr [Goh19], with $2 \cdot 10^7$ and $2 \cdot 10^9$ samples, each consisting of 8 pairs. Bellini et al. [BGHR23] propose a polishing step in which a neural distinguisher initially trained on 10^7 pairs is retrained with an additional 10^9 pairs. This polishing step is also used by Wang et al. [WWS24b]. Lu et al. [LLS+24] use different training strategies depending on the scenario: in some cases, they employ a staged approach with $8 \cdot (2 \cdot 10^7 + 2 \cdot 2^{25})$ pairs, while in others they train directly on $2 \cdot 10^7 \cdot 8$ pairs.

We conclude this section on SIMECK with a critical discussion of [BBD+23]. For $v =$ 33K ciphertext pairs in the validation set, the worst-case standard error is approximately 0.28%, giving a 95% confidence interval of roughly ±0.54%.

8.3.3 SPECK

SPECK is a family of ARX block ciphers, denoted as SPECK-B/K, designed to encrypt blocks of size B with a key of size K. The variants SPECK-32/64, SPECK-48/96, SPECK-64/128, SPECK-96/96, and SPECK-128/256 consist of 22, 23, 27, 29, and 34 rounds, respectively. SPECK is the cipher with the most published neural distinguishers to date and therefore serves as the primary benchmark. Neural differential distinguishers have been developed for all SPECK versions.

Table 6 provides an overview of the differential neural distinguishers developed for the SPECK family of block ciphers. In the standard setting (2-1-CT-R) for SPECK-32, Gohr's original 8-round distinguisher [Goh19] remained the main benchmark for several years. Bellini et al. [BGHR23] achieve comparable accuracy with an automated and cipher-

Table 6: Overview of the Neural Differential Distinguishers for SPECK.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SPECK-32	Quantum	2-1-CT-R	60K	2K	-	5	0.53	[KJL+23]†
	MLP	2-2-δ-D	33K	33K	-	5	0.5416	[BBDH21]†
	$\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$	2-1-δ_{tr}-R	10M	1M	-	6	0.673	[LRC23]
	MLP	2-1-δ_{tr}-R	20M	2M	-	6	0.688	[ERP22]
	MLP	2-1-δ-R	20M	2M	-	6	0.72	[ERP22]
	$\mathcal{ND}^{\text{ensmbl.}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	6	0.781	[HGH+23]
	$\mathcal{ND}_{\text{Gohr}}$	100-1-A-R	20M	2M	-	6	1	[BGPT21]†
	$\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.504	[PMDC22]
	DenseNet	2-1-CT-R	2M	2M	-	7	0.531	[SM24]†
	$\mathcal{ND}^{\text{pruned}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.596	[BBP22]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-δ-R	20M	2M	-	7	0.583	[BGPT21]
	$\mathcal{ND}_{\text{Gohr}}$	2-2-δ-D	20M	2M	-	7	0.599	[WWH21]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	-	✓	7	0.614	[WW21]†
	$\mathcal{ND}^{\text{attntn.}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.6169	[DCC23]
	$\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$	8-1-CT-R	80M	8M	-	7	0.6939	[LRC23]
	$\mathcal{ND}_{\text{Gohr}}$	16-1-CT-R	20M	2M	-	7	0.7009	[CSYY23]
	$\mathcal{ND}^{\text{attntn.}}_{\text{Gohr}}$	16-1-CT-R	160M	16M	-	7	0.728	[DCC23]
	INC	8-2-A-D	80M	8M	-	7	0.764	[WWS24a]
	INC	64-1-A-R	64M	6.4M	-	7	0.9713	[YW23]
	INC	2-1-CT-R	40M	2M	-	8	0.5135	[BLYZ23]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	4040M	2M	-	8	0.514	[Goh19]
	DBitNet	2-1-CT-R	2020M	2M	✓	8	0.514	[BGHR23]
	DBitNet	2-2-CT-D	2020M	2M	✓	8	0.519	[WWS24b]
	MLP	2-1-A-R	20M	2M	-	8	0.5208	[LJSC24a]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-A-R	1280M	128M	-	8	0.6502	[LRCL23]
	INC	32-1-A-R	1280M	2M	-	9	0.5045	[ZWW24]
SPECK-32 RK	$\mathcal{ND}_{\text{Gohr}}$	2-2-δ-D	20M	2M	-	7	0.559	[WWH21]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.576	[WWH21]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	9	0.5932	[TSL23]
	INC	2-1-CT-R	1342M	2M	-	10	0.5562	[BLYZ23]
SPECK-32 Unkeyed	MLP	2-2-δ-D	66K	66K	-	8	0.515	[BBD+23]†
SPECK-48	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	-	✓	6	0.726	[WW21]†
	DBitNet	2-2-CT-D	20M	2M	✓	8	0.506	[WWS24b]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-A-R	1280M	128M	-	8	0.5462	[LRCL23]
SPECK-64	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT$_{\text{tr}}$-R	20M	2M	-	6	0.662	[HLF+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	6	0.754	[HLF+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.623	[HLF+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	-	✓	7	0.632	[WW21]†
	INC	32-1-A-R	1280M	2M	-	7	0.641	[HLF+24]
	DBitNet	2-1-CT-R	20M	2M	✓	8	0.537	[BGHR23]
	DBitNet	2-2-CT-D	20M	2M	✓	8	0.559	[WWS24b]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	12.8M	-	8	0.632	[HRC21c]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-A-R	1280M	128M	-	8	0.7181	[LRCL23]
SPECK-96	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT$_{\text{tr}}$-R	20M	2M	-	7	0.681	[HLF+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.832	[HLF+24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	7	0.850‡	[CSY23]
SPECK-128	MLP	2-2-δ-D	33K	33K	-	7	0.8266	[BBDH21]†
	DBitNet	2-1-CT-R	20M	2M	✓	10	0.593	[BGHR23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-). Best distinguishers for cipher variants with at least two distinguishers in the standard setting are highlighted.

/ Unknown quantity.

† A critical discussion of these results is provided in the text.

RK Related key setting.

‡ In [CSY23], the accuracy of the teacher network for SPECK-96 was not given, but we were able to retrieve it by running the model from the authors' repository; we give the average of 10 runs, each with 10^6 samples.

† In [BBP22], the authors evaluated several pruned neural distinguishers; we report the smallest one, Gohr's $\mathcal{ND}_{\text{Gohr}}$ with depth 1, 7 channels removed from C1, 21 from C2, 25 from C3, 46 neurons from D1, and 36 from D2.

agnostic pipeline that is not specifically tailored to SPECK.[5] In general, convolutional

[5]Bao et al. [BLYZ23] state that "the simple training pipeline [of [BGHR23]] did not produce $\mathcal{ND}$s with the same accuracy as Gohr's on 8-round Speck32/64; it needs a further polishing step to achieve similar accuracy, demanding more time and data." This statement is only partially accurate. While the polishing step of [BGHR23] is indeed required to match Gohr's 8-round accuracy, it is a highly simplified, automated analog of Gohr's 8-round staged training procedure and therefore does not require more data than Gohr's original training scheme.

architectures consistently achieve the highest accuracy among neural network distinguishers for SPECK in various settings [BLYZ23, BGHR23, CSY23], with one surprising exception: The strongest reported 8-round SPECK-32 distinguisher in the standard setting is due to Lv et al. [LJSC24a].[6] Their work shows that carefully selected differential-linear features can outperform approaches based primarily on differential information. The method searches for low-Hamming-weight differential-linear approximations and selects the most informative ones using LGBM feature-importance scores. The resulting distinguishers achieve high accuracy and lower evaluation cost and can be viewed as a successful form of advanced feature engineering: rather than feeding raw ciphertext pairs to a convolutional architecture, a comparatively simple classifier, such as an MLP, is applied to structured differential-linear features. The single-pair distinguisher extends naturally to multiple pairs by evaluating these features on each pair, aggregating the results into empirical bias estimates, and making a single joint decision. Notably, their strongest 8-round distinguisher uses an input difference that differs substantially from those commonly used in the SPECK neural-cryptanalysis literature, and its training is neither staged nor based on a dataset as large as those used in comparable works. This provides an important data point, which we discuss in detail in Subsection 8.4.

Using multiple ciphertext pairs ($n = 32, 128$) together with advanced feature-engineering techniques further improves distinguisher performance beyond the best result in the standard setting [LRCL23, ZWW24], extending the analysis to 9 rounds [ZWW24]. In the related-key setting, distinguishers can be trained for up to 10 rounds [BLYZ23]. The classification task becomes easier when distinguishing among ciphertext distributions from different input differences ($E = \mathrm{D}$), rather than distinguishing a single cipher-generated distribution from the uniform distribution ($E = \mathrm{R}$). Hence, [WWS24b] slightly outperforms [BGHR23] for SPECK-32 and SPECK-64, despite using the same setup.

For the case of SPECK, several high-performing distinguishers rely on either large datasets, staged training, or both. In [Goh19], the training relies on a staged procedure, using $2 \cdot 10^7$ pairs for the lower-round distinguishers and an additional $2 \cdot 10^9$ pairs for the final 8-round model (2020M pairs). Similarly, in [BGHR23], the authors use a polishing step that retrains a distinguisher initially trained with 10^7 samples using an additional 10^9 fresh samples (1010M pairs). The authors of [BLYZ23] replace Gohr's staged training with a freezing-layer method: starting from a pretrained $(r-1)$-round distinguisher, they freeze all convolutional layers and retrain only the fully connected layers on r-round data using an additional 10^7 pairs. In the related-key setting, they generate a fresh training set of 2^{24} pairs in each of 20 epochs; applying the freezing-layer method therefore requires the same amount of data again for the retraining stage (671M pairs). Zhang et al. and Huang et al. [ZWW24, HLF+24] use a staged training approach with $4 \cdot 10^7$ samples, each containing 16 pairs (640M pairs). Wang and Wang [WWS24b] build upon the staged training procedure proposed in [BGHR23]. Since the validation-set size is not explicitly stated by the authors, we assume it is the same as in [BGHR23]. It is noteworthy that most of the high-performing distinguishers for SPECK-32 are trained either on very large datasets or using some form of staged training.

We conclude this section on SPECK with a critical discussion of [BGPT21, BBDH21, BBD+23, KJL+23, LJSC24a, SM24, WW21]. Wang et al. [WW21] do not specify the size of their validation set, making it impossible to assess the statistical reliability of their reported accuracy. In [KJL+23] the authors use a validation set of only 1K ciphertext pairs. The standard error is about 1.6%, giving a two-sided 95% confidence interval of roughly ±3.1%. Hence, we regard the claimed distinguishing advantage as statistically inconclusive and exclude this result from quantitative comparison. In [BBDH21, BBD+23], Baksi et al. use validation sets of only 16K and 33K ciphertext pairs, respectively. For

[6] The discussion of multi-pair neural distinguishers in [LJSC24a] should be interpreted with care: input-size limitations do not imply that multi-pair distinguishers require a different neural architecture, since Gohr [GLN22] shows how to obtain multi-pair distinguishers from single-pair ones by score aggregation.

$v = 16$K, the worst-case standard error is about 0.40%, giving a two-sided 95% confidence interval of roughly ±0.77%. For $v = 33$K, the worst-case standard error is about 0.28%, giving a two-sided 95% confidence interval of roughly ±0.54%. In [BGPT21], the authors use a validation set of only 20K samples. For $v = 20$K, the worst-case standard error is about 0.35%, giving a two-sided 95% confidence interval of roughly ±0.69%. Sajwan et al. [SM24] report an accuracy of 53.1% (round 7) on 2M training, respectively validation samples, and provide a comparison in which DenseNet outperforms $\mathcal{ND}_{\text{Gohr}}$. Given such a small number of training samples, both networks exhibit severe overfitting, and the authors themselves call the result only "marginal."

8.4 Discussion

Based on our comprehensive assessment of research in neural differential cryptanalysis (Subsection 5.1), we identify several promising directions and critical challenges that merit further investigation. Our analysis focuses primarily on thoroughly vetted cryptographic primitives, specifically SIMON (Table 4), SIMECK (Table 5), SPECK (Table 6).

Critical Review Overall, 35 entries in our tables are highlighted due to methodological concerns, including validation sets that are too small, confidence intervals that, in some cases, include random guessing, and other issues affecting comparability. Our survey is intended to serve as a reference point for future work and to identify results that cannot be reliably compared under due diligence standards. We hope that these critical assessments help future studies improve their experimental design by choosing and reporting appropriate training and validation set sizes, and by further investigating highly unlikely results, such as full-round attacks against modern, previously unbroken block ciphers, or highly unusual validation-accuracy patterns.

Network Architectures (N) Findings on optimal neural network architectures for cryptanalysis remain mixed. Several works, including [BBCD21, MPKM+22], report that MLPs outperform convolutional architectures; similar claims have also been made for LSTMs [SSL+22]. In contrast, our results show that convolutional architectures consistently perform best, with 7 out of 8 top distinguishers in the standard setting relying on convolutions. This observation is plausible, given DCNNs' strong feature-extraction capabilities across many domains, particularly in image recognition. Once relevant features are explicitly identified, however, classical or simpler neural models can achieve performance comparable to more complex architectures [BGPT21]. This is also illustrated by [LJSC24a], who obtain a state-of-the-art distinguisher for 8-round SPECK32 using a simple MLP after extracting relevant differential-linear features. A similar extensive feature extraction step was performed by [BGPT21]. Thus, one of the main reasons to use a convolutional architecture is its black-box feature-extraction capability.

We further observe that, when $\mathcal{ND}_{\text{Gohr}}$ is used, the kernel size remains one of the most sensitive hyperparameters and must be tuned carefully for each cipher [GLN22]. As more generic alternatives, DBitNet and INC can be used: the former relies on dilated convolutions, while the latter combines multiple kernel sizes. Both approaches reduce the need for manual kernel-size tuning.

A meaningful comparison between concrete approaches remains difficult because performance is influenced by many factors beyond architecture alone. These include the number of ciphertext samples, the choice of input differences, the sophistication of feature-engineering techniques, and variations in experimental design (including the choice of hyperparameters). This complexity highlights the need for comprehensive benchmarking studies to establish more definitive conclusions about optimal architectures and training strategies, a challenge we address in detail in Subsection 10.1.

Unlike natural language processing, neural cryptanalysis has not consistently benefited from the "bigger is better" scaling paradigm described by Kaplan et al.[KMH+20]. Existing research has not conclusively shown that deeper or wider neural architectures reliably improve distinguishing performance in cryptographic settings. On the contrary, Gohr [Goh19] uses shallower architectures for distinguishers targeting near-uniform ciphertext distributions, specifically for 7- and 8-round SPECK encryption. Moreover, Buacuieti et al. [BBP22] show that the resulting neural distinguisher remains only sparsely activated. Overall, current architectures appear to struggle when ciphertext distributions approach uniformity, even when these distributions are provably non-uniform. Learning such weak statistical deviations may require more sophisticated modeling approaches, potentially including mechanisms that support memorization or retrieval of rare informative patterns, which we examine in detail in Subsection 10.2.

Multi-Pair Distinguishers ($n > 2$) Neural distinguishers that process multiple ciphertext pairs simultaneously have historically shown only limited practical advantages over simple score aggregation based on a single-pair distinguisher [GLN22]. At the same time, they substantially increase training costs: larger inputs typically require larger architectures, and the expanded input space demands significantly larger datasets to avoid overfitting. However, this may not reflect an inherent limitation of multi-pair distinguishers, but rather a limitation of the prevailing training setup, in which plaintext pairs are typically sampled independently. For example, one could imagine that neural distinguishers trained on structured plaintext sets, rather than independently sampled pairs, might be able to exploit additional inter-pair relations. Whether such structured multi-pair distinguishers can reliably outperform single-pair distinguishers by learning genuine inter-pair features remains an open question.

Surprisingly, our comprehensive analysis shows that, for SPECK, SIMON, and SIMECK, multi-pair distinguishers successfully break one additional round compared to their single-pair counterparts. This suggests that multi-pair inputs may provide a measurable advantage during training, beyond what can be achieved through post-hoc score aggregation alone. Since single-pair distinguishers fail to train in these settings, multi-pair distinguishers may be inherently beneficial for learning extremely weak distinguishing signals.

Interestingly, similar ideas have been widely studied in the machine learning community, in particular under the name of Multiple Instance Learning [DLLP97] (MIL), but the corresponding techniques have so far not been applied at all in the context of neural distinguishers. A typical benchmark for MIL is the Elephant dataset [ATH02], where samples are groups of images, with a positive label if the group contains an elephant and a negative label otherwise. This problem mirrors the case of high rounds neural distinguishers, where most pairs are not helpful, but rare pairs that follow a 'good' differential pattern (the elephants) determine the label. Recent approaches to the MIL problem, such as [ITW18], appear promising directions for improving multi-pair classifiers. Similarly, the problem of anomaly detection has received considerable attention in the machine learning community; if we treat the 'elephant pair' as an anomaly within an otherwise unremarkable distribution, adapting approaches such as Deep One-Class Classification [RGD+18] could yield interesting results. Finally, the Deep Set framework [KATT20] considers functions of sets and addresses issues such as permutation invariance, which are relevant to multiple pair classification, for which the order of the pairs has no importance.

This evolving effectiveness of multi-pair architectures represents a significant development and offers a promising direction for future cryptanalytic research; however, this line of work has so far largely ignored the substantial body of work in the deep learning community, and we believe there is significant room for improvement by incorporating these techniques.

Multiple Input Differences ($m > 1$) Gohr et al. [GLN22] establish an important relationship between neural differential distinguishers and statistical distance: the accuracy of a neural distinguisher correlates with the statistical distance between the ciphertext-difference distributions being separated, particularly in the common setting where encryption is distinguished from a uniform distribution, i.e., $E = \mathrm{R}$. Consequently, distinguishers tend to perform better for input differences that induce more clearly separable output distributions.

It is therefore unsurprising that distinguishers perform better when trained to separate two ciphertext-pair distributions induced by two carefully selected input differences, i.e., in the $E = \mathrm{D}$ setting. This trend is consistent across all block ciphers considered in our analysis. Despite this advantage, however, integrating multiple-difference approaches into practical key-recovery attacks remains challenging. The attack framework pioneered by Gohr [Goh19] fundamentally relies on distinguishing real ciphertext distributions from uniform ones. Adapting this framework to exploit the additional statistical power of multiple input differences would likely require substantial modifications to the underlying cryptanalytic methodology. Consequently, before further investigating distinguishers in this particular setting, researchers should first identify concrete key-recovery scenarios in which this specific distinguishing capability can be effectively exploited.

One further promising research direction is to exploit structural relationships between differential characteristics by switching bits for adjoining differentials (SBfADs) [BGL+22]. For multi-difference distinguishers, the requirements could be relaxed to require conformance to any one output difference, rather than requiring all differentials to share identical output differences.

Feature Engineering (T) The impact of feature engineering remains debated. Several works, such as [BGL+22, LLS+24], report improvements from engineered input representations. However, Gohr et al.[GLN22] show that, with careful hyperparameter selection, neural distinguishers can learn key-independent feature transformations in a black-box manner. Interestingly, among the best distinguishers in the standard setting, only the one by [LJSC24a] combines feature engineering with an MLP architecture, a model class shown to be unable to learn relevant feature extraction on its own [BGPT21]. This suggests that carefully designed feature engineering can be highly effective, particularly when paired with architectures that lack strong intrinsic feature-extraction capabilities.

Results that achieve higher accuracies than these standard-setting distinguishers typically also use more plaintexts, i.e., $n > 2$, making it difficult to determine whether the observed improvement is due to feature engineering or simply to the additional data available to the distinguisher. For comparability, it is therefore essential to first demonstrate the benefit of a proposed feature-engineering method in the standard setting (and possibly even using a baseline architecture such as $\mathcal{ND}_{\mathrm{Gohr}}$). Otherwise, based on our data, one must assume that feature engineering has at most a limited impact, and possibly no measurable impact at all. Comprehensive benchmarking frameworks would provide valuable insight into the relative merits of automated and engineered feature extraction for cryptanalytic applications; we discuss this challenge further in Subsection 10.1.

The explainability analyses of [Goh19, BGPT21, BLYZ23] show that Gohr's architecture $\mathcal{ND}_{\mathrm{Gohr}}$ exploits information beyond purely differential features. Consequently, feature-engineering approaches that restrict the input representation to the output difference, i.e., $F = \delta$, may yield distinguishers that are inherently less competitive. This observation, however, applies only to certain classes of ciphers [GLN22]. Whether such additional features are relevant should therefore be verified for the target cipher before adopting a restricted feature representation.

(Staged) Training When applied, as in the case of SPECK32 [Goh19, BGHR23, BLYZ23] and SIMON32 [BGL+22, BGHR23], staged training appears to yield substantial improvements in accuracy and, in some cases, even in the number of distinguishable rounds. This seems to hold particularly when no advanced feature engineering is used, as in [LJSC24a].

Both handcrafted feature extraction and the standard staged training pipeline [Goh19] require considerable expert knowledge. For this reason, the more generic pipeline proposed in [BGHR23] or [BLYZ23] may provide a useful alternative for obtaining strong distinguishers with less manual intervention. More generally, larger datasets can be expected to approximate the underlying ciphertext distribution more accurately, especially when the distribution of ciphertext pairs is close to uniform. A staged training pipeline that uses massive datasets of size 10^9 allows even very small statistical deviations to be represented in the training data and potentially learned by the network.

Conclusion Overall, our analysis suggests that progress in neural differential cryptanalysis depends less on simply increasing model size and more on carefully matching the learning setup to the statistical structure of the target cipher. As a practical default, researchers should start from strong convolutional baselines, in particular $\mathcal{ND}_{\text{Gohr}}$, DBitNet, or INC, since convolutional architectures dominate the best distinguishers in the standard setting and provide useful black-box feature-extraction capabilities. However, larger or deeper networks should not be assumed to automatically improve performance. Instead, sensitive hyperparameters, especially kernel size in $\mathcal{ND}_{\text{Gohr}}$-based models, should be tuned carefully for each cipher.

Multi-pair distinguishers are among the most promising directions for future work. Although they are more expensive to train and have historically offered limited advantages over score aggregation, our results show that they can extend the number of distinguishable rounds for SPECK, SIMON, and SIMECK. Structured plaintext sets, Multiple Instance Learning, Deep Sets, and anomaly-detection methods provide natural methodological starting points for designing more principled multi-pair distinguishers.

By contrast, distinguishers trained to separate distributions arising from multiple input differences should be pursued with some caution. While the $E = D$ setting often yields higher distinguishing accuracy, its practical relevance for key recovery remains unclear. Before investing further effort into this setting, future work should first identify concrete key-recovery frameworks that can exploit this specific distinguishing capability.

Feature engineering should be evaluated under stricter experimental controls. Claims of improvement should first be demonstrated in the standard setting, ideally against a strong convolutional baseline and with the same number of plaintexts. Otherwise, improvements may be attributable to additional data rather than to the engineered features themselves. Moreover, feature representations restricted to ciphertext differences, such as $\mathrm{F}=\delta$, should be adopted only after verifying that the target cipher does not permit additional information. Feature engineering based on differential-linear masks has proven highly effective and should therefore be explored further in future work.

Finally, staged training appears to be a highly effective yet underexplored component of successful neural distinguishers for many ciphers. Its benefit is especially plausible for high-round settings, where ciphertext-pair distributions are extremely close to uniform, and very large datasets are needed to expose small statistical deviations. Future work should therefore systematically apply staged training, preferably using a generic pipeline, to obtain the best possible distinguishers.

Comparing the various works on Neural Differential Cryptanalysis has been challenging due to differences in experimental parameters, such as the choice of attacked block ciphers and simultaneous changes to multiple factors, including input data and model architecture. These variations make it difficult to attribute performance differences to a single variable, even when reporting results up to the standards discussed in Section 9. Instead, we

primarily rely on patterns observed across studies to identify the most effective approaches. Such insights are only possible through a large-scale analysis of the entire domain. To simplify this process in the future, we propose a benchmarking challenge in Subsection 10.1, in which experiments are conducted under a shared experimental setup. This approach mirrors the long-standing practice in machine learning of using standardized benchmarks to evaluate and compare methods, as exemplified by ImageNet [DDS+09] for object recognition. This would strengthen validation of new methods and make it feasible to determine whether a proposed technique outperforms the state of the art. Furthermore, a standardized dataset would improve reproducibility by reducing dependence on the authors' codebase.

Key Insights Overall, in the standard setting, strong convolutional baselines such as $\mathcal{ND}_{\text{Gohr}}$, DBitNet, and INC remain the most reliable starting points, provided that they are trained on sufficiently large datasets ($\geq 10^7$ pairs). In contrast, smaller datasets, simpler architectures such as MLPs, and feature representations based solely on ciphertext differences have consistently led to weaker performance. More complex pipelines that utilize larger models, elaborate feature engineering, or additional samples do not automatically yield improvements and should therefore be evaluated through fair comparisons against the existing baselines, including aggregated variants where applicable. Among feature-engineering approaches, differential-linear features have shown clear benefits. For extending the number of distinguishable rounds, staged training and multi-pair distinguishers appear particularly promising.

9 Neural Distinguisher Training: Best Practices

Neural network training is not a deterministic process: it is subject to significant variations in the outcome that are caused, for example, by the (random) network parameter initialization process, and the batch process of training data and corresponding differing movement through the optimization plane. Further, the chosen hyperparameters and neural network architectures heavily influence the training outcome.

To correctly interpret the success of neural network training, it is important to carefully distinguish between training, validation, and test data. Each dataset plays an important role: the *training data* is used to compute the model's loss and update its parameters. However, the goal of neural network training is not good performance (low loss) on known data, but instead, generalization to previously unseen data. To monitor the model's performance on previously unseen data during training, *validation data* is used.

A commonly observed phenomenon during neural network training is *overfitting*. At some point during training, the model stops learning new generalizable features of the training data and instead uses its parameters to memorize the training dataset. This leads to an increasing validation loss. Instead of using the model trained for the maximum number of epochs, one better uses the model with the minimum *validation data loss*. However, the validation data has now been used in model optimization and can no longer be used to characterize performance based on previously unseen data. Fresh *test data* should be used for the final characterization instead.

The number of parameters of a deep neural network *does not* straightforwardly relate to its computational training cost. Instead, it depends on the computations required by the particular layers used in the network model. The computational training cost should be measured in terms of the required number of FLOPs (floating point operations) or MACs (multiply-accumulate operations). Popular deep learning libraries such as TensorFlow and PyTorch provide routines to obtain neural network parameter counts as well as FLOPs.[7]

[7]The performance of libraries for training neural distinguishers has been compared in [BBD+23].

For example, FLOPs can be evaluated with the TensorFlow Keras module `keras-flops`, and the TensorFlow native routine `model.count_params()` provides the parameter count.

Though not unique to neural differential cryptanalysis, these best practices were frequently overlooked in papers during our literature review, underscoring the importance of emphasizing these standards.

Commonly Overlooked Best Practices for Neural Distinguisher Training

1. **Results Reporting I:** Clearly indicate the results obtained on training, validation, and test datasets and the size of each dataset.
2. **Results Reporting II:** Denote accuracy (or any other metrics) with error margins on multiple sets of *freshly generated* test data.
3. **Neural Network Reporting:** Indicate the network's memory requirements using FLOPs and the number of neural network parameters, and training time per epoch on the specific computational environment (e.g., number and type of GPUs or CPUs).
4. **Open Reproducibility:** Publish the code and trained model parameters to enable review, replication, and future comparisons.

10 Future Challenges

10.1 Benchmarking Challenge

As the field of neural cryptanalysis grows, it is becoming more difficult to compare different works on a given primitive due to significant variability in the architectures used, training regimes, distinguishing experiments, or feature engineering. To better understand neural distinguishers, we see the creation of a benchmarking platform as an important challenge in the medium term. The goal of such a platform would be to compare neural architectures submitted by authors across standard problem sets and display them on a leaderboard. This objective, however, is not straightforward, and we discuss some friction points below.

Defining Problems A problem can be defined as an n-m-T-E configuration, a primitive, a training pipeline, and a dataset size. A logical first step would be to evaluate all models on the initial SPECK32 problem in the 2-1-CT-R setting to identify the top-performing architectures.

Training regimes are critical: Gohr's work [Goh19] required an advanced pipeline with pre-training on likely differences followed by re-training with 100× more samples to reach 8 rounds. Subsequent research often employs similar polishing techniques. This creates a distinction between raw performance (training from scratch under consistent conditions) and enhanced accuracy (using pretraining [Goh19], layer freezing [GLN22], previous-round distinguisher retraining [BR21], or increased final-round samples).

Sample quantity also matters. Many works follow Gohr's approach [Goh19] (10^7 training, 10^6 test samples), as reduction significantly impacts performance. Multiple-pair sample approaches [BBCD21] pose comparison challenges: fixing the sample count gives unfair advantages to models that see more pairs, while fixing the pair count may disadvantage models trained on fewer samples (e.g., 10^7 pairs per sample would mean training on a single sample). Distinguishers across several n categories could be compared using the score proposed in [GLN22] to verify performance beyond score aggregation.

Metrics The first challenge in comparing different models is defining what to compare. As of now, the main metrics used to compare neural distinguishers are accuracy, true positive rate, true negative rate, and, more recently [BGHR23], the number of floating point

operations (FLOPS), which impacts the training time and quantifies the time complexity of the inference part in a key recovery attack. In the Deep Learning community, the EfficientNet framework [TL19], which proposes techniques to scale neural networks based on inference speed or parameter count constraints, is often used as a baseline for new models. For neural distinguishers, we could similarly use the number of parameters and the FLOPs ratio relative to the original architecture from Gohr to provide context for the obtained accuracy. However, we also need dedicated metrics adapted to the specific use cases of neural cryptanalysis. In particular, the current metrics do not provide much information on the key recovery complexity, which largely depends on the *wrong key response profile* (see Subsection 4.3), prepended differentials, and neutral bits.

10.2 AI-$\mathcal{ND}$ Challenge

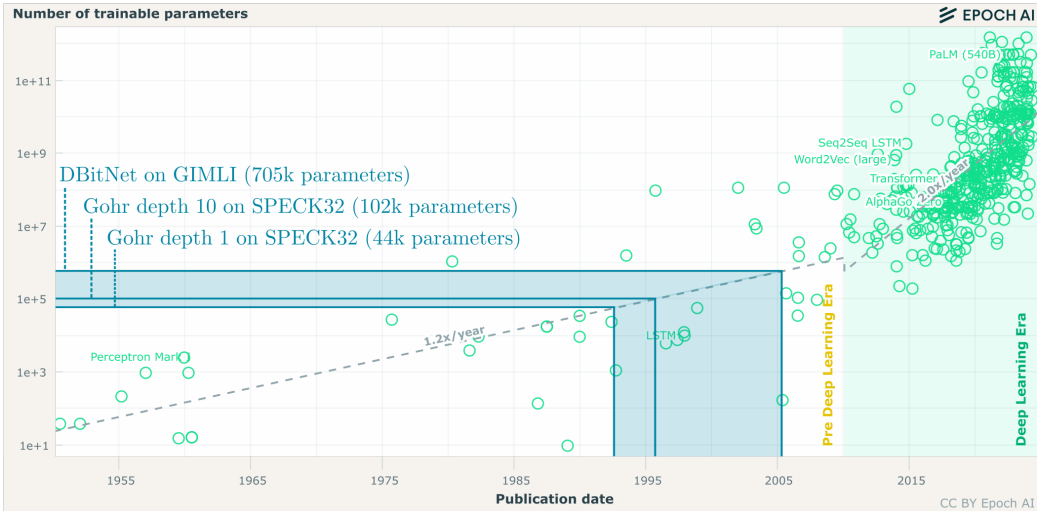


Figure 4: Adapted from [Epo23] with added data for Gohr's $\mathcal{ND}_{\text{Gohr}}$ on Table 6, and DBitNet on Table 14 from [BGHR23, Table 5].

The neural network architectures currently employed in Neural Differential Cryptanalysis have origins that trace back several years. For instance, the Inception Module, introduced by Google researchers in a seminal paper in 2014 [SLJ+15], was used. Similarly, Kaiming He et al. [HZRS16] won the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2015 using ResNet. Attention was introduced in "Attention is all you need" at NeurIPS 2017 [VSP+17], and Squeeze-and-Excitation Networks at CVPR 2018 [HSS18].

In recent years, deeper and more complex models have led to higher parameter counts. Figure 4 illustrates the general trend of the increasing parameter count in deep learning models. This is particularly evident in the case of Large Language Models (LLMs), such as GPT, which contain billions of parameters. The deep learning models used to date in Neural Differential Cryptanalysis have lower parameter counts than more modern "Deep Learning Era" models. Challenges when increasing the model parameter count are higher computational load, longer training times, and overfitting.

However, the advancement of AI technologies, such as transformers and reinforcement learning, coupled with increased computational power, holds significant potential to enhance cryptographic neural differential distinguishers. Transformers, with their capability to handle long-range dependencies and their effectiveness in capturing complex patterns, offer a robust framework for analyzing cryptographic data. Reinforcement learning, on

the other hand, provides a powerful approach for optimizing neural network performance through iterative feedback and learning from interactions. When applied to cryptographic neural differential distinguishers, these advanced AI methodologies can yield more accurate models. The increased computational power available today enables training deeper, more complex networks that can explore a larger hypothesis space and uncover subtle cryptographic weaknesses that simpler models might miss.

Up until now, cryptographers have mainly attempted to apply AI models. As illustrated in Subsection 10.1, a leaderboard with cryptographically meaningful metrics should be established. Based on the existence of transparent metrics, the **AI-$\mathcal{ND}$ Challenge** aims at (i) motivating cryptographers to use more advanced AI technologies, but also at (ii) motivating cryptographers to establish an AI-competition[8] to allow AI researchers and engineers to apply state-of-the-art methods to Neural Differential Cryptanalysis.

10.3 Combining Classical and ML-aided Cryptanalysis

Given the current limitations of Neural Differential Cryptanalysis, in particular the small number of rounds for which neural distinguishers can typically be trained and their inability, for certain cipher classes, to learn information beyond the ciphertext-pair difference distribution, neural techniques are unlikely to consistently outperform classical cryptanalysis in the near future. Indeed, Gohr's neural key-recovery attack on SPECK32 can no longer be considered state of the art, as it was later surpassed in time complexity by a classical attack [BdST+23].

However, this does not imply that Neural Differential Cryptanalysis is generally ineffective. Rather, it provides a distinct set of tools that can complement classical methods and may offer advantages in specific settings. Researchers should therefore continue working on these open limitations while, in the meantime, using Neural Differential Cryptanalysis in combination with classical attacks. Its main strengths are its ability to rapidly identify differential or integral distinguishers with limited expert knowledge, efficiently approximate ciphertext-pair difference distributions, and extract previously unknown features, such as in [Goh19, BLYZ23, ZYS+26], which can then be exploited in classical attacks.

11 Conclusions

In this paper, we perform a systematic review of the follow-ups to Gohr's seminal paper on neural distinguishers. In the process, we identify and classify works focusing on training neural distinguishers. This systematic review uncovered a young yet vast body of research and a need for common methodological guidelines to grow the field, which we attempt to provide. We also identified three future challenges: comparing neural distinguisher results using standardized benchmarks, scaling up to much larger, more ambitious architectures, and combining classical and ML-aided cryptanalysis.

Over the past six years, multiple new settings have been explored for differential cryptanalysis, using multiple pairs per sample or polytopic differences, with the same or varied keys across samples. In addition, various feature engineering techniques, particularly partial inversion, have been explored. These address the question of what clues we can give the neural distinguisher, and multiple avenues are left to explore in that direction. But more fundamentally, what matters perhaps more is what question we ask the neural distinguisher, given this clue, or, said differently, what task we ask the neural network to perform. So far, much of the literature has focused on differential-based properties for a single pair and a single input difference, but many variations could be built, as well as tasks related to different types of cryptanalysis or entirely new distinguishing experiments.

[8] Small AI-competitions are hosted on platforms such as Kaggle, while large AI-competitions include the *"Makrikadis"* time series forecasting competition [MSA20], or ILSVRC [RDS+15].

References

[AA16] Martín Abadi and David G. Andersen. Learning to protect communications with adversarial neural cryptography. *CoRR*, abs/1610.06918, 2016. doi:10.48550/arXiv.1610.06918.

[AAEK23] Hicham Tahiri Alaoui, Ahmed Azouaoui, and Jamal El Kafi. Artificial neural networks cryptanalysis of merkle-hellman knapsack cryptosystem. In Janusz Kacprzyk, Mostafa Ezziyyani, and Valentina Emilia Balas, editors, *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 196–205, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-26384-2_18.

[ABA22] Abdullah F. Al-Aboosi, Matan Broner, and Fadhil Y. Al-Aboosi. Bingo: A semi-centralized password storage system. *J. Cybersecur. Priv.*, 2(3):444–465, 2022. doi:10.3390/JCP2030023.

[AEA17] Wasan Awad and El-Sayed M El-Alfy. Computational intelligence in cryptology. In *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, pages 1636–1652. IGI Global Scientific Publishing, 2017. doi:10.4018/978-1-5225-1759-7.ch065.

[Ago19] Agohr. Deep SPECK implementation. https://github.com/agohr/deep_speck, 2019. GitHub repository for deep_speck.

[AI-23a] AI-Lab-Y. NASA implementation. https://github.com/AI-Lab-Y/NASA, 2023. GitHub repository for CSY23.

[AI-23b] AI-Lab-Y. ND_mc implementation. https://github.com/AI-Lab-Y/ND_mc, 2023. GitHub repository for CSYY23.

[AI-24] AI-Cipher-Sicurity. Code for HLF+24. https://github.com/AI-Cipher-Sicurity/code, 2024. GitHub repository for HLF+24.

[Ali23] Alin-dl. DeepKatan implementation. https://github.com/alin-dl/deepkatan, 2023. GitHub repository for LLHC23.

[Ano21] AnonymousSubmissionEuroCrypt2021. A deeper look at machine-learning-based cryptanalysis (implementation). https://github.com/AnonymousSubmissionEuroCrypt2021/A-Deeper-Look-at-Machine-Learning-Based-Cryptanalysis, 2021. GitHub repository for BGPT21.

[ASC22] ASC8384. Neural distinguishers (code). https://github.com/ASC8384/Neural-Distinguishers, 2022. GitHub repository for SSL+22.

[ATH02] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 561–568. MIT Press, 2002. URL: https://dl.acm.org/doi/10.5555/2968618.2968690.

[AVS+24] Indrakanti Aishwarya, Lakshmy Koduvayur Viswanathan, Chungath Srinivasan, Girish Mishra, Saibal K. Pal, and M. Sethumadhavan. Improving the security of the LCB block cipher against deep learning-based attacks. *Cryptogr.*, 8(4):55, 2024. doi:10.3390/CRYPTOGRAPHY8040055.

[Bak21] Anubhab Baksi. Classical and physical security of symmetric key cryptographic algorithms. In *29th IFIP/IEEE International Conference on Very Large Scale Integration, VLSI-SoC 2021, Singapore, Singapore, October 4-7, 2021*, pages 1–2. IEEE, 2021. doi:10.1109/VLSI-SOC53125.2021.9606988.

[BBC+26] Anubhab Baksi, Jakub Breier, Anupam Chattopadhyay, Tomas Gerlich, Sylvain Guilley, Naina Gupta, Takanori Isobe, Arpan Jati, Petr Jedlicka, Hyunjun Kim, Fukang Liu, Zdenek Martinasek, Kosei Sakamoto, Hwajeong Seo, and Rentaro Shiba. BAKSHEESH: similar yet different from GIFT (and ZORRO). *IACR Commun. Cryptol.*, 2(4):31, 2026. doi:10.62056/AE890LMOL.

[BBCD21] Anubhab Baksi, Jakub Breier, Yi Chen, and Xiaoyang Dong. Machine learning assisted differential distinguishers for lightweight ciphers. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2021, Grenoble, France, February 1-5, 2021*, pages 176–181. IEEE, 2021. doi:10.23919/DATE51398.2021.9474092.

[BBD+23] Anubhab Baksi, Jakub Breier, Vishnu Asutosh Dasu, Xiaolu Hou, Hyunji Kim, and Hwajeong Seo. New results on machine learning-based distinguishers. *IEEE Access*, 11:54175–54187, 2023. doi:10.1109/ACCESS.2023.3270396.

[BBDH21] Anubhab Baksi, Jakub Breier, Vishnu Asutosh Dasu, and Xiaolu Hou. Machine learning attacks on speck. *Security and Implementation of Lightweight Cryptography (SILC)*, pages 1–6, 2021. URL: https://www.esat.kuleuven.be/cosic/events/silc2020/wp-content/uploads/sites/4/2021/09/Submission10.pdf.

[BBP22] Nicoleta-Norica Bacuieti, Lejla Batina, and Stjepan Picek. Deep neural networks aiding cryptanalysis: A case study of the speck distinguisher. In Giuseppe Ateniese and Daniele Venturi, editors, *Applied Cryptography and Network Security - 20th International Conference, ACNS 2022, Rome, Italy, June 20-23, 2022, Proceedings*, volume 13269 of *Lecture Notes in Computer Science*, pages 809–829. Springer, 2022. doi:10.1007/978-3-031-09234-3_40.

[BdST+23] Alex Biryukov, Luan Cardoso dos Santos, Je Sen Teh, Aleksei Udovenko, and Vesselin Velichkov. Meet-in-the-filter and dynamic counting with applications to speck. In Mehdi Tibouchi and Xiaofeng Wang, editors, *Applied Cryptography and Network Security - 21st International Conference, ACNS 2023, Kyoto, Japan, June 19-22, 2023, Proceedings, Part I*, volume 13905 of *Lecture Notes in Computer Science*, pages 149–177. Springer, 2023. doi:10.1007/978-3-031-33488-7_6.

[BFG+24] Emanuele Bellini, Mattia Formenti, David Gérault, Juan Grados, Anna Hambitzer, Yun Ju Huang, Paul Huynh, Mohamed Rachidi, Raghvendra Rohit, and Sharwan K. Tiwari. Claasping ARADI: automated analysis of the ARADI block cipher. In Sourav Mukhopadhyay and Pantelimon Stanica, editors, *Progress in Cryptology - INDOCRYPT 2024 - 25th International Conference on Cryptology in India, Chennai, India, December 18-21, 2024, Proceedings, Part II*, volume 15496 of *Lecture Notes in Computer Science*, pages 90–113. Springer, 2024. doi:10.1007/978-3-031-80311-6_5.

[BG11] Céline Blondeau and Benoît Gérard. Multiple differential cryptanalysis: Theory and practice. In Antoine Joux, editor, *Fast Software Encryption -*

18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers, volume 6733 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2011. doi:10.1007/978-3-642-21702-9_3.

[BGG+23] Emanuele Bellini, David Gérault, Juan Grados, Yun Ju Huang, Rusydi H. Makarim, Mohamed Rachidi, and Sharwan K. Tiwari. CLAASP: A cryptographic library for the automated analysis of symmetric primitives. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *Selected Areas in Cryptography - SAC 2023 - 30th International Conference, Fredericton, Canada, August 14-18, 2023, Revised Selected Papers*, volume 14201 of *Lecture Notes in Computer Science*, pages 387–408. Springer, 2023. doi:10.1007/978-3-031-53368-6_19.

[BGHR23] Emanuele Bellini, David Gérault, Anna Hambitzer, and Matteo Rossi. A cipher-agnostic neural training pipeline with automated finding of good input differences. *IACR Trans. Symmetric Cryptol.*, 2023(3):184–212, 2023. doi:10.46586/TOSC.V2023.I3.184-212.

[BGL+22] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Enhancing differential-neural cryptanalysis. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 318–347. Springer, 2022. doi:10.1007/978-3-031-22963-3_11.

[BGPT21] Adrien Benamira, David Gérault, Thomas Peyrin, and Quan Quan Tan. A deeper look at machine learning-based cryptanalysis. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 805–835. Springer, 2021. doi:10.1007/978-3-030-77870-5_28.

[BHR+22] Emanuele Bellini, Anna Hambitzer, Matteo Rossi, et al. A survey on machine learning applied to symmetric cryptanalysis. *RENDICONTI DEL SEMINARIO MATEMATICO*, 80:107–122, 2022. doi:11583/2985015.

[BJS24] BJSeok. InputDiffExplorePCA (code). https://github.com/BJSeok/InputDiffExplorePCA/tree/master, 2024. GitHub repository for SCL24.

[BLY23] BLYZ23 Authors. BLYZ23 code archive. https://www.dropbox.com/scl/fo/tznsz3xz30q6jh2gqvp6i/ANnxmHWgLxhXWr63_PKvkYo?rlkey=oiae3k7p7h601cs0dxyk0gatk&dl=0, 2023. Code archive (Dropbox folder) for BLYZ23.

[BLYZ23] Zhenzhen Bao, Jinyu Lu, Yiran Yao, and Liu Zhang. More insight on deep learning-aided cryptanalysis. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 436–467. Springer, 2023. doi:10.1007/978-981-99-8727-6_15.

[Boa24a] Sara Boancă. Exploring patterns and assessing the security of pseudo-random number generators with machine learning. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART*, pages 186–193. INSTICC, SciTePress, 2024. doi:10.5220/0012312900003636.

[Boa24b] Sara Boancă. Optimizations for learning from linear feedback shift register variations with artificial neural networks. In Ilias Maglogiannis, Lazaros S. Iliadis, John MacIntyre, Markos Avlonitis, and Antonios Papaleonidas, editors, *Artificial Intelligence Applications and Innovations - 20th IFIP WG 12.5 International Conference, AIAI 2024, Corfu, Greece, June 27-30, 2024, Proceedings, Part IV*, volume 714 of *IFIP Advances in Information and Communication Technology*, pages 197–210. Springer, 2024. doi:10.1007/978-3-031-63223-5_15.

[BP22] Carlo Brunetta and Pablo Picazo-Sanchez. Modelling cryptographic distinguishers using machine learning. *J. Cryptogr. Eng.*, 12(2):123–135, 2022. doi:10.1007/S13389-021-00262-X.

[BPC24a] Amrita Bose, Debranjan Pal, and Dipanwita Roy Chowdhury. Cryptographic distinguishers through deep learning for lightweight block ciphers. In V. S. Shankar Sriram, Anila Glory H., Gang Li, and Shiva Raj Pokhrel, editors, *Applications and Techniques in Information Security - 14th International Conference, ATIS 2024, Tamil Nadu, India, November 22-24, 2024, Proceedings*, volume 2306 of *Communications in Computer and Information Science*, pages 47–63. Springer, 2024. doi:10.1007/978-981-97-9743-1_4.

[BPC24b] Amrita Bose, Debranjan Pal, and Dipanwita Roy Chowdhury. Deep learning-based differential distinguishers for cryptographic sequences. In Sourav Mukhopadhyay and Pantelimon Stanica, editors, *Progress in Cryptology - INDOCRYPT 2024 - 25th International Conference on Cryptology in India, Chennai, India, December 18-21, 2024, Proceedings, Part II*, volume 15496 of *Lecture Notes in Computer Science*, pages 114–133. Springer, 2024. doi:10.1007/978-3-031-80311-6_6.

[BR21] Emanuele Bellini and Matteo Rossi. Performance comparison between deep learning-based and conventional cryptographic distinguishers. In Kohei Arai, editor, *Intelligent Computing - Proceedings of the 2021 Computing Conference, Volume 3, SAI 2021, Virtual Event, 15-16 July, 2021*, volume 285 of *Lecture Notes in Networks and Systems*, pages 681–701. Springer, 2021. doi:10.1007/978-3-030-80129-8_48.

[Bru21] Carlo Brunetta. *Cryptographic Tools for Privacy Preservation*. PhD thesis, Department of Computer Science & Engineering, Chalmers University of Technology, Gothenburg, Sweden, 2021. URL: https://research.chalmers.se/publication/524966/file/524966_Fulltext.pdf.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990. doi:10.1007/3-540-38424-3_1.

[BSL21] K. V. L. Bhargavi, Chungath Srinivasan, and K. V. Lakshmy. Panther: A sponge based lightweight authenticated encryption scheme. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, *Progress in Cryptology - INDOCRYPT 2021 - 22nd International Conference on Cryptology in India, Jaipur, India, December 12-15, 2021, Proceedings*, volume 13143 of *Lecture Notes in Computer Science*, pages 49–70. Springer, 2021. doi:10.1007/978-3-030-92518-5_3.

[BTCS+15] Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. The simon and speck lightweight block ciphers. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2015. doi:10.1145/2744769.2747946.

[Can23] CangXiXi. A deep learning aided differential distinguisher improvement framework with more lightweight and universal (code). https://github.com/CangXiXi/A-Deep-Learning-aided-Differential-Distinguisher-Improvement-Framework, 2023. GitHub repository for LRC23.

[CdOAB+18] Murilo Coutinho, Robson de Oliveira Albuquerque, Fábio Borges, Luis Javier García Villalba, and Tai-Hoon Kim. Learning perfectly secure cryptography to protect communications with adversarial neural cryptography. *Sensors*, 18(5):1306, 2018. doi:10.3390/S18051306.

[CDS22] Luan Cardoso Dos Santos. *Design, Cryptanalysis and Protection of Symmetric Encryption Algorithms*. PhD thesis, Universite Du Luxembourg, The Faculty of Science, Technology and Medicine, 2022. URL: https://orbilu.uni.lu/bitstream/10993/52955/1/Thesis-PostDefense.pdf.

[CLC12] Jung-Wei Chou, Shou-De Lin, and Chen-Mou Cheng. On the effectiveness of using state-of-the-art machine learning techniques to launch cryptographic distinguishing attacks. In Ting Yu, V. N. Venkatakrishan, and Apu Kapadia, editors, *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence, AISec 2012, Raleigh, NC, USA, October 19, 2012*, pages 105–110. ACM, 2012. doi:10.1145/2381896.2381912.

[Cry21] Crypto-TII. Deep_learning_vs_conventional_distinguishers. https://github.com/Crypto-TII/deep_learning_vs_conventional_distinguishers, 2021. GitHub repository for BR21.

[Cry23a] CryptAnalystDesigner. Differential neural cryptanalysis of Simeck32 (code). https://github.com/CryptAnalystDesigner/Differential-Neural-Cryptanalysis-Simeck32, 2023. GitHub repository for ZLWL23.

[Cry23b] CryptAnalystDesigner. Multiple-cipher distinguisher for DES, Chaskey, and PRESENT (code). https://github.com/CryptAnalystDesigner/MutipleCipherDesChaskeyPresent, 2023. GitHub repository for ZWC23.

[Cry23c] Crypto-TII. NNBits implementation. https://github.com/Crypto-TII/nnbits, 2023. GitHub repository for HGH+23.

[Cry23d] Crypto-TII. The CLAASP framework. https://github.com/Crypto-TII/claasp, 2023. GitHub repository for BGG+23 (CLAASP framework).

[Cry23e] CryptoUSM. deep-lbc-slim (code). https://github.com/CryptoUSM/deep-lbc-slim, 2023. GitHub repository for TTJ23.

[Cry24a] CryptAnalystDesigner. Differential neural cryptanalysis on AES (code). https://github.com/CryptAnalystDesigner/differential-neural-cryptanalysis-on-aes, 2024. GitHub repository for ZWL24.

[Cry24b] CryptAnalystDesigner. Neural distinguisher with inception (code). https://github.com/CryptAnalystDesigner/NeuralDistingsuisherWithInception, 2024. GitHub repository for ZWW24.

[CS21] Bang Yuan Chong and Iftekhar Salam. Investigating deep learning approaches on the security analysis of cryptographic algorithms. *Cryptogr.*, 5(4):30, 2021. doi:10.3390/CRYPTOGRAPHY5040030.

[CSIR02] Julio César Hernández Castro, José María Sierra, Pedro Isasi, and Arturo Ribagorda. Genetic cryptoanalysis of two rounds TEA. In Peter M. A. Sloot, Chih Jeng Kenneth Tan, Jack J. Dongarra, and Alfons G. Hoekstra, editors, *Computational Science - ICCS 2002, International Conference, Amsterdam, The Netherlands, April 21-24, 2002. Proceedings, Part III*, volume 2331 of *Lecture Notes in Computer Science*, pages 1024–1031. Springer, 2002. doi:10.1007/3-540-47789-6_108.

[CSY23] Yi Chen, Yantian Shen, and Hongbo Yu. Neural-aided statistical attack for cryptanalysis. *Comput. J.*, 66(10):2480–2498, 2023. doi:10.1093/COMJNL/BXAC099.

[CSYY23] Yi Chen, Yantian Shen, Hongbo Yu, and Sitong Yuan. A new neural distinguisher considering features derived from multiple ciphertext pairs. *Comput. J.*, 66(6):1419–1433, 2023. doi:10.1093/COMJNL/BXAC019.

[DCC23] Haoran Deng, Xianghui Cao, and Yu Cheng. Attention in differential cryptanalysis on lightweight block cipher SPECK. In *20th Annual International Conference on Privacy, Security and Trust, PST 2023, Copenhagen, Denmark, August 21-23, 2023*, pages 1–9. IEEE, 2023. doi:10.1109/PST58708.2023.10320201.

[DCW23] Yibin Deng, Jiale Chen, and Jun Wang. An image compression encryption based on the semi-tensor product and the dft measurement matrix. *Optik*, 288:171175, 2023. doi:10.1016/j.ijleo.2023.171175.

[DDK+23] Itai Dinur, Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Efficient detection of high probability statistical properties of cryptosystems via surrogate differentiation. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 98–127. Springer, 2023. doi:10.1007/978-3-031-30634-1_4.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. doi:10.1109/CVPR.2009.5206848.

[Dif22a] Differential-Neural. An assessment of differential neural distinguishers (code). https://github.com/differential-neural/An-Assessment-of-Differential-Neural-Distinguishers, 2022. GitHub repository for "An Assessment of Differential Neural Distinguishers".

[Dif22b] Differential-Neural-Cryptanalysis. Implementation for SPECK32 and SIMON32. https://github.com/differential-neural-cryptanalysis/speck32_simon32, 2022. GitHub repository for BGL+22.

[Dif23] Differentialdistinguisher. Investigating and Enhancing the Neural Distinguisher (code). https://github.com/differentialdistinguisher/ND_interpretability, 2023. GitHub repository for "Investigating and Enhancing the Neural Distinguisher for Differential Cryptanalysis".

[Dif24a] Differentialdistinguisher. AutoND_New (code). https://github.com/differentialdistinguisher/AutoND_New, 2024. GitHub repository for WWS24.

[Dif24b] Differentialdistinguisher. CD_ND (code). https://github.com/differentialdistinguisher/CD_ND, 2024. GitHub repository for WW24b.

[Dif24c] Differentialdistinguisher. Enhanced related-key differential neural distinguishers (code). https://github.com/differentialdistinguisher/Enhanced-related-key-differential-neural-distinguishers, 2024. GitHub repository for WW24a.

[DLLP97] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. doi:10.1016/S0004-3702(96)00034-3.

[DM23] Shivank Kumar Dadhwal and Girish Mishra. Machine learning-based classification between block cipher and stream cipher. In S. Smys, Khaled A. Kamel, and Ram Palanisamy, editors, *Inventive Computation and Information Technologies*, pages 531–542, Singapore, 2023. Springer Nature Singapore. doi:10.1007/978-981-19-7402-1_38.

[DNS24] Jimmy Dani, Kalyan Nakka, and Nitesh Saxena. Breaking indistinguishability with transfer learning: A first look at SPECK32/64 lightweight block ciphers. *CoRR*, abs/2405.19683, 2024. doi:10.48550/ARXIV.2405.19683.

[DRD22] DRDOINTERNSHIP. DeepLearning_Cryptograhy implementation. https://github.com/DRDOINTERNSHIP/DeepLearning_Cryptograhy, 2022. GitHub repository for RRSM22.

[DS25] Brooke Dalton and Mark Stamp. Classifying world war II era ciphers with machine learning. *Cryptologia*, 49(2):97–127, 2025. doi:10.1080/01611194.2024.2304888.

[DZF+22] Ming Duan, Rui Zhou, Chaohui Fu, Sheng Guo, and Qianqiong Wu. Vulnerability testing on the key scheduling algorithm of present using deep learning. In Wenbo Shi, Xiaofeng Chen, and Kim-Kwang Raymond Choo, editors, *Security and Privacy in New Computing Environments*, pages 307–318, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-030-96791-8_23.

[EAZD23] Zahra Ebadi Ansaroudi, Rocco Zaccagnino, and Paolo D'Arco. On pseudorandomness and deep learning: A case study. *Applied Sciences*, 13(5), 2023. doi:10.3390/app13053372.

[EGP23] Amirhossein Ebrahimi, David Gérault, and Paolo Palmieri. Deep learning-based rotational-xor distinguishers for AND-RX block ciphers: Evaluations on simeck and simon. In Claude Carlet, Kalikinkar Mandal, and Vincent

Rijmen, editors, *Selected Areas in Cryptography - SAC 2023 - 30th International Conference, Fredericton, Canada, August 14-18, 2023, Revised Selected Papers*, volume 14201 of *Lecture Notes in Computer Science*, pages 429–450. Springer, 2023. doi:10.1007/978-3-031-53368-6_21.

[ELR20] Maria Eichlseder, Gregor Leander, and Shahram Rasoolzadeh. Computing expected differential probability of (truncated) differentials and expected linear potential of (multidimensional) linear hulls in SPN block ciphers. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 345–369. Springer, 2020. doi:10.1007/978-3-030-65277-7_16.

[ElS21] Muhammad ElSheikh. *MILP-aided Cryptanalysis of Some Block Ciphers*. PhD thesis, Concordia University, 2021. URL: https://spectrum.library.concordia.ca/id/eprint/988518/.

[Epo23] Epoch AI. Announcing epoch ai's updated parameter, compute and data trends database, 2023. Accessed: July 20, 2026. URL: https://epoch.ai/latest/announcing-updated-pcd-database.

[ERP22] Amirhossein Ebrahimi, Francesco Regazzoni, and Paolo Palmieri. Reducing the cost of machine learning differential attacks using bit selection and a partial ml-distinguisher. In Guy-Vincent Jourdan, Laurent Mounier, Carlisle M. Adams, Florence Sèdes, and Joaquín García-Alfaro, editors, *Foundations and Practice of Security - 15th International Symposium, FPS 2022, Ottawa, ON, Canada, December 12-14, 2022, Revised Selected Papers*, volume 13877 of *Lecture Notes in Computer Science*, pages 123–141. Springer, 2022. doi:10.1007/978-3-031-30122-3_8.

[Ess23] Bernhard Esslinger. *Learning and Experiencing Cryptography with CrypTool and SageMath*. Artech House Computer Security. Artech House Publishers, unabridged edition, 2023.

[FAAQ24] Ahmed Fanfakh, Nihad Abduljalil, and Ali Kadhum M. Al-Qurabat. Parallel multi-core implementation of the optimized Speck cipher. *International Journal of Safety & Security Engineering*, 14(3):843–852, 2024. doi:10.18280/ijsse.140316.

[FLW+23] Zhuohui Feng, Ye Luo, Chao Wang, Qianqian Yang, Zhiquan Liu, and Ling Song. Improved differential cryptanalysis on SPECK using plaintext structures. In Leonie Simpson and Mir Ali Rezazadeh Baee, editors, *Information Security and Privacy - 28th Australasian Conference, ACISP 2023, Brisbane, QLD, Australia, July 5-7, 2023, Proceedings*, volume 13915 of *Lecture Notes in Computer Science*, pages 3–24. Springer, 2023. doi:10.1007/978-3-031-35486-1_1.

[Fuk80] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. doi:10.1007/BF00344251.

[GA19] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, Jun. 2019. doi:10.1609/aimag.v40i2.2850.

[GHRS26] David Gérault, Anna Hambitzer, Eyal Ronen, and Adi Shamir. Deep neural cryptography. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology - EUROCRYPT 2026 - 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Rome, Italy, May 10-14, 2026, Proceedings, Part VI*, volume 16546 of *Lecture Notes in Computer Science*, pages 513–543. Springer, 2026. doi:10.1007/978-3-032-25333-0_18.

[GHZ+18] Aidan N. Gomez, Sicong Huang, Ivan Zhang, Bryan M. Li, Muhammad Osama, and Lukasz Kaiser. Unsupervised cipher cracking using discrete gans. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=BkeqO7x0-.

[GJS20a] Aron Gohr, Sven Jacob, and Werner Schindler. Efficient solutions of the CHES 2018 AES challenge using deep residual neural networks and knowledge distillation on adversarial examples. *IACR Cryptol. ePrint Arch.*, 2020:165, 2020. URL: https://eprint.iacr.org/2020/165.

[GJS20b] Aron Gohr, Sven Jacob, and Werner Schindler. Subsampling and knowledge distillation on adversarial examples: New techniques for deep learning based side channel evaluations. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 567–592. Springer, 2020. doi:10.1007/978-3-030-81652-0_22.

[GLN22] Aron Gohr, Gregor Leander, and Patrick Neumann. An assessment of differential-neural distinguishers. *IACR Cryptol. ePrint Arch.*, 2022:1521, 2022. URL: https://eprint.iacr.org/2022/1521.

[GLP+24] Yue-Tian Goi, Shu-Min Leong, Raphaël C.-W. Phan, Shangqi Lai, and Ana Salagean. Unveiling the black box: Neural cryptanalysis with XAI. In *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2024, Kuching, Malaysia, October 6-10, 2024*, pages 1951–1956. IEEE, 2024. doi:10.1109/SMC54092.2024.10831564.

[Goh19] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 150–179. Springer, 2019. doi:10.1007/978-3-030-26951-7_6.

[Goh22] Aron Gohr. Brute force cryptanalysis. *IACR Cryptol. ePrint Arch.*, 2022:53, 2022. URL: https://eprint.iacr.org/2022/053.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ASCON. *IACR Trans. Symmetric Cryptol.*, 2021(3):102–136, 2021. doi:10.46586/TOSC.V2021.I3.102-136.

[Gre17] Sam Greydanus. Learning the enigma with recurrent neural networks. *CoRR*, abs/1708.07576, 2017. arXiv:1708.07576.

[GVWT21] David Gunning, Eric Vorm, Jennifer Yunyan Wang, and Matt Turek. Darpa's explainable ai (xai) program: A retrospective. *Applied AI Letters*, 2(4):e61, 2021. doi:10.1002/ail2.61.

[GZAL22] Hicham Grari, Khalid Zine-Dine, Ahmed Azouaoui, and Siham Lamzabi. Deep learning-based cryptanalysis of a simplified AES cipher. *Int. J. Inf. Secur. Priv.*, 16(1):1–16, 2022. doi:10.4018/IJISP.300325.

[Hal22] Roger A. Hallman. Poster evegan: Using generative deep learning for cryptanalysis. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 3355–3357. ACM, 2022. doi:10.1145/3548606.3563493.

[HBLC25] Zezhou Hou, Zhenzhen Bao, Jinyu Lu, and Shaozhen Chen. Observations on the bayesiankeysearch with applications to simon and simeck. *IACR Trans. Symmetric Cryptol.*, 2025(3):755–799, 2025. doi:10.46586/TOSC.V2025.I3.755-799.

[HCM+24] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: A review on explainable artificial intelligence. *Cogn. Comput.*, 16(1):45–74, 2024. doi:10.1007/S12559-023-10179-8.

[HGG20] Benjamin Hettwer, Stefan Gehrer, and Tim Güneysu. Applications of machine learning techniques in side-channel attacks: a survey. *J. Cryptogr. Eng.*, 10(2):135–162, 2020. doi:10.1007/S13389-019-00212-8.

[HGH+23] Anna Hambitzer, David Gérault, Yun Ju Huang, Najwa Aaraj, and Emanuele Bellini. Nnbits: Bit profiling with a deep learning ensemble based distinguisher. In Mike Rosulek, editor, *Topics in Cryptology - CT-RSA 2023 - Cryptographers' Track at the RSA Conference 2023, San Francisco, CA, USA, April 24-27, 2023, Proceedings*, volume 13871 of *Lecture Notes in Computer Science*, pages 493–523. Springer, 2023. doi:10.1007/978-3-031-30872-7_19.

[HK00] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. doi:10.1080/00401706.2000.10485983.

[HLF+24] Tianrong Huang, Yingying Li, Qinggan Fu, Yincen Chen, and Ling Song. Improving differential-neural cryptanalysis for large-state SPECK. In Sokratis K. Katsikas, Christos Xenakis, Christos Kalloniatis, and Costas Lambrinoudakis, editors, *Information and Communications Security - 26th International Conference, ICICS 2024, Mytilene, Greece, August 26-28, 2024, Proceedings, Part I*, volume 15056 of *Lecture Notes in Computer Science*, pages 40–57. Springer, 2024. doi:10.1007/978-981-97-8798-2_3.

[HLG+23] Ying Huang, Lang Li, Ying Guo, Yu Ou, and Xiantong Huang. An efficient differential analysis method based on deep learning. *Comput. Networks*, 224:109622, 2023. doi:10.1016/J.COMNET.2023.109622.

[HLLL24] Ying Huang, Lang Li, Di Li, and Yongchao Li. Iabc: A neural integral distinguisher for and-rx ciphers. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 47(1):1–15, 2024. doi:10.3233/JIFS-238122.

[HLZH25] Yemao Hu, Lang Li, Siqi Zhu, and Zhiwen Hu. Enhancing neural distinguishers with partial difference bits leakage. *Internet Things*, 29:101438, 2025. doi:10.1016/J.IOT.2024.101438.

[HLZW20] Botao Hou, Yongqiang Li, Haoyue Zhao, and Bin Wu. Linear attack on round-reduced DES using deep learning. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II*, volume 12309 of *Lecture Notes in Computer Science*, pages 131–145. Springer, 2020. doi:10.1007/978-3-030-59013-0_7.

[HRC21a] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Cryptanalysis of round-reduced SIMON32 based on deep learning. *IACR Cryptol. ePrint Arch.*, 2021:362, 2021. URL: https://eprint.iacr.org/2021/362.

[HRC21b] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Improve neural distinguisher for cryptanalysis. *IACR Cryptol. ePrint Arch.*, 2021:1017, 2021. URL: https://eprint.iacr.org/2021/1017.

[HRC21c] ZeZhou Hou, JiongJiong Ren, and ShaoZhen Chen. Improve neural distinguishers of simon and speck. *Security and Communication Networks*, 2021(1):9288229, 2021. doi:10.1155/2021/9288229.

[HRC21d] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Sat-based method to improve neural distinguisher and applications to SIMON. *IACR Cryptol. ePrint Arch.*, 2021:452, 2021. URL: https://eprint.iacr.org/2021/452.

[HRC23] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Practical attacks of round-reduced SIMON based on deep learning. *Comput. J.*, 66(10):2517–2534, 2023. doi:10.1093/COMJNL/BXAC102.

[HSS18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7132–7141. Computer Vision Foundation / IEEE Computer Society, 2018. doi:10.1109/CVPR.2018.00745.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.90.

[ITW18] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2132–2141. PMLR, 2018. URL: http://proceedings.mlr.press/v80/ilse18a.html.

[ITY25] Mohamed Fadl Idris, Je Sen Teh, and Mohd Najwadi Yusoff. Diffgen: a data-driven framework for generating truncated differentials. *Appl. Intell.*, 55(5):329, 2025. doi:10.1007/S10489-025-06248-0.

[ITYY21] Mohamed Fadl Idris, Je Sen Teh, Jasy Liew Suet Yan, and Wei-Zhu Yeoh. A deep learning approach for active s-box prediction of lightweight generalized feistel block ciphers. *IEEE Access*, 9:104205–104216, 2021. doi:10.1109/ACCESS.2021.3099802.

[JIN24] JIN-smile. Improved related-key differential-based neural distinguishers (code). https://github.com/JIN-smile/Improved-Related-key-Differential-based-Neural-Distinguishers, 2024. GitHub repository for LLS+24.

[JKM21] Aayush Jain, Varun Kohli, and Girish Mishra. Deep learning based differential distinguisher for lightweight block ciphers. *CoRR*, abs/2112.05061, 2021. arXiv:2112.05061.

[JM24] Ongee Jeong and Inkyu Moon. Deep learning-based hash function cryptanalysis. In *15th International Conference on Information and Communication Technology Convergence, ICTC 2024, Jeju Island, Republic of Korea, October 16-18, 2024*, pages 1302–1303. IEEE, 2024. doi:10.1109/ICTC62082.2024.10826852.

[KATT20] Shivam Kalra, Mohammed Adnan, Graham W. Taylor, and Hamid R. Tizhoosh. Learning permutation invariant representations using memory networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pages 677–693. Springer, 2020. doi:10.1007/978-3-030-58526-6_40.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL: http://arxiv.org/abs/1412.6980.

[KEI+22] Hayato Kimura, Keita Emura, Takanori Isobe, Ryoma Ito, Kazuto Ogawa, and Toshihiro Ohigashi. Output prediction attacks on block ciphers using deep learning. In Jianying Zhou, Sridhar Adepu, Cristina Alcaraz, Lejla Batina, Emiliano Casalicchio, Sudipta Chattopadhyay, Chenglu Jin, Jingqiang Lin, Eleonora Losiouk, Suryadipta Majumdar, Weizhi Meng, Stjepan Picek, Jun Shao, Chunhua Su, Cong Wang, Yury Zhauniarovich, and Saman A. Zonouz, editors, *Applied Cryptography and Network Security Workshops - ACNS 2022 Satellite Workshops, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, SiMLA, Rome, Italy, June 20-23, 2022, Proceedings*, volume 13285 of *Lecture Notes in Computer Science*, pages 248–276. Springer, 2022. doi:10.1007/978-3-031-16815-4_15.

[KEI+23] Hayato Kimura, Keita Emura, Takanori Isobe, Ryoma Ito, Kazuto Ogawa, and Toshihiro Ohigashi. A deeper look into deep learning-based output prediction attacks using weak SPN block ciphers. *J. Inf. Process.*, 31:550–561, 2023. doi:10.2197/IPSJJIP.31.550.

[Khj23] Khj1594012. Quantum neural distinguisher for SPECK32 (code). https://github.com/khj1594012/quantum-neural-distinguisher-for-speck32, 2023. GitHub repository for KJL+23.

[KJK+24] Dukyoung Kim, Kyoungbae Jang, Hyunji Kim, Yujin Oh, Seyoung Yoon, and Hwajeong Seo. Cs-net: A deep learning-based analysis for the cryptography and steganography. In Jongsung Kim, Jungsoo Park, and Wai-Kong Lee, editors, *Information Security and Cryptology - ICISC 2024 - 27th International Conference, Seoul, South Korea, November 20-22, 2024, Revised Selected Papers*, volume 15596 of *Lecture Notes in Computer Science*, pages 244–257. Springer, 2024. doi:10.1007/978-981-96-5566-3_13.

[KJL+22] Hyunji Kim, Kyungbae Jang, Sejin Lim, Yeajun Kang, Won-Woong Kim, and Hwajeong Seo. Quantum neural network based distinguisher for differential cryptanalysis on simplified block ciphers. *IACR Cryptol. ePrint Arch.*, 2022:1671, 2022. URL: https://eprint.iacr.org/2022/1671.

[KJL+23] Hyunji Kim, Kyungbae Jang, Sejin Lim, Yeajun Kang, Won-Woong Kim, and Hwajeong Seo. Quantum neural network based distinguisher on SPECK-32/64. *Sensors*, 23(12):5683, 2023. doi:10.3390/S23125683.

[KKJ+24] Dukyoung Kim, Hyunji Kim, Kyungbae Jang, Seyoung Yoon, and Hwajeong Seo. Deep-learning-based neural distinguisher for format-preserving encryption schemes ff1 and ff3. *Electronics*, 13(7), 2024. doi:10.3390/electronics13071196.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall/CRC, 1 edition, 2007.

[KLJW23] Man Kang, Yongqiang Li, Lin Jiao, and Mingsheng Wang. Differential analysis of arx block ciphers based on an improved genetic algorithm. *Chinese Journal of Electronics*, 32(2):225–236, 2023. doi:10.23919/cje.2021.00.415.

[KLK+23] Hyunji Kim, Sejin Lim, Yeajun Kang, Wonwoong Kim, Dukyoung Kim, Seyoung Yoon, and Hwajeong Seo. Deep-learning-based cryptanalysis of lightweight block ciphers revisited. *Entropy*, 25(7), 2023. doi:10.3390/e25070986.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. arXiv:2001.08361.

[KMS02] Alexander Klimov, Anton Mityagin, and Adi Shamir. Analysis of neural cryptography. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 288–298. Springer, 2002. doi:10.1007/3-540-36178-2_18.

[Kop20] Nils Kopal. Of ciphers and neurons - detecting the type of ciphers using artificial neural networks. In Beáta Megyesi, editor, *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020, Budapest, Hungary, June 15-17, 2020*, Linköping Electronic Conference Proceedings, pages 77–86. Linköping University Electronic Press, 2020. doi:10.3384/ECP2020171011.

[KS22] S. K. Karthika and Kunwar Singh. Theoretical and deep learning based analysis of biases in salsa 128 bits. In Ilsun You, Hwankuk Kim, and Pelin

Angin, editors, *Mobile Internet Security - 6th International Symposium, MobiSec 2022, Jeju, South Korea, December 15-17, 2022, Revised Selected Papers*, volume 1644 of *Communications in Computer and Information Science*, pages 147–164. Springer, 2022. doi:10.1007/978-981-99-4430-9_11.

[KSJS21] Hyunji Kim, Gyeongju Song, Kyoungbae Jang, and Hwajeong Seo. Cryptanalysis of caesar using quantum support vector machine. *IACR Cryptol. ePrint Arch.*, 2021:1058, 2021. URL: https://eprint.iacr.org/2021/1058.

[KVD+25] Benjamin D. Kim, Vipindev Adat Vasudevan, Rafael G. L. D'Oliveira, Alejandro Cohen, Thomas Stahlbuhk, and Muriel Médard. Cryptanalysis via machine learning based information theoretic metrics. *CoRR*, abs/2501.15076, 2025. doi:10.48550/ARXIV.2501.15076.

[KY21] Manoj Kumar and Tarun Yadav. MILP based differential attack on round reduced WARP. In Lejla Batina, Stjepan Picek, and Mainack Mondal, editors, *Security, Privacy, and Applied Cryptography Engineering - 11th International Conference, SPACE 2021, Kolkata, India, December 10-13, 2021, Proceedings*, volume 13162 of *Lecture Notes in Computer Science*, pages 42–59. Springer, 2021. doi:10.1007/978-3-030-95085-9_3.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

[LCLH22] Dongdong Lin, Shaozhen Chen, Manman Li, and Zezhou Hou. The construction and application of (related-key) conditional differential neural distinguishers on KATAN. In Alastair R. Beresford, Arpita Patra, and Emanuele Bellini, editors, *Cryptology and Network Security - 21st International Conference, CANS 2022, Dubai, United Arab Emirates, November 13-16, 2022, Proceedings*, volume 13641 of *Lecture Notes in Computer Science*, pages 203–224. Springer, 2022. doi:10.1007/978-3-031-20974-1_10.

[LF24] Ronniel D Labio and Enrique Festijo. Neural network–based cryptanalysis of present and d-present block ciphers. In *Proceedings of the 9th International Cryptology and Information Security Conference 2024*, pages 110–117, 2024. URL: http://psasir.upm.edu.my/id/eprint/119577/1/119577.pdf#page=120.

[Li24] Facai Li. Mobilenet-based neural differential distinguishers for speck, gift and katan. In *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, pages 1334–1337, 2024. doi:10.1109/EIECC64539.2024.10929185.

[LJSC24a] Guangqiu Lv, Chenhui Jin, Zhen Shi, and Ting Cui. Approximating neural distinguishers using differential-linear imbalance. *J. Supercomput.*, 80(19):26865–26889, 2024. doi:10.1007/S11227-024-06375-4.

[LJSC24b] Guangqiu Lv, Chenhui Jin, Zhen Shi, and Ting Cui. Unveiling the neutral difference and its automated search. *IET Inf. Secur.*, 2024:1–15, 2024. doi:10.1049/2024/2939486.

[LKE+21] Ernst Leierzopf, Nils Kopal, Bernhard Esslinger, Harald Lampesberger, and Eckehard Hermann. A massive machine-learning approach for classical cipher type detection using feature engineering. In Carola Dahlke,

editor, *Proceedings of the 4th International Conference on Historical Cryptology, HistoCrypt 2021, Amsterdam, The Netherlands, postponed to June 20-22, 2022*, Linköping Electronic Conference Proceedings, pages 111–120. Linköping University Electronic Press, 2021. doi:10.3384/ECP183164.

[LLHC23] Dongdong Lin, Manman Li, Zezhou Hou, and Shaozhen Chen. Conditional differential analysis on the KATAN ciphers based on deep learning. *IET Inf. Secur.*, 17(3):347–359, 2023. doi:10.1049/ISE2.12099.

[LLL+21] Guozhen Liu, Jingwen Lu, Huina Li, Peng Tang, and Weidong Qiu. Preimage attacks against lightweight scheme xoodyak based on deep learning. In Kohei Arai, editor, *Advances in Information and Communication*, pages 637–648, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-73103-8_45.

[LLS+24] Jinyu Lu, Guoqiang Liu, Bing Sun, Chao Li, and Li Liu. Improved (related-key) differential-based neural distinguishers for SIMON and SIMECK block ciphers. *Comput. J.*, 67(2):537–547, 2024. doi:10.1093/COMJNL/BXAC195.

[LMK+21] Ernst Leierzopf, Vasily Mikhalev, Nils Kopal, Bernhard Esslinger, Harald Lampesberger, and Eckehard Hermann. Detection of classical cipher types with feature-learning approaches. In Yue Xu, X. Rosalind Wang, Anton R. Lord, Yee Ling Boo, Richi Nayak, Yanchang Zhao, and Graham J. Williams, editors, *Data Mining - 19th Australasian Conference on Data Mining, AusDM, Brisbane, QLD, Australia, December 14-15, 2021, Proceedings*, volume 1504 of *Communications in Computer and Information Science*, pages 152–164. Springer, 2021. doi:10.1007/978-981-16-8531-6_11.

[LMSV07] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. *Cryptography and Cryptanalysis Through Computational Intelligence*, page 1–49. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-71078-3_1.

[LRC22] Jiashuo Liu, Jiongjiong Ren, and Shaozhen Chen. Effective network parameter reduction schemes for neural distinguisher. *IACR Cryptol. ePrint Arch.*, 2022:1765, 2022. URL: https://eprint.iacr.org/2022/1765.

[LRC23] Jiashuo Liu, Jiongjiong Ren, and Shaozhen Chen. A deep learning aided differential distinguisher improvement framework with more lightweight and universality. *Cybersecur.*, 6(1):47, 2023. doi:10.1186/S42400-023-00176-7.

[LRC24] Xiaowei Li, Jiongjiong Ren, and Shaozhen Chen. Improved deep learning aided key recovery framework: applications to large-state block ciphers. *Frontiers Inf. Technol. Electron. Eng.*, 25(10):1406–1420, 2024. doi:10.1631/FITEE.2300848.

[LRCL23] Jiashuo Liu, Jiongjiong Ren, Shaozhen Chen, and ManMan Li. Improved neural distinguishers with multi-round and multi-splicing construction. *J. Inf. Secur. Appl.*, 74:103461, 2023. doi:10.1016/J.JISA.2023.103461.

[LSE+26] Haolin Liu, Adrien Siproudhis, Samuel Experton, Peter Lorenz, Christina Boura, and Thomas Peyrin. Navigating the deep: End-to-end extraction on deep neural networks. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology - EUROCRYPT 2026 - 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques,*

Rome, Italy, May 10-14, 2026, Proceedings, Part VI, volume 16546 of *Lecture Notes in Computer Science*, pages 482–512. Springer, 2026. doi:10.1007/978-3-032-25333-0_17.

[LTJ+21] Ting Rong Lee, Je Sen Teh, Norziana Jamil, Jasy Liew Suet Yan, and Jiageng Chen. Lightweight block cipher security evaluation based on machine learning classifiers and active s-boxes. *IEEE Access*, 9:134052–134064, 2021. doi:10.1109/ACCESS.2021.3116468.

[LTZ22a] Lijun Lyu, Yi Tu, and Yingjie Zhang. Deep learning assisted key recovery attack for round-reduced simeck32/64. In *Information Security: 25th International Conference, ISC 2022, Bali, Indonesia, December 18–22, 2022, Proceedings*, page 443–463, Berlin, Heidelberg, 2022. Springer-Verlag. doi:10.1007/978-3-031-22390-7_26.

[LTZ22b] Lijun Lyu, Yi Tu, and Yingjie Zhang. Improving the deep-learning-based differential distinguisher and applications to simeck. In *25th IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2022, Hangzhou, China, May 4-6, 2022*, pages 465–470. IEEE, 2022. doi:10.1109/CSCWD54268.2022.9776036.

[LWAZ+23] Cathy Yuanchen Li, Emily Wenger, Zeyuan Allen-Zhu, Kristin Lauter, and Francois Charton. Salsa verde: a machine learning attack on learning with errors with sparse small secrets. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL: https://dl.acm.org/doi/10.5555/3666122.3668444.

[MGMP21] Girish Mishra, Indivar Gupta, S V S S N V G Krishna Murthy, and S K Pal. Deep learning based cryptanalysis of stream ciphers. *Defence Science Journal*, 71(4):499–506, Jul. 2021. doi:10.14429/dsj.71.16209.

[MJBH22] Luca Mariot, Domagoj Jakobovic, Thomas Bäck, and Julio C. Hernandez-Castro. Artificial intelligence for the design of symmetric cryptographic primitives. In *Security and Artificial Intelligence*, pages 3–24. Springer Nature, 2022. doi:10.1007/978-3-030-98795-4_1.

[MKMP21] Girish Mishra, SVSSNVG Krishna Murthy, and SK Pal. Dependency of lightweight block ciphers over s-boxes: A deep learning based analysis. *Journal of Discrete Mathematical Sciences and Cryptography*, 26:1–21, 2021. doi:10.1080/09720529.2021.1932889.

[MLR+24] Isabella Martínez, Valentina López, Daniel Rambaut, Germán Obando, Valérie Gauthier-Umaña, and Juan F. Pérez. Recent advances in machine learning for differential cryptanalysis. In Marta Tabares, Paola Vallejo, Biviana Suarez, Marco Suarez, Oscar Ruiz, and Jose Aguilar, editors, *Advances in Computing*, pages 45–56, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-47372-2_5.

[MLYW22] Pingchuan Ma, Zhibo Liu, Yuanyuan Yuan, and Shuai Wang. Neurald: Detecting indistinguishability violations of oblivious RAM with neural distinguishers. *IEEE Trans. Inf. Forensics Secur.*, 17:982–997, 2022. doi:10.1109/TIFS.2022.3155274.

[Moh23] Hesham Mohammed. *Cross-Layer Cryptography for Wireless Communication*. PhD thesis, State University of New York at Albany, 2023. URL:

https://scholarsarchive.library.albany.edu/cgi/viewcontent.cgi?article=4456&context=legacy-etd.

[Mon23] Monkey2022. Practical attacks on round-reduced SIMON based on deep learning (code). https://github.com/Monkey2022/Practical-Attacks-of-Round-Reduced-SIMON-Based-on-Deep-Learning, 2023. GitHub repository for HRC23.

[MP87] Marvin Minsky and Seymour Papert. *Perceptrons - an introduction to computational geometry*. MIT Press, 1987.

[MPKM+22] Girish Mishra, S. K. Pal, S. V. S. S. N. V. G. Krishna Murthy, Ishan Prakash, and Anshul Kumar. Deep learning-based differential distinguisher for lightweight ciphers gift-64 and pride. In Shikha Agrawal, Kamlesh Kumar Gupta, Jonathan H. Chan, Jitendra Agrawal, and Manish Gupta, editors, *Machine Intelligence and Smart Systems*, pages 245–257, Singapore, 2022. Springer Nature Singapore. doi:10.1007/978-981-16-9650-3_19.

[MPM+21] Girish Mishra, S. K. Pal, S. V. S. S. N. V. G. Krishna Murthy, Kanishk Vats, and Rakshak Raina. Distinguishing lightweight block ciphers in encrypted images. *Defence Science Journal*, 71(5):647–655, Sep. 2021. doi:10.14429/dsj.71.16843.

[MS20] Eran Malach and Shai Shalev-Shwartz. The implications of local correlation on learning some deep functions. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: https://dl.acm.org/doi/10.5555/3495724.3495836.

[MSA20] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. doi:10.1016/j.ijforecast.2019.04.014.

[NMN24] Mahendra Shridhar Naik, Madhavi Mallam, and Chaitra Soppinhalli Nataraju. Machine learning-based lightweight block ciphers for resource-constrained internet of things networks: a review. *International Journal of Electrical & Computer Engineering (2088-8708)*, 14(3), 2024. doi:10.11591/ijece.v14i3.pp2896-2907.

[Nor22] NoricaBacuieti. The SPECK attack (implementation). https://github.com/NoricaBacuieti/TheSpeckAttack, 2022. GitHub repository for BBP22.

[NP33] Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933. doi:10.1098/rsta.1933.0009.

[NR23] Abderrahmane Nitaj and Tajjeeddine Rachidi. Applications of neural network-based AI in cryptography. *Cryptogr.*, 7(3):39, 2023. doi:10.3390/CRYPTOGRAPHY7030039.

[PCDC24] Debranjan Pal, Mainak Chaudhury, Abhijit Das, and Dipanwita Roy Chowdhury. Deep learning-based differential distinguishers for nist standard authenticated encryption and permutations. In Debasis Giri, Jaideep Vaidya, S. Ponnusamy, Zhiqiang Lin, Karuna Pande Joshi, and V. Yegnanarayanan, editors, *Proceedings of the Tenth International Conference on Mathematics and Computing*, pages 1–13, Singapore, 2024. Springer Nature Singapore. doi:10.1007/978-981-97-2069-9_1.

[Pea23] Peacker. CLAASP white paper code. https://github.com/peacker/claasp_white_paper, 2023. GitHub repository for BGG+23 (CLAASP white paper code).

[PJ21] Stjepan Picek and Domagoj Jakobovic. Evolutionary computation and machine learning in cryptology. In Krzysztof Krawiec, editor, *GECCO '21: Genetic and Evolutionary Computation Conference, Companion Volume, Lille, France, July 10-14, 2021*, pages 1089–1118. ACM, 2021. doi:10.1145/3449726.3461420.

[PJ22] Stjepan Picek and Domagoj Jakobovic. Evolutionary computation and machine learning in security. In Jonathan E. Fieldsend and Markus Wagner, editors, *GECCO '22: Genetic and Evolutionary Computation Conference, Companion Volume, Boston, Massachusetts, USA, July 9 - 13, 2022*, pages 1572–1601. ACM, 2022. doi:10.1145/3520304.3534087.

[PKM23] Seonghwan Park, Hyunil Kim, and Inkyu Moon. Automated classical cipher emulation attacks via unified unsupervised generative adversarial networks. *Cryptogr.*, 7(3):35, 2023. doi:10.3390/CRYPTOGRAPHY7030035.

[PLH+24] Thomas Prantl, Marco Lauer, Lukas Horn, Simon Engel, David Dingel, André Bauer, Christian Krupitzer, and Samuel Kounev. Security analysis of a decentralized, revocable and verifiable attribute-based encryption scheme. In *Proceedings of the 19th International Conference on Availability, Reliability and Security, ARES 2024, Vienna, Austria, 30 July 2024 - 2 August 2024*, pages 24:1–24:11. ACM, 2024. doi:10.1145/3664476.3664487.

[PMC+22] Debranjan Pal, Upasana Mandal, Mainak Chaudhury, Abhijit Das, and Dipanwita Roy Chowdhury. A deep neural differential distinguisher for ARX based block cipher. *IACR Cryptol. ePrint Arch.*, 2022:1195, 2022. URL: https://eprint.iacr.org/2022/1195.

[PMDC22] Debranjan Pal, Upasana Mandal, Abhijit Das, and Dipanwita Roy Chowdhury. Deep learning based differential classifier of PRIDE and RC5. In Srikanth Prabhu, Shiva Raj Pokhrel, and Gang Li, editors, *Applications and Techniques in Information Security - 13th International Conference, ATIS 2022, Manipal, India, December 30-31, 2022, Revised Selected Papers*, volume 1804 of *Communications in Computer and Information Science*, pages 46–58. Springer, 2022. doi:10.1007/978-981-99-2264-2_4.

[PMK20] Manan Pareek, Girish Mishra, and Varun Kohli. Deep learning based analysis of key scheduling algorithm of PRESENT cipher. *IACR Cryptol. ePrint Arch.*, 2020:981, 2020. URL: https://eprint.iacr.org/2020/981.

[PPS14] Kenneth G. Paterson, Bertram Poettering, and Jacob C. N. Schuldt. Big bias hunting in amazonia: Large-scale computation and exploitation of RC4 biases (invited paper). In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference*

on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, volume 8873 of *Lecture Notes in Computer Science*, pages 398–419. Springer, 2014. doi:10.1007/978-3-662-45611-8_21.

[PPWR23] Raphaël C.-W. Phan, Arghya Pal, KokSheik Wong, and Sailaja Rajanala. Cryptographically distinguishing autoencoder for cipher cryptanalysis. In *IEEE Global Communications Conference, GLOBECOM 2023, Kuala Lumpur, Malaysia, December 4-8, 2023*, pages 4467–4472. IEEE, 2023. doi:10.1109/GLOBECOM54140.2023.10437839.

[PSM24] Pooja, Shantanu, and Girish Mishra. Related-key neural distinguisher for round-reduced present cipher. In Swagatam Das, Snehanshu Saha, Carlos A. Coello Coello, and Jagdish C. Bansal, editors, *Advances in Data-Driven Computing and Intelligent Systems*, pages 393–405, Singapore, 2024. Springer Nature Singapore. doi:10.1007/978-981-99-9524-0_30.

[PTD22] Milena Gjorgjievska Perusheska, Hristina Mihajloska Trpceska, and Vesna Dimitrova. Deep learning-based cryptanalysis of different aes modes of operation. In Kohei Arai, editor, *Advances in Information and Communication*, pages 675–693, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-030-98015-3_47.

[PVM24] Mattia Paravisi, Andrea Visconti, and Dario Malchiodi. Security analysis of cryptographic algorithms: Hints from machine learning. In Lazaros S. Iliadis, Ilias Maglogiannis, Antonios Papaleonidas, Elias Pimenidis, and Chrisina Jayne, editors, *Engineering Applications of Neural Networks - 25th International Conference, EANN 2024, Corfu, Greece, June 27-30, 2024, Proceedings*, volume 2141 of *Communications in Computer and Information Science*, pages 569–580. Springer, 2024. doi:10.1007/978-3-031-62495-7_43.

[PYW24] Qi Pang, Yuanyuan Yuan, and Shuai Wang. Mpcdiff: Testing and repairing mpc-hardened deep learning models. In *Network and Distributed System Security (NDSS) Symposium*. NDSS, 2024.

[QLW+26] Xiaokang Qi, Hao Lei, Longxiang Wei, Xiaohan Sun, and Meiqin Wang. Cryptanalytic extraction of neural networks with various activation functions. *IACR Trans. Symmetric Cryptol.*, 2026(1):468–505, 2026. doi:10.46586/TOSC.V2026.I1.468-505.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi:10.1007/s11263-015-0816-y.

[RGD+18] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4390–4399. PMLR, 2018. URL: http://proceedings.mlr.press/v80/ruff18a.html.

[Riv91] Ronald L. Rivest. Cryptography and machine learning. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan, November 11-14, 1991, Proceedings*, volume 739 of *Lecture Notes in Computer Science*, pages 427–439. Springer, 1991. doi:10.1007/3-540-57332-1_36.

[RKS23] Vignesh Rajakumar, Lakshmy KV, and Chungath Srinivasan. Deep learning based cryptanalysis on slim cipher. In *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pages 1–6, 2023. doi:10.1109/CISCT57197.2023.10351339.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22:400–407, 1951. URL: https://www.jstor.org/stable/2236626.

[Ros24] Matteo Rossi. *Automatic differential cryptanalysis of symmetric ciphers*. PhD thesis, Polytechnic University of Turin, Italy, 2024. URL: https://iris.polito.it/retrieve/71ab9e32-1306-4c60-b3fc-307f820bd399/conv_thesis_rossi_final.pdf.

[RRSM22] Reshma Rajan, Rupam Kumar Roy, Diptakshi Sen, and Girish Mishra. Deep learning-based differential distinguisher for lightweight cipher giftcofb. In Shikha Agrawal, Kamlesh Kumar Gupta, Jonathan H. Chan, Jitendra Agrawal, and Manish Gupta, editors, *Machine Intelligence and Smart Systems*, pages 397–406, Singapore, 2022. Springer Nature Singapore. doi:10.1007/978-981-16-9650-3_31.

[SAH+20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nat.*, 588(7839):604–609, 2020. doi:10.1038/S41586-020-03051-4.

[SBG+25] Arpita Sarkar, Malay Bhattacharyya, Utpal Garain, Saibal Kumar Pal, Shantanu, Sanghamitra Bandyopadhyay, and Nikhil R. Pal. Leveraging synergy to design neural differential distinguishers for lightweight block ciphers. *IEEE Trans. Emerg. Top. Comput. Intell.*, 9(2):2001–2013, 2025. doi:10.1109/TETCI.2024.3485756.

[Seo24] Byoungjin Seok. Truncated differential-neural key recovery attacks on round-reduced hight. *Electronics*, 13(20), 2024. doi:10.3390/electronics13204053.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. J.*, 28(4):656–715, 1949. doi:10.1002/J.1538-7305.1949.TB00928.X.

[SHM+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi:10.1038/NATURE16961.

[SHS+18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi:10.1126/science.aar6404.

[SL25] Byoungjin Seok and Changhoon Lee. A novel approach to construct a good dataset for differential-neural cryptanalysis. *IEEE Trans. Dependable Secur. Comput.*, 22(1):246–262, 2025. doi:10.1109/TDSC.2024.3387662.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9. IEEE Computer Society, 2015. doi:10.1109/CVPR.2015.7298594.

[SLL24] Jiali Shi, Chao Li, and Guoqiang Liu. Differential attack with constants on μ2 block cipher. *Comput. J.*, 67(1):195–209, 2024. doi:10.1093/COMJNL/BXAC168.

[SM24] Ayan Sajwan and Girish Mishra. Comparative analysis of resnet and densenet for differential cryptanalysis of speck 32/64 lightweight block cipher. In Atul Chaturvedi, Sartaj Ul Hasan, Bimal Kumar Roy, and Boaz Tsaban, editors, *Cryptology and Network Security with Machine Learning*, pages 495–504, Singapore, 2024. Springer Nature Singapore. doi:10.1007/978-981-97-0641-9_34.

[SMR+24] Avital Shafran, Eran Malach, Thomas Ristenpart, Gil Segev, and Stefano Tessaro. Is ml-based cryptanalysis inherently limited? simulating cryptographic adversaries via gradient-based methods. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 37–71. Springer, 2024. doi:10.1007/978-3-031-68391-6_2.

[So20] Jaewoo So. Deep learning-based cryptanalysis of lightweight block ciphers. *Secur. Commun. Networks*, 2020:3701067:1–3701067:11, 2020. doi:10.1155/2020/3701067.

[Som23] Åvald Åslaugson Sommervoll. *Machine learning for offensive cyber operations*. PhD thesis, Institute for Informatics, University of Oslo, 2023. URL: https://nva.sikt.no/registration/019c9131172a-7acb2b72-fb80-4652-a622-140766b7630a.

[SS23] Samuel Stevens and Yu Su. Memorization for good: Encryption with autoregressive language models. *CoRR*, abs/2305.10445, 2023. doi:10.48550/ARXIV.2305.10445.

[SSL+22] Tao Sun, Dongsu Shen, Saiqin Long, Qingyong Deng, and Shiguo Wang. Neural distinguishers on tinyjambu-128 and gift-64. In Mohammad Tanveer, Sonali Agarwal, Seiichi Ozawa, Asif Ekbal, and Adam Jatowt, editors, *Neural Information Processing - 29th International Conference, ICONIP 2022, Virtual Event, November 22-26, 2022, Proceedings, Part V*, volume 1792 of *Communications in Computer and Information Science*, pages 419–431. Springer, 2022. doi:10.1007/978-981-99-1642-9_36.

[SSL+24] Dongsu Shen, Yijian Song, Yuan Lu, Saiqin Long, and Shujuan Tian. Neural differential distinguishers for GIFT-128 and ASCON. *J. Inf. Secur. Appl.*, 82:103758, 2024. doi:10.1016/J.JISA.2024.103758.

[SST24] Ajeet Singh, Kaushik Bhargav Sivangi, and Appala Naidu Tentu. Machine learning and cryptanalysis: An in-depth exploration of current practices and future potential. *Journal of Computing Theories and Applications*, 1(3):257–272, 2024. doi:10.62411/jcta.9851.

[SSUM14] Marek Sys, Petr Svenda, Martin Ukrop, and Vashek Matyas. Constructing empirical tests of randomness. In *Proceedings of the 11th International Conference on Security and Cryptography*. SCITEPRESS - Science and Technology Publications, 2014. doi:10.5220/0005023902290237.

[Sug24] Nobuyuki Sugio. Implementation of cryptanalytic program for ASCON using chatgpt. In *Twelfth International Symposium on Computing and Networking, CANDAR 2024 - Workshops, Naha, Japan, November 26-29, 2024*, pages 307–313. IEEE, 2024. doi:10.1109/CANDARW64572.2024.00057.

[SWL+25] Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, François Charton, and Kristin E. Lauter. Salsa fresca: Angular embeddings and pre-training for ML attacks on learning with errors. *Trans. Mach. Learn. Res.*, 2025, 2025. URL: https://openreview.net/forum?id=w4nd5695sq.

[SZM20] Heng-Chuan Su, Xuan-Yong Zhu, and Duan Ming. Polytopic attack on round-reduced simon32/64 using deep learning. In Yongdong Wu and Moti Yung, editors, *Information Security and Cryptology - 16th International Conference, Inscrypt 2020, Guangzhou, China, December 11-14, 2020, Revised Selected Papers*, volume 12612 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2020. doi:10.1007/978-3-030-71852-7_1.

[Tan23] Quan Quan Tan. *Cryptanalysis of lightweight symmetric-key cryptographic algorithms*. PhD thesis, Nanyang Technical University, Singapore, 2023. URL: https://dr.ntu.edu.sg/entities/publication/5f94e1d3-f43f-4ebb-8779-cb88e58d7013.

[Tar21] Tarunyadav. Differential-ML-Distinguisher (code). https://github.com/tarunyadav/Differential-ML-Distinguisher, 2021. GitHub repository for YK21.

[Tar24] Tarunyadav. Improved differential distinguisher for GIFT-128 and ASCON (code). https://github.com/tarunyadav/Improved-Differential-Distinguisher-GIFT128-ASCON, 2024. GitHub repository for YK24.

[TCHC07] Juan M. E. Tapiador, John A. Clark, and Julio C. Hernandez-Castro. *Non-linear Cryptanalysis Revisited: Heuristic Search for Approximations to S-Boxes*, page 99–117. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-77272-9_7.

[TD21] Zakaria Tolba and Makhlouf Derdour. Deep learning for cryptanalysis attack on iomt wireless communications via smart eavesdropping. In *2021 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6. IEEE, 2021. doi:10.1109/ICNAS53565.2021.9628924.

[TDD22] Zakaria Tolba, Makhlouf Derdour, and Nour El Houda Dehimi. Machine learning based cryptanalysis techniques: perspectives, challenges and future directions. In *4th International Conference on Pattern Analysis and Intelligent Systems, PAIS 2022, Oum El Bouaghi, Algeria, October 12-13, 2022*, pages 1–7. IEEE, 2022. doi:10.1109/PAIS56586.2022.9946889.

[TDF+22] Zakaria Tolba, Makhlouf Derdour, Mohamed Amine Ferrag, S. M. Muyeen, and Mohamed Benbouzid. Automated deep learning BLACK-BOX attack for multimedia P-BOX security assessment. *IEEE Access*, 10:94019–94039, 2022. doi:10.1109/ACCESS.2022.3204175.

[TH21] Wenqiang Tian and Bin Hu. Deep learning assisted differential cryptanalysis for the lightweight cipher SIMON. *KSII Trans. Internet Inf. Syst.*, 15(2):600–616, 2021. URL: https://doi.org/10.3837/tiis.2021.02.012, doi:10.3837/TIIS.2021.02.012.

[TL19] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. URL: http://proceedings.mlr.press/v97/tan19a.html.

[TSL23] Erzhena Tcydenova, Byoungjin Seok, and Changhoon Lee. Related-key neural distinguisher on block ciphers speck-32/64, hight and gost. *Journal of Platform Technology*, 11(1):72–84, 2023. URL: https://koreascience.kr/article/JAKO202325879122981.page.

[TTJ23] Weijian Teng, Je Sen Teh, and Norziana Jamil. On the security of lightweight block ciphers against neural distinguishers: Observations on lbc-iot and SLIM. *J. Inf. Secur. Appl.*, 76:103531, 2023. doi:10.1016/J.JISA.2023.103531.

[Tu22] Yi Tu. *Machine learning-aided and SAT-aided cryptanalysis of symmetric-key primitives*. PhD thesis, Nanyang Technical University, Singapore, 2022. URL: https://dr.ntu.edu.sg/entities/publication/df06e6dd-7554-46d0-a949-0320fcb431b0.

[Val84] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984. doi:10.1145/1968.1972.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL: https://dl.acm.org/doi/10.5555/3295222.3295349.

[WCCL22] Emily Wenger, Mingjie Chen, François Charton, and Kristin E. Lauter. SALSA: attacking lattice cryptography with transformers. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022,*

New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL: https://dl.acm.org/doi/10.5555/3600270.3602805.

[Wea47] W Weaver. Letter to norbert wiener, 4 March 1947. URL: https://is.muni.cz/el/phil/jaro2016/PLIN041/um/Weaver-1947-original.pdf.

[WG24] Wanqing Wu and Mingyu Guo. Improved integral neural distinguisher model for lightweight cipher PRESENT. *Cybersecur.*, 7(1):65, 2024. doi:10.1186/S42400-024-00258-0.

[WIO24] Hayato Watanabe, Ryoma Ito, and Toshihiro Ohigashi. On the effects of neural network-based output prediction attacks on the design of symmetric-key ciphers. In Shlomi Dolev, Michael Elhadad, Miroslaw Kutylowski, and Giuseppe Persiano, editors, *Cyber Security, Cryptology, and Machine Learning - 8th International Symposium, CSCML 2024, Be'er Sheva, Israel, December 19-20, 2024, Proceedings*, volume 15349 of *Lecture Notes in Computer Science*, pages 201–218. Springer, 2024. doi:10.1007/978-3-031-76934-4_13.

[WNB+26] Ping Wang, Shishir Nagaraja, Aurélien Bourquard, Haichang Gao, and Jeff Yan. Sok: Acoustic side channels. *ACM Comput. Surv.*, 58(7):172:1–172:33, 2026. doi:10.1145/3778350.

[WQW+24] Zehan Wu, Kexin Qiao, Zhaoyang Wang, Junjie Cheng, and Liehuang Zhu. Mixture differential cryptanalysis on round-reduced simon32/64 using machine learning. *Mathematics*, 12(9):1401, 2024. doi:10.3390/math12091401.

[WTZ+22] Huijiao Wang, Jiapeng Tian, Xin Zhang, Yongzhuang Wei, and Hua Jiang. Multiple differential distinguisher of simeck32/64 based on deep learning. *Security and Communication Networks*, 2022(1):7564678, 2022. doi:10.1155/2022/7564678.

[WW21] Gao Wang and Gaoli Wang. Improved differential-ml distinguisher: Machine learning based generic extension for differential analysis. In Debin Gao, Qi Li, Xiaohong Guan, and Xiaofeng Liao, editors, *Information and Communications Security - 23rd International Conference, ICICS 2021, Chongqing, China, November 19-21, 2021, Proceedings, Part II*, volume 12919 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2021. doi:10.1007/978-3-030-88052-1_2.

[WW22] Feifan Wang and Gaoli Wang. Improved differential-linear attack with application to round-reduced speck32/64. In Giuseppe Ateniese and Daniele Venturi, editors, *Applied Cryptography and Network Security - 20th International Conference, ACNS 2022, Rome, Italy, June 20-23, 2022, Proceedings*, volume 13269 of *Lecture Notes in Computer Science*, pages 792–808. Springer, 2022. doi:10.1007/978-3-031-09234-3_39.

[WW24a] Gao Wang and Gaoli Wang. Enhanced related-key differential neural distinguishers for SIMON and SIMECK block ciphers. *PeerJ Comput. Sci.*, 10:e2566, 2024. doi:10.7717/PEERJ-CS.2566.

[WW24b] Gao Wang and Gaoli Wang. Keeping classical distinguisher and neural distinguisher in balance. *J. Inf. Secur. Appl.*, 84:103816, 2024. doi:10.1016/J.JISA.2024.103816.

[WWH21] Gao Wang, Gaoli Wang, and Yu He. Improved machine learning assisted (related-key) differential distinguishers for lightweight ciphers. In *20th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2021, Shenyang, China, October 20-22, 2021*, pages 164–171. IEEE, 2021. doi:10.1109/TRUSTCOM53373.2021.00039.

[WWS24a] Gao Wang, Gaoli Wang, and Siwei Sun. Investigating and enhancing the neural distinguisher for differential cryptanalysis. *IEICE Trans. Inf. Syst.*, 107(8):1016–1028, 2024. doi:10.1587/TRANSINF.2024EDP7011.

[WWS24b] Gao Wang, Gaoli Wang, and Siwei Sun. A new (related-key) neural distinguisher using two differences for differential cryptanalysis. *IET Inf. Secur.*, 2024(1), 2024. doi:10.1049/2024/4097586.

[XLC+22] Ruiqi Xia, Manman Li, Shaozhen Chen, et al. Cryptographic algorithms identification based on deep learning. In *CS & IT Conference Proceedings*, volume 12. CS & IT Conference Proceedings, 2022. doi:10.5121/csit.2022.121217.

[YBBP23] Trevor Yap, Adrien Benamira, Shivam Bhasin, and Thomas Peyrin. Peek into the black-box: Interpretable neural network using SAT equations in side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):24–53, 2023. doi:10.46586/TCHES.V2023.I2.24-53.

[YK21] Tarun Yadav and Manoj Kumar. Differential-ml distinguisher: Machine learning based generic extension for differential cryptanalysis. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 191–212. Springer, 2021. doi:10.1007/978-3-030-88238-9_10.

[YK22] Tarun Yadav and Manoj Kumar. Modeling large s-box in MILP and a (related-key) differential attack on full round PIPO-64/128. In Lejla Batina, Stjepan Picek, and Mainack Mondal, editors, *Security, Privacy, and Applied Cryptography Engineering - 12th International Conference, SPACE 2022, Jaipur, India, December 9-12, 2022, Proceedings*, volume 13783 of *Lecture Notes in Computer Science*, pages 3–27. Springer, 2022. doi:10.1007/978-3-031-22829-2_1.

[YK24] Tarun Yadav and Manoj Kumar. ML based improved differential distinguisher with high accuracy: Application to GIFT-128 and ASCON. In Johann Knechtel, Urbi Chatterjee, and Domenic Forte, editors, *Security, Privacy, and Applied Cryptography Engineering - 14th International Conference, SPACE 2024, Kottayam, India, December 14-17, 2024, Proceedings*, volume 15351 of *Lecture Notes in Computer Science*, pages 287–316. Springer, 2024. doi:10.1007/978-3-031-80408-3_18.

[YLFP24] Shivani Yadao, Nguyen Thi Dieu Linh, Areesha Fatima, and Bhuvan Puri. Ai-driven cryptographic algorithm identification: Exploring methodologies and practical applications. In *2024 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, pages 56–61. IEEE, 2024. doi:10.1109/ICMLANT63295.2024.00015.

[YW23] Xiaoteng Yue and Wanqing Wu. Improved neural differential distinguisher model for lightweight cipher speck. *Applied Sciences*, 13:6994, 2023. doi:10.3390/app13126994.

[YW24] Xue Yuan and Qichun Wang. Improving differential-neural distinguisher for simeck family. *IACR Cryptol. ePrint Arch.*, 2024:2002, 2024. URL: https://eprint.iacr.org/2024/2002.

[YWZW26] Yufei Yuan, Wenling Wu, Lei Zhang, and Ruichen Wu. Fourier analysis of neural distinguishers. *IACR Trans. Symmetric Cryptol.*, 2026(1):441–467, 2026. doi:10.46586/TOSC.V2026.I1.441-467.

[ZDW+23] Rui Zhou, Ming Duan, Qi Wang, Qianqiong Wu, Sheng Guo, Lulu Guo, and Zheng Gong. Neural-linear attack based on distribution data and its application on DES. *IACR Cryptol. ePrint Arch.*, 2023:580, 2023. URL: https://eprint.iacr.org/2023/580.

[ZG24] Yue Zhong and Jieming Gu. Lightweight block ciphers for resource-constrained environments: A comprehensive survey. *Future Gener. Comput. Syst.*, 157:288–302, 2024. doi:10.1016/J.FUTURE.2024.03.054.

[ZKL20] Behnam Zahednejad, Lishan Ke, and Jing Li. A novel machine learning-based approach for security analysis of authentication and key agreement protocols. *Secur. Commun. Networks*, 2020:8848389:1–8848389:15, 2020. doi:10.1155/2020/8848389.

[ZL22] Behnam Zahednejad and Lijun Lyu. An improved integral distinguisher scheme based on neural networks. *Int. J. Intell. Syst.*, 37(10):7584–7613, 2022. doi:10.1002/INT.22895.

[ZLF+24] Xinhao Zheng, Yang Li, Cunxin Fan, Huaijin Wu, Xinhao Song, and Junchi Yan. Learning plaintext-ciphertext cryptographic problems via anf-based SAT instance representation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL: https://dl.acm.org/doi/10.5555/3737916.3739233.

[ZLHH25] Siqi Zhu, Lang Li, Zhiwen Hu, and Yemao Hu. Bcs: A neural distinguisher method based on differential propagation uncertainty of nonlinear components and network adaptability. *Physica Scripta*, 100(3), 2025. doi:10.1088/1402-4896/adae63.

[ZLWL23] Liu Zhang, Jinyu Lu, Zilong Wang, and Chao Li. Improved differential-neural cryptanalysis for round-reduced SIMECK32/64. *Frontiers Comput. Sci.*, 17(6):176817, 2023. doi:10.1007/S11704-023-3261-Z.

[ZWC23] Liu Zhang, Zilong Wang, and Yindong Chen. Improving the accuracy of differential-neural distinguisher for des, chaskey, and PRESENT. *IEICE Trans. Inf. Syst.*, 106(7):1240–1243, 2023. doi:10.1587/TRANSINF.2022EDL8094.

[ZWL24] Liu Zhang, Zilong Wang, and Jinyu Lu. Differential-neural cryptanalysis on AES. *IEICE Trans. Inf. Syst.*, 107(10):1372–1375, 2024. doi:10.1587/TRANSINF.2024EDL8044.

[ZWW22] Liu Zhang, Zilong Wang, and Boyang Wang. Improving differential-neural cryptanalysis with inception blocks. *IACR Cryptol. ePrint Arch.*, 2022:183, 2022. URL: https://eprint.iacr.org/2022/183.

[ZWW24] Liu Zhang, Zilong Wang, and Baocang Wang. Improving differential-neural cryptanalysis. *IACR Communications in Cryptology*, 1(3), 2024. doi:10.62056/ay11wa3y6.

[ZYS+26] Liu Zhang, Yiran Yao, Danping Shi, Dongchen Chai, Jian Guo, and Zilong Wang. Neural-inspired advances in integral cryptanalysis. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology - EUROCRYPT 2026 - 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Rome, Italy, May 10-14, 2026, Proceedings, Part VI*, volume 16546 of *Lecture Notes in Computer Science*, pages 451–481. Springer, 2026. doi:10.1007/978-3-032-25333-0_16.

[ZYSH24] Jiahao Zhao, Qianqian Yang, Ling Song, and Lei Hu. A note on neutral bits for ARX ciphers from the perspective of BCT. In Dongdai Lin, Meiqin Wang, and Moti Yung, editors, *Information Security and Cryptology - 20th International Conference, Inscrypt 2024, Kunming, China, December 14-16, 2024, Revised Selected Papers, Part II*, volume 15544 of *Lecture Notes in Computer Science*, pages 322–341. Springer, 2024. doi:10.1007/978-981-96-4734-7_16.

[ZZ21] Wenyu Zhang and Yaqun Zhao. Ensemble learning-based differential distinguishers for lightweight cipher. In *EITCE 2021: 5th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, October 22 - 24, 2021*, pages 28–34. ACM, 2021. doi:10.1145/3501409.3501415.

[ZZC+22] Qingqing Zhang, Hongxing Zhang, Xiaotong Cui, Xing Fang, and Xingyang Wang. Side channel analysis of SPECK based on transfer learning. *Sensors*, 22(13):4671, 2022. doi:10.3390/S22134671.

[ZZS21] Zimin Zhang, Wenying Zhang, and Hongfang Shi. Genetic algorithm assisted state-recovery attack on round-reduced xoodyak. In Elisa Bertino, Haya Schulmann, and Michael Waidner, editors, *Computer Security - ESORICS 2021 - 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4-8, 2021, Proceedings, Part II*, volume 12973 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2021. doi:10.1007/978-3-030-88428-4_13.

[ZZW24] Weixi Zheng, Liu Zhang, and Zilong Wang. Theoretical explanation and improvement of deep learning-aided cryptanalysis. *IACR Cryptol. ePrint Arch.*, 2024:322, 2024. URL: https://eprint.iacr.org/2024/322.

[ZZY+21] Runlian Zhang, Mi Zhang, Jiaxu Yan, Yixing Li, Xiaonian Wu, and Lingchen Li. Differential cryptanalysis of twegift-128 based on neural network. In *Sixth IEEE International Conference on Data Science in Cyberspace, DSC 2021, Shenzhen, China, October 9-11, 2021*, pages 529–534. IEEE, 2021. doi:10.1109/DSC53577.2021.00084.

A Comparative Review of the remaining Neural Differential Distinguishers

A.1 AES

AES is a widely used block cipher standardized by NIST in 2001 and designed for general-purpose encryption. It operates on 128-bit blocks and supports key sizes of 128, 192, or

Table 7: Overview of the Neural Differential Distinguishers for AES.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
AES-128	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT$_{\text{tr}}$-R	20M	2M	-	2	0.9981	[ZWL24]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	2	1	[ZWL24]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
[RK] Related key setting.

256 bits. The cipher's structure comprises 10, 12, or 14 rounds (depending on the key size), each involving four transformations: SubBytes (substitution), ShiftRows (permutation), MixColumns (linear mixing), and AddRoundKey. Notably, AES's SubBytes transformation uses a single 8-bit S-box followed by an affine transformation.

A.2 ARADI

Table 8: Overview of the Neural Differential Distinguishers for ARADI.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
ARADI	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	✓	5	0.5954	[BFG+24]
ARADI [RK]	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	✓	6	0.5631	[BFG+24]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
[RK] Related key setting.

ARADI is a low-latency block cipher introduced by the NSA in 2024, specifically designed for memory encryption applications. It operates on 128-bit blocks and utilizes a 256-bit key. The cipher's structure comprises 16 rounds, each involving a combination of substitution and permutation operations. Notably, ARADI's round function employs a unique S-box, a linear layer, and a key addition layer.

A.3 ASCON

Table 9: Overview of the Neural Differential Distinguishers for ASCON.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
ASCON	MLP	2-2-δ-D	1.1M	1.1M	-	3	0.9861	[BBCD21]
	MLP	2-1-δ-R	17M	2M	-	4	0.502	[YK24]
	MLP	2-1-δ-R	20M	20M	-	4	0.5069	[SSL+24]
ASCON$^{\text{Unkeyed}}$	Classical ML	2-2-δ-D	64K	16K	-	3	0.916	[BBD+23]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

ASCON is an SPN-based permutation with an input size of 320 bits. It can be used within a sponge construction to build the authenticated ciphers ASCON-128 and ASCON-128a, both using 128-bit keys, 12 rounds in the initialization, and 64- and 128-bit messages, respectively, with 6 and 8 rounds in the encryption process. The ASCON-hash, also based on a sponge construction, hashes 64-bit messages over 12 rounds. ASCON was announced as the winner of the NIST Lightweight Cryptography Competition in February 2023.

We end this section on ASCON with a critical discussion of [BBD+23]. Baksi et al. [BBD+23] use a validation set of only 8K ciphertext pairs. For $v = 8$K, the worst-case standard error is about 0.56%, giving a two-sided 95% confidence interval of roughly ±1.10%.

A.4 CHASKEY

Table 10: Overview of the Neural Differential Distinguishers for CHASKEY.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
CHASKEY-PERMUTATION	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	17M	40K	-	4	0.6161	[BBCD21]†
	$\mathcal{ND}_{\text{Gohr}}$	32-1-CT-R	20M	2M	-	4	0.7712	[CSYY23]
	INC	16-1-CT-R	60M	2M	-	5	0.5181	[ZWC23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

CHASKEY is an ARX-based permutation with an input size of 128 bits. It operates through 8 rounds of Addition, Rotation, and XOR operations on four 32-bit words

We end this section on CHASKEY with a critical discussion of [BBCD21]. Baksi et al. [BBCD21] use a validation set of only 20K ciphertext pairs. For $v = 20$K, the worst-case standard error is about 0.35%, giving a two-sided 95% confidence interval of roughly ±0.69%.

A.5 DES

DES (Data Encryption Standard) is a 16-round Feistel block cipher with 56-bit keys and 64-bit blocks.

Table 11: Overview of the Neural Differential Distinguishers for DES.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
DES	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	5	0.58	[CSY23]
	$\mathcal{ND}_{\text{Gohr}}$	4-1-CT-R	20M	2M	-	6	0.5653	[CSYY23]
	INC	32-1-CT-R	1280M	32M	-	7	0.5114	[ZWC23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

Zhang et al. [ZWC23] use a staged training approach to obtain a distinguisher for 7-round DES: $4 \cdot 10^7$ samples, 16 pairs each (640M ciphertext pairs).

A.6 FF1 and FF3

FF1 and FF3 are format-preserving encryption algorithms, with 10 and 8 rounds, respectively, with block sizes of 32 and 128 bits and key sizes of 128 bits. We use the notations FFX-D when the domain is digits and FFX-L when the domain is lowercase characters.

We end this section on FF1 and FF3 with a critical discussion of [KKJ+24]. In [KKJ+24], the authors perform neural cryptanalysis of FF1 and FF3 for digits (FFX-D) and lowercase letters (FFX-L). We report the best results in the 2-1-CT-R setting. The number of training and test samples was not specified, nor is the source code available.

Table 12: Overview of the Neural Differential Distinguishers for FF.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
FF1-D	MLP	2-1-CT-R	-	-	-	10	0.855	[KKJ+24]†
FF1-L	MLP	2-1-CT-R	-	-	-	2	0.522	[KKJ+24]†
FF3-D	MLP	2-1-CT-R	-	-	-	8	0.977	[KKJ+24]†
FF3-L	MLP	2-1-CT-R	-	-	-	2	0.554	[KKJ+24]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

Table 13: Overview of the Neural Differential Distinguishers for GIFT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GIFT-64	UNet	12-1-*A*-R	-	-	-	4	0.725	[ZLHH25]†
	$\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	6	0.516	[PMDC22]
	LSTM	3-2-CT-R	17M	4M	-	6	0.5754	[SSL+22]
	MLP	2-1-δ-R	2.1M	300K	-	FULL	0.96	[MPKM+22]†
GIFT-128	MLP	2-1-δ-R	17M	2M	-	7	0.55	[YK24]
	MLP	2-1-δ-R	20M	2M	-	7	0.5542	[SSL+24]
TweGIFT-128	MLP	2-1-CT-R	2M	200K	-	6	0.5675	[ZZY+21]
	MLP	2-1-CT-R	2M	200K	-	7	0.5002	[ZZY+21] †
GIFT-COFB	MLP	2-4-δ-D	20K	20K	-	4	0.615	[RRSM22]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

A.7 GIFT

GIFT is a PRESENT-inspired SPN cipher that uses 128-bit keys to encrypt 64-bit (GIFT64) or 128-bit (GIFT128) blocks for 28 and 40 rounds, respectively. GIFT was one of the finalists of the NIST Lightweight Cryptography Competition.

We end this section on GIFT with a critical discussion of [RRSM22, MPKM+22, ZZY+21]. In [ZLHH25], the number of training and test samples is not specified. In [ZZY+21], the authors claim a distinguisher on 7 rounds because the training accuracy was 0.6487, despite the validation accuracy being non-significant (0.5002); in the table, we report this 7-round distinguisher as it is the best one claimed by the authors, but also their 6-round distinguisher, which has a significant validation accuracy. In [RRSM22], only 10K samples are used for training and testing. For $v = 10$K, the worst-case standard error is about 0.50%, giving a two-sided 95% confidence interval of roughly ±0.98%. In [MPKM+22], the authors claim a full round distinguisher on GIFT-64 with over 90% accuracy, using 2^{20} samples in total, of which 15% are kept for validation, respectively testing, and a MLP architecture; they also claim a full round distinguisher on PRIDE with 100% accuracy. Full-round attacks on modern and reputable ciphers are an extraordinary claim and require extraordinary evidence, which the author's manuscript does not provide.

A.8 GIMLI

GIMLI is a 24-round permutation acting on 384 bits, from which a hash function GIMLI-HASH and an authenticated cipher GIMLI-CIPHER are derived.

We end this section on GIMLI with a critical discussion of [BBCD21]. In [BBCD21], the authors use only 1.4K validation samples. For $v = 1{,}400$, the worst-case standard error is about 1.34%, giving a two-sided 95% confidence interval of roughly ±2.62%. Since the

Table 14: Overview of the Neural Differential Distinguishers for GIMLI.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GIMLI	DBitNet	2-1-CT-R	20M	2M	✓	11	0.524	[BGHR23]
GIMLI-HASH	MLP	2-2-δ-D	400K	40K	-	8	0.5219	[BBCD21]†
GIMLI-CIPHER	MLP	2-2-δ-D	400K	40K	-	8	0.5099	[BBCD21]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

random-guessing baseline lies within this interval, the reported distinguishing advantage is statistically inconclusive. We therefore exclude this result from quantitative comparison.

A.9 GOST

GOST is a block cipher developed by the Soviet Union. It operates on 64-bit blocks with a 256-bit key and follows a Feistel network structure with 32 rounds. Each round applies a key-dependent substitution using fixed S-boxes, followed by modular addition and bitwise rotations to ensure diffusion and security.

Table 15: Overview of the Neural Differential Distinguishers for GOST.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GOST	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	9	0.5430	[TSL23]
GOST $^{\text{RK}}$	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	14	0.7134	[TSL23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
$^{\text{RK}}$ Related key setting.

A.10 HIGHT

Table 16: Overview of the Neural Differential Distinguishers for HIGHT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
HIGHT	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	9	0.7472	[TSL23]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-δ_{tr}-R	20M	2M	-	10	0.5707	[Seo24]
	DBitNet	2-1-CT-R	20M	2M	✓	10	0.751	[BGHR23]
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	11	0.7472	[TSL23]
	LSTM	2-1-A-R	10M	500K	-	15	0.5015	[BPC24b]†
HIGHT $^{\text{RK}}$	DBitNet	2-1-CT-R	20M	2M	✓	14	0.563	[BGHR23]
	DBitNet	2-2-CT-D	20M	2M	✓	14	0.640	[WWS24b]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
$^{\text{RK}}$ Related key setting.
† A critical discussion of these results is provided in the text.

HIGHT is a 32-round ARX-based block cipher operating on 64-bit blocks and 128-bit keys.

We conclude this section on HIGHT with a critical discussion [BPC24b]. Bose et al. [BPC24b] claim improvements in distinguishing additional encryption rounds by sequentially training models on ciphertext pairs. These results should be interpreted with caution.

From a cryptographic perspective, distinguishing advantage is expected to decrease as the number of rounds increases; however, this monotonic degradation is not reflected in their reported results. Moreover, the findings are not fully consistent with prior empirical evidence: Bellini et al. [BGHR23] report stronger distinguishers for rounds 9 and 10 of HIGHT, which calls into question whether the proposed architecture is in fact capturing the relevant differential patterns more effectively.

Given these theoretical and empirical inconsistencies, we treat the results of [BPC24b] as anomalous unless they are supported by sufficient statistical evidence. In line with the criterion used throughout this section, we consider a claimed distinguishing advantage statistically conclusive only when the random-guessing baseline falls outside the corresponding 95% confidence interval. For an effective test set of 2.5×10^5 ciphertext pairs, this means that only reported accuracies above approximately 0.50196 are treated as statistically significant and included in the quantitative comparison.

A.11 KATAN

Table 17: Overview of the Neural Differential Distinguishers for KATAN.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
KATAN32	$\mathcal{ND}_{\text{Gohr}}$	2-1-δ-R	20M	2M	-	51	0.533	[LCLH22]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	59	0.575	[LCLH22]
	$\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$	2-1-CT-R	20M	2M	-	62	0.509	[PMDC22]
	DBitNet	2-1-CT-R	20M	2M	✓	69	0.505	[BGHR23]
KATAN32 [C]	$\mathcal{ND}_{\text{Gohr}}$	64-1-δ-R	64M	6.4M	-	58	0.602	[LLHC23]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	85	0.570	[LCLH22]
KATAN32 [RK,C]	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	112	0.647	[LCLH22]
KATAN48	$\mathcal{ND}_{\text{Gohr}}$	2-1-δ-R	20M	2M	-	40	0.58	[LCLH22]
	$\mathcal{ND}_{\text{Gohr}}$	96-1-δ-R	960M	96M	-	50	0.54	[LCLH22]
KATAN48 [C]	$\mathcal{ND}_{\text{Gohr}}$	64-1-δ-R	64M	6.4M	-	47	0.582	[LLHC23]
	$\mathcal{ND}_{\text{Gohr}}$	96-1-δ-R	960M	96M	-	72	0.582	[LCLH22]
KATAN48 [RK,C]	$\mathcal{ND}_{\text{Gohr}}$	48-1-δ-R	960M	96M	-	96	0.625	[LCLH22]
KATAN64	$\mathcal{ND}_{\text{Gohr}}$	2-1-δ-R	20M	2M	-	31	0.718	[LCLH22]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	36	0.548	[LCLH22]
KATAN64 [C]	$\mathcal{ND}_{\text{Gohr}}$	64-1-δ-R	64M	6.4M	-	26	0.613	[LLHC23]
	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	61	0.613	[LCLH22]
KATAN64 [RK,C]	$\mathcal{ND}_{\text{Gohr}}$	128-1-δ-R	1280M	128M	-	86	0.728	[LCLH22]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

[RK] Related key setting.

[C] Conditional setting.

KATAN is a family of FSR-based block ciphers with block sizes 32, 48, or 64, key size 80, and 254 rounds.

For KATAN32, [BGHR23] reached statistically significant accuracies up to 69 rounds in an automatically generated distinguisher, and noted that this can be improved to a 71-round distinguisher with 0.5034 ± 0.0002 accuracy using their simple polishing step. In contrast, [LCLH22] reached 51 rounds in the standard setting and 59 when using 64 pairs.

A.12 KNOT

KNOT is an SPN-based permutation acting on a 256, 384, or 512-bit state; when used in a MonkeyDuplex construction to build a cipher, it uses 28 to 52 rounds, depending on the version.

A.13 LBCIoT

LBCIoT is a 32-round block cipher encrypting 32-bit plaintexts with an 80-bit key.

Table 18: Overview of the Neural Differential Distinguishers for KNOT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
KNOT-256	MLP	2-2-δ-D	1.6M	1.6M	-	10	0.5912	[BBCD21]
KNOT-512	MLP	2-2-δ-D	1.6M	1.6M	-	12	0.6032	[BBCD21]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

Table 19: Overview of the Neural Differential Distinguishers for LBCIoT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
LBC-IoT	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	7	0.6408	[TTJ23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

A.14 LEA

LEA is an ARX-based block cipher, encrypting 128-bit plaintexts with 128-, 192-, or 256-bit keys for 24, 28, or 32 rounds, respectively.

Table 20: Overview of the Neural Differential Distinguishers for LEA.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
LEA-128	DBitNet	2-1-CT-R	20M	2M	✓	11	0.512	[BGHR23]
	Transformer	2-1-A-R	10M	500K	-	13	0.5012	[BPC24b]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

We conclude this section on LEA with a critical discussion [BPC24b]. Bose et al. [BPC24b] claim improvements in distinguishing additional encryption rounds by sequentially training models on ciphertext pairs. These results should be interpreted with caution. From a cryptographic perspective, distinguishing advantage is expected to decrease as the number of rounds increases; however, this monotonic degradation is not reflected in their reported results. Moreover, the findings are not fully consistent with prior empirical evidence: Bellini et al. [BGHR23] report stronger distinguishers for rounds 9 and 10 of HIGHT, which calls into question whether the proposed architecture is in fact capturing the relevant differential patterns more effectively.

Given these theoretical and empirical inconsistencies, we treat the results of [BPC24b] as anomalous unless they are supported by sufficient statistical evidence. In line with the criterion used throughout this section, we consider a claimed distinguishing advantage statistically conclusive only when the random-guessing baseline falls outside the corresponding 95% confidence interval. For an effective test set of 2.5×10^5 ciphertext pairs, this means that only reported accuracies above approximately 0.50196 are treated as statistically significant and included in the quantitative comparison.

A.15 PRESENT

PRESENT is an SPN-based block cipher, encrypting 64-bit blocks with 80 (PRESENT-80) or 128-bit keys (PRESENT-128) for 31 rounds.

We conclude this section on PRESENT with a critical discussion [BPC24b]. Bose et al. [BPC24b] claim improvements in distinguishing additional encryption rounds by

Table 21: Overview of the Neural Differential Distinguishers for PRESENT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
PRESENT-64/80	$\mathcal{ND}_{\text{Gohr}}$	8-1-CT-R	20M	2M	-	7	0.5853	[CSYY23]
	UNet	12-1-A-R	-	-	-	7	0.664	[ZLHH25]†
	DBitNet	2-1-CT-R	20M	2M	✓	8	0.512	[BGHR23]
	$\mathcal{ND}_{\text{Gohr}}$	2-2-δ-D	20M	2M	-	8	0.515	[WWH21]
	INC	32-1-CT-R	960M	32M	-	8	0.5416	[ZWC23]
	LSTM	2-1-A-R	10M	500K	-	12	0.5014	[BPC24b]†
PRESENT-64/80 [RK]	MLP	6-1-δ-R	4.2M*	1.9M*	-	5	0.614	[PSM24]
	$\mathcal{ND}_{\text{Gohr}}$	2-2-δ-D	20M	2M	-	10	0.517	[WWH21]

Class: n-m-T-E, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings n-m-T-E are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
[RK] Related key setting.

sequentially training models on ciphertext pairs. These results should be interpreted with caution. From a cryptographic perspective, distinguishing advantage is expected to decrease as the number of rounds increases; however, this monotonic degradation is not reflected in their reported results. Moreover, the findings are not fully consistent with prior empirical evidence: Bellini et al. [BGHR23] report stronger distinguishers for rounds 9 and 10 of HIGHT, which calls into question whether the proposed architecture is in fact capturing the relevant differential patterns more effectively.

Given these theoretical and empirical inconsistencies, we treat the results of [BPC24b] as anomalous unless they are supported by sufficient statistical evidence. In line with the criterion used throughout this section, we consider a claimed distinguishing advantage statistically conclusive only when the random-guessing baseline falls outside the corresponding 95% confidence interval. For an effective test set of 2.5×10^5 ciphertext pairs, this means that only reported accuracies above approximately 0.50196 are treated as statistically significant and included in the quantitative comparison.

A.16 PRIDE

PRIDE is a 20-round SPN cipher using 64-bit blocks and 128-bit keys.

Table 22: Overview of the Neural Differential Distinguishers for PRIDE.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
PRIDE	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-A	50K	50K	-	6	0.545	[PMDC22]†
	MLP	2-1-δ-R	2.1M	300K	-	20	1	[MPKM+22]†

Class: n-m-T-E, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings n-m-T-E are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

We conclude this section on PRIDE with a critical discussion [MPKM+22, PMDC22]. In [MPKM+22], the authors claim a full round distinguisher on GIFT-64 with over 90% accuracy, using 2^{20} samples in total, of which 15% are kept for validation, respectively testing, and a MLP architecture; they also claim a full round distinguisher on PRIDE with 100% accuracy. Full-round attacks on modern and reputable ciphers are an extraordinary claim and require extraordinary evidence, which the author's manuscript does not provide. In [PMDC22], the authors use a validation set of only 25K samples. For $v = 25$K, the worst-case standard error is about 0.32%, giving a two-sided 95% confidence interval of roughly $\pm 0.62\%$.

A.17 SHA3

SHA3-256 is a 24-round sponge-based hash function with an output size of 256.

Table 23: Overview of the Neural Differential Distinguishers for SHA3.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SHA3-256	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	2M	-	3	0.9904	[CSYY23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).

A.18 SKINNY

SKINNY is an SPN-based block cipher; SKINNY128 processes 128-bit plaintexts with 128, 256, and 384-bit keys for 40, 48, and 56 rounds, respectively.

Table 24: Overview of the Neural Differential Distinguishers for SKINNY.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SKINNY128$^{\text{Unkeyed}}$	Classical ML	2-2-δ-D	32K	32K	-	6	0.9912	[BBD$^+$23]†
	Classical ML	2-2-δ-D	2M	2M	-	7	0.5456	[BBD$^+$23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

We end this section on SKINNY with a critical discussion of [BBD$^+$23]. Baksi et al. [BBD$^+$23] use only 16K ciphertext pairs in their validation set. For $v = 16$K, the worst-case standard error is about 0.40%, giving a two-sided 95% confidence interval of roughly $\pm 0.77\%$.

A.19 SLIM

Table 25: Overview of the Neural Differential Distinguishers for SLIM.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SLIM	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	200K	-	3	0.5036	[TTJ23]†
	$\mathcal{ND}_{\text{Gohr}}$	2-1-CT-R	2M	-	-	5	0.814	[RKS23]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
/ Unknown quantity.
† A critical discussion of these results is provided in the text.

SLIM is a 32-round block cipher encrypting 32-bit plaintexts with an 80-bit key. In [RKS23], the authors performed experiments with low key entropy (10 and 100 keys, respectively, for 1M samples), as well as with one random key per sample. We report the last one for comparability and note that the results were very close in the 3 cases.

We end this section on SLIM with a critical discussion of [RKS23, TTJ23]. Rajakumar et al. [RKS23] do not specify the size of their validation set, making it impossible to assess the statistical reliability of their reported accuracy. In [TTJ23], the reported accuracy

is 0.5036 on 10^5 validation samples. For $v = 10^5$, the worst-case standard error is about 0.16%, giving a two-sided 95% confidence interval of roughly ±0.31%. Thus, the reported distinguishing advantage is only marginally statistically significant at the 95% confidence level

A.20 TEA and XTEA

Table 26: Overview of the Neural Differential Distinguishers for TEA and XTEA.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
TEA	MLP	2-1-CT-R$^+$	2M	20K	-	8	0.545	[BR21]†
	DBitNet	2-1-CT-R	20M	2M	✓	10	0.563	[BGHR23]
XTEA	DBitNet	2-1-CT-R	20M	2M	✓	10	0.598	[BGHR23]

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

TEA and its successor, XTEA, are 64-round block ciphers that encrypt 64-bit plaintexts with a 128-bit key.

We conclude this section on TEA and XTEA with a critical discussion [BR21]. The authors use a validation set of only 10K ciphertext pairs. For $v = 10$K, the worst-case standard error is about 0.50%, giving a two-sided 95% confidence interval of roughly ±0.98%.

A.21 TinyJAMBU

TinyJambu-128 is an authenticated encryption algorithm based on a 640-round NLFSR-based permutation, which encrypts 128-bit blocks. TinyJambu-128 was among the ten NIST's lightweight cryptography finalists.

Table 27: Overview of the Neural Differential Distinguishers for TinyJAMBU.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
TinyJAMBU-128	MLP	2-1-δ-R	2.097M	262K	-	FULL	0.9958	[SSL$^+$22]†

Class: *n-m-T-E*, from Subsection 8.2. Under this convention, Gohr's initial experiments are 2-1-CT-R, and the results obtained in greyed out settings *n-m-T-E* are not directly comparable. **AutoND:** indicates if the neural distinguisher was automatically generated (✓) or is the result of an elaborate, manually designed training procedure (-).
† A critical discussion of these results is provided in the text.

We conclude this section on TinyJAMBU with a critical discussion [SSL$^+$22]. In [SSL$^+$22], the authors claimed a full-round distinguisher on TinyJambu, which we challenge. In the provided code, the ciphertexts in a sample use the same key, nonce, and associated data, which would provide a trivial distinguisher. As noted by the designers of TinyJambu[9]: 'When nonce is reused, an attacker can decrypt the ciphertext since the encryption of TinyJAMBU is somehow similar to the Cipher Feedback mode.'.

[9] https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf