# SoK: 6 Years of Neural Differential Cryptanalysis

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

[1] Technology Innovation Institute, Abu Dhabi, UAE {name.lastname}@tii.ae
[2] Technical University of Darmstadt, Cryptoplexity moritz.huppert@tu-darmstadt.de
[3] Radboud University, The Netherlands stjepan.picek@ru.nl

**Abstract.** At CRYPTO 2019, A. Gohr introduced Neural Differential Cryptanalysis and used deep learning to improve the state-of-the-art cryptanalysis of 11-round SPECK32. As of February 2025, according to Google Scholar, Gohr's article has been cited 229 times. The variety of targeted cryptographic primitives, techniques, settings, and evaluation methodologies that appear in these follow-up works grants a careful systematization of knowledge, which we provide in this paper. More specifically, we propose a taxonomy of these 229 publications and systematically review the 66 papers focusing on neural differential distinguishers, pointing out promising directions. We then highlight future challenges in the field, particularly the need for improved comparability of neural distinguishers and advancements in scaling.

**Keywords:** Neural Differential Cryptanalysis, Systematization of Knowledge

## 1 Introduction

The security of most digital applications relies on cryptography, the science of protecting the integrity, authenticity, and confidentiality of data. Confidentiality is about ensuring that only intended parties can read exchanged data. Typically, an *encryption key* is used in a secure *cipher* algorithm to encrypt the *plaintext* into a *ciphertext*. The recipient, knowing the *decryption key*, can easily retrieve the plaintext from this ciphertext. On the other hand, it is computationally intractable for an *adversary* who does not know the key.

The cornerstone of symmetric cryptography, where the encryption and decryption keys are the same, is *block ciphers*, which encrypt fixed-size messages, usually through iterations of a simple round function. Block ciphers play an important role in confidentiality but can also serve as building blocks to construct other primitives, such as hash functions and MAC schemes. Therefore, the security analysis of block ciphers (or cryptanalysis) is a crucially important field.

In the classical security notion, Pseudo Random Permutation (PRP) security, an adversary algorithm is assumed to have black-box access to an oracle function, implementing either (A) the studied block cipher (with a hidden, random key) or (B) a random permutation. A block cipher is considered secure under this notion if no such adversary can distinguish between situation A and B faster than using the trivial strategy of enumerating all possible keys to find one that matches the oracle's output (A) or be convinced that no such key exists (B). On the other hand, if a *distinguisher* exists, the block cipher is considered broken, as a good distinguisher can usually be used to retrieve the key. The performance of a distinguisher is usually expressed in terms of *time complexity* (number of operations to be performed by the attacker), *data complexity* (amount of queries to the oracle), and *memory complexity*.

The main goal of cryptanalysis is to estimate how many iterations of the round functions (or *rounds*) are needed for security. This is an iterative process, and new results continue to be published regularly years after the release of a cipher. Therefore, cryptographers are

eager to build and improve tools that help with this tedious task. Deep learning, due to its strength at detecting and distinguishing patterns, has long been seen as a potential candidate to assist the task of cryptographers.

Deep learning has experienced significant advancements in recent years, leading to remarkable achievements in various domains. Initially, Frank Rosenblatt introduced Multi-Layer Perceptrons (MLPs) in his book Perceptron in 1958 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 computational intelligence in cryptanalytic tasks has remained limited. Earlier approaches typically relied on extensive precomputation [PPS14], exploited implementation flaws (e.g., side-channel attacks) [RD20], targeted inherently weak cryptographic schemes [Gre17, GHZ+18], or generally proved ineffective [CLC12a]. It was not until Gohr's seminal work [Goh19a], presented at CRYPTO 2019 that a breakthrough was achieved by combining deep learning with traditional cryptanalytic techniques. Gohr's work was the first to demonstrate that neural networks could be successfully leveraged in cryptanalysis, producing attacks that improved upon state-of-the-art techniques against a round-reduced version of a modern block cipher.

Gohr pioneered the application of differential cryptanalysis—a powerful technique for analyzing block ciphers—to neural networks, creating an approach now termed neural differential cryptanalysis. First introduced by Biham and Shamir in 1991 [BS91], differential cryptanalysis examines how input differences ($\delta$) propagate through ciphers, seeking high-probability differentials where specific plaintext differences yield predictable ciphertext differences ($\Delta$). While modern ciphers are designed to resist such analysis, Gohr's work [Goh19a] demonstrated that deep neural networks could serve as statistical distinguishers with superior accuracy compared to conventional methods, particularly on the NSA-designed SPECK32 [BTCS+15] reduced to 8 rounds, while significantly reducing time complexity for 11-round key recovery attacks. This breakthrough challenged conventional wisdom by showing neural networks could discriminate between ciphertext pairs derived from fixed versus random input differences more effectively than traditional approaches, despite the black-box nature of neural computation.

Figure 1 a) shows the basic scheme for a neural differential distinguisher experiment as introduced by Gohr in [Goh19a]. Figure 1 b) gives a broad overview of the research directions in Neural Differential Cryptanalysis: Researchers have explored every part of the basic pipeline. A majority of the works citing Gohr that focus on neural differential distinguishers have attempted to apply the scheme to **other symmetric primitives** and improve the distinguishing advantage by **changing the network architecture**, **increasing the number of samples** available to the distinguisher, or **changing the sample format**. A classification taxonomy for neural cryptanalysis was introduced at FSE 2024 [BGH+23], categorizing approaches based on four key dimensions: the number of input ciphertexts $n$, the number of distinct input differences $m$ employed in the analysis, the feature engineering techniques $E$ applied to the ciphertext pair, and the distinguishing experiment type $T$ being conducted. This framework systematically compares neural cryptanalytic methods and clarifies their relative strengths across attack scenarios.

Finally, we also observe emerging research directions aimed at enhancing neural distinguishers across multiple dimensions: increasing **automation** of the attack pipeline,

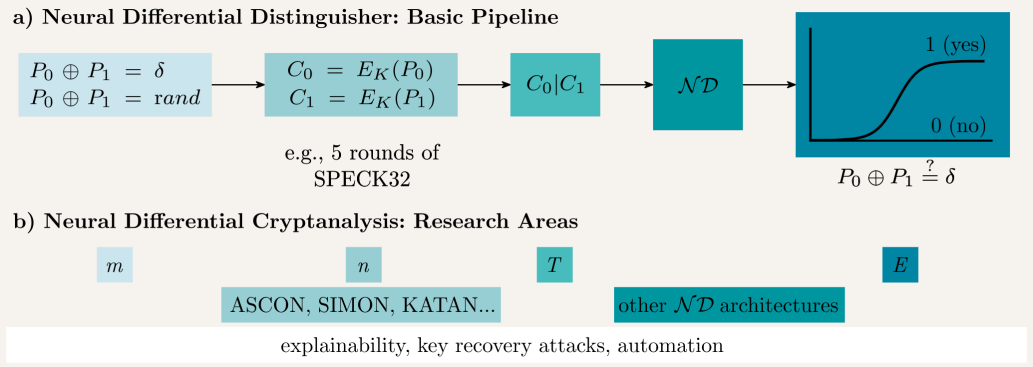


**Figure 1: a) Neural Differential Distinguisher: Basic Pipeline.** Start with two plaintext $P_0, P_1$, where $P_0 \oplus P_1 = \delta$ or $P_0 \oplus P_1 = \text{rand}$. Encrypt them using a symmetric key $K$ to obtain ciphertexts $C_0, C_1$. Concatenate the ciphertexts $C_0|C_1$ and input them into a neural distinguisher $\mathcal{ND}$. The neural distinguisher's output is a neuron with a sigmoid activation function. The sigmoid curve indicates a binary decision output to answer if $P_0 \oplus P_1 \stackrel{?}{=} \delta$. **b) Neural Differential Cryptanalysis: Research Areas.**

improving **transparency** through explainability techniques that reveal the cryptographic features being learned, and boosting **effectiveness** in practical key recovery attacks.

While a substantial body of literature has focused on developing and analyzing effective neural differential distinguishers, this paper is, to the best of our knowledge, the first to systematically organize a large collection of research (229 papers) and highlight promising directions and challenges in this area. Recent surveys [BHR+22, NR23, CLC12b, MLR+23, SST24] 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 [Goh19a] along with 15 related follow-up studies, though these were not selected systematically and include preprints. 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 on computational intelligence applications in cryptography, with a focus on the automated design and cryptanalysis of ciphers. However, their work predates Gohr's introduction of differential machine learning-based cryptanalysis in [Goh19a], 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 [Goh19a] 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+23] 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.

The explosive growth of neural cryptanalysis literature has created a body of work that is too vast for any single researcher to review comprehensively. This absence of systematic knowledge organization has fostered several troubling trends: research teams independently

investigating nearly identical questions, sometimes reaching contradictory conclusions (exemplified by [Seo24]'s false claim of developing the first truncated neural distinguisher and the significant disagreements on architecture suitability among [BR21], [SSL+22], and [BBCD22]), and fundamental misinterpretations of seminal concepts – particularly regarding the aggregation of multiple ciphertext pair predictions, an issue explicitly addressed by Gohr [GLN22]. As this field continues its rapid expansion, these problems will only intensify, underscoring the critical need for a comprehensive systematization of knowledge in neural cryptanalysis.

**Our Contributions.** In our systematization of knowledge, we have achieved the following:

1. **Comprehensive Field Review:** We conducted an exhaustive survey of the follow-up work (Section 4). In this process, we have identified the **full body of research** in the field of Neural Differential Cryptanalysis. We analyzed the directions of the field, resulting in a detailed taxonomy of Neural Differential Cryptanalysis (Section 4).
2. **Explainability and Key Recovery Overview:** We provide a comprehensive overview of recent advancements in explainability techniques using neural differential distinguishers (Section 5). Since analyzing neural distinguishers constitutes the core contribution of our work, we provide a comprehensive overview of advancements in neural-aided key recovery in Appendix C as a contextual application of our findings.
3. **Rigorous Classification and Comparison:** We systematically classify and compare peer-reviewed research outcomes on neural differential distinguishers (Section 6), across various techniques, architectures, and primitives. We also identify promising research directions and severe methodological issues in some peer-reviewed papers and challenge their results.
4. **Best Practice Recommendations:** Evaluating research involving the training of neural networks presents significant challenges. We have developed a comprehensive set of best-practice guidelines specifically tailored for reviewers of Neural Differential Cryptanalysis research (Section 7).
5. **Future Challenges:** We identify and discuss two major challenges set to shape the next six years of neural cryptanalysis (Section 8).

## 2 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 [Goh19a]. However, even in that work, the author mentioned a number of related works at the intersection between 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 provide a brief historical overview of various approaches.

Already in 1947, researchers started considering connections between cryptography and artificial intelligence [Wea47]. While this attempt was devoid of any technical details, it still showcases the interest of the scientific community in combining these two domains. In 1984, L. Valiant discussed learnable Boolean functions and mentioned the evidence from cryptography 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 [MP88]. In 1994, 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 verify 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 employed neural networks in a framework inspired by generative adversarial networks (GANs) to develop an encryption scheme [CdOAB+18a]. Although this early research did not show any formal security, Coutinho *et al.* demonstrated in 2018 [CdOAB+18b] 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 to conduct nonlinear cryptanalysis and applied it to the MARS cipher 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 [CLC12b]. On the other hand, Svenda *et al.* in 2014 used evolutionary algorithms to construct empirical tests for randomness [SSUM14]. Finally, in 2017, Awad and El-Alfy surveyed computational intelligence applications in cryptography, focusing on the automated design and cryptanalysis of ciphers [AEA17].

# 3 Preliminaries

This section introduces key concepts in machine learning-assisted differential cryptanalysis: conventional differential cryptanalysis (Subsection 3.1), deep learning applications (Subsection 3.2), and neural network-aided key recovery (Subsection 3.3).

## 3.1 Differential Cryptanalysis

Differential cryptanalysis [BS91] is a chosen plaintext attack analyzing how plaintext perturbations propagate through ciphers. While typically using bitwise XOR differences, 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:

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

## 3.2 Training Neural Differential Distinguishers

For plaintexts $p_1, p_2 \in \{0,1\}^b$ with ciphertexts $c_i = F(p_i) \in \{0,1\}^b$, neural distinguishers approximate the function for fixed difference $\delta \in \{0,1\}^b$:

$$Y(c_1||c_2) = \begin{cases} 1, & \text{if } p_1 \oplus p_2 = \delta, \\ 0, & \text{else.} \end{cases}$$

Success requires identifying nonrandom properties in output distributions resulting from input difference $\delta$. Training typically uses balanced datasets: 50% samples $(c_1, c_2, 0)$ with random $p_1, p_2$, and 50% samples $(c_1, c_2, 1)$ where $p_2 = p_1 \oplus \delta$. Networks are trained via stochastic gradient descent [RM51][1] using loss functions such as mean squared error.

[1]We introduce essential machine learning terminology needed to understand the techniques used in neural differential cryptanalysis: Stochastic gradient descent is an iterative optimization method that updates the weights of a neural network by calculating error gradients on small random subsets ("batches") of the training data rather than the entire dataset. The "loss function" (e.g., mean squared error) quantifies

Following Gohr [Goh19a], effective implementations use approximately $10^7$ training samples, $10^6$ testing samples, batch sizes around 5000, and up to 200 training epochs. Performance enhancements often include Adam optimizer [KB15], L2 regularization [HK00], and cipher-specific architectures [GLN22, BGH+23].

### 3.3 Neural-aided Key Recovery

Neural distinguishers $ND$ that approximate $Y(c_1||c_2)$ enable practical key recovery attacks on block ciphers, demonstrating their concrete cryptanalytic value. This section outlines the attack methodology based on Gohr's seminal approach [Goh19a], which has become the standard framework in subsequent research. We denote the $r$-round reduced block cipher with secret key $K$ as $F_K^r$.

**The Basic Attack** The attack leverages a pre-trained neural distinguisher $ND_r$ for $F^r$ to compromise $F_K^{r+1}$ with secret key $K$. For example, a 5-round SPECK32/64 distinguisher enables attacks on 6-round SPECK32/64.

The attack targets the last round key $k_{r+1}$ in round-based ciphers that use function $f_k$ with keys $k_1, \ldots, k_{r+1}$ derived from master key $K$ and begins by querying the oracle $F_K^{r+1}$ with a conforming pair $p_1$ and $p_2 = p_1 \oplus \delta$, obtaining ciphertext pair $(c_1, c_2)$. Next, for some random key guess $k'$, the attacker computes $c_i' = f_{k'}^{-1}(c_i)$ and evaluates $R = D_r(c_1', c_2')$. We rank key candidates by prediction score, as the correct key yields $R \approx 1$ (the distribution matches what $ND_r$ was trained to recognize), while incorrect keys produce $R \lesssim 1$.

For SPECK32, where round keys (16 bits) are smaller than the master key (64 bits), we can feasibly enumerate all candidates. After identifying the last round key, the process can be repeated to recover earlier round keys until the entire sequence is reconstructed.

Our simplified attack explanation omitted that prediction scores exhibit variance, which can be mitigated by aggregating scores across multiple ciphertext pairs for each key candidate, thereby enhancing the statistical reliability of the distinguisher. In [Goh19a] the responses for a given key guess $k'$ are aggregated into a single score by the equation:

$$s_{k'} = \sum_{i=1}^{n} \log_2 \left( \frac{R_i^{k'}}{1 - R_i^{k'}} \right),$$

where $R_i^k$ represents the distinguisher's response for the $i$-th ciphertext pair.

We discuss various optimizations of this basic attack in Appendix B, including round extension via probabilistic differentials, computational cost reduction through Bayesian Optimization, and stopping conditions via Upper Confidence Bounds.

## 4 Neural Differential Cryptanalysis: A Taxonomy of Research Directions

### 4.1 Selected Literature

As of February 03, 2025, a total of 229 works cite Gohr's work [Goh19a] on Google Scholar. Among these, we discarded 4 references that were either redundant or not linked to a paper and 33 that were not available in English. Additionally, 34 references are not peer-reviewed, and only available as preprints [BBCD20, BBDH21, BBC+23, BGL+21, DNS24, ElS21, GJS20, GLN22, Goh22, HRC21a, HRC21b, HRC21d, JKM20, JKM21, Jun05, KJL+22,

prediction error, while an "epoch" represents one complete pass through the training dataset. The Adam optimizer is an advanced gradient descent variant that adapts learning rates individually for each weight. L2 regularization prevents overfitting by penalizing large weight values, essentially constraining the model's complexity.

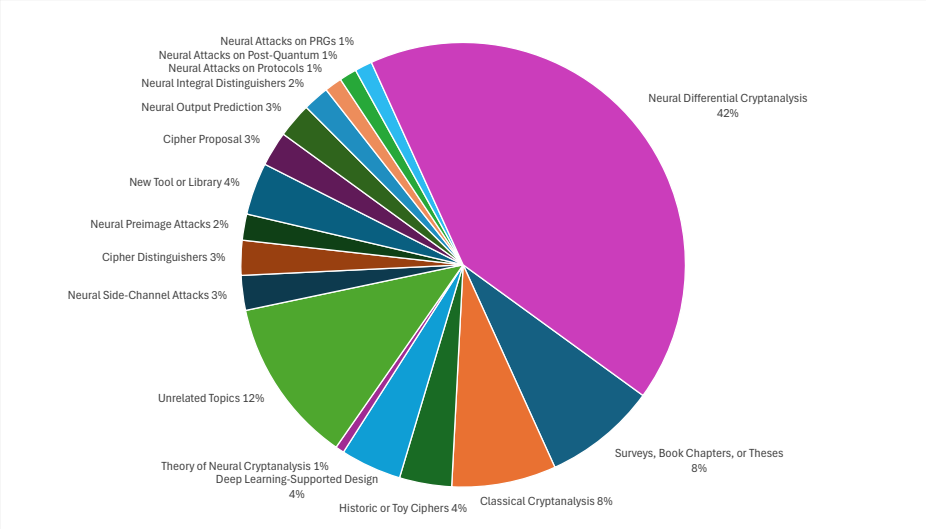


**Figure 2:** Taxonomy of the peer-reviewed publications in English citing [Goh19a].

KVD+25, LLL+22, LTJ+20, LSW+23, LRC22, PMC+22, PMK20, SM23a, SS23, SWL+24, Sug24b, WNB+23, YW24, ZL20a, ZL20b, ZW22, ZZW24, ZDW+23]. After excluding these, we are left with 158 peer-reviewed references, which we systematically categorize as shown in Figure 2.

We consider the following references outside the field of research on Neural Differential Cryptanalysis as they are surveys, overviews, theses, or book chapters that treat the use of "ML in cryptography" [BHR+22, Bru21, CDS22, MLR+23, NMN24, NR23, PJ21, PJ22, Ros24, Som23, Tan23, Tu22, ZG24], or their research focuses on other topics such as: classical cryptanalysis [Bak21, BCdST+23, ELR20, FLW+23, GPT21, KS22, KY21, PLH+24, SLL24, WW22, YK21b, YK22], the theory of Neural Differential Cryptanalysis [SMR+24], cryptanalysis of historic or toy ciphers [GZDAL22, KSJS21, KLK+23, LMK+21, PKM23, PMDC22], deep learning-supported design of cryptographic algorithms [AKVS+24, CS21, HLG+23, ITYY21, LTJ+21, MJBHC22, WIO24], neural side-channel attacks [GJS21, TDD22, YBBP23, ZZC+22], distinguishers between different ciphers [BPS22, DM23, MPM+21, XLC+22], neural preimage attacks [JMTD22, LLL+21, PTD22a], the introduction of a new tool or library [BGG+23, Ess23, Hal22, ITY25, LJSC24b, PVM24], the proposal of a new cipher [BSL21, CDJ+21, DCW23, FAAQ24], neural output prediction attacks [JM24, KEI+22, KEI+23, LF24], neural integral distinguishers [HLLL, WG24, ZL22], neural attacks on protocols [TD21, ZKL20], post-quantum schemes [LWAZ+24, WCCL22], pseudorandom number generators [Boa24b, EAZD23], or other unrelated topics [AABAA22, AAEK22, Boa24a, DDK+23, DZF+21, PTD22b, HLZW20, KLJW23, Kar23, MGKMP21, MKMP21, MLYW22, PYW24, RRSM22b, So20, Sug24a, TDF+22, ZZS21, ZLF+24], which leaves us with a total of 66 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 [BR21, BGPT21, BBCD22, BGL+22, BBP22, BBD+23, BLYZ23, BGH+23, BPRC24, BFG+24, BPC24, CSY23, CSYY23, DCC23, ERP22, EGP23, GLP+24, HRC21c, HGH+23, HRC23, HLF+24, HLZH25, KJL+23, KKJ+24, LCLH22, LTZ22a, LTZ22b, LLHC23, LRC23, LRCL23, LLS+24, LRC24, LJSC24a, MPKM+22, PPWR23, PSM23, PCDC24, RRSM22a, RLS23, SZM21, SSL+22, SM23b, SCL24, SSL+24, SBG+24, Seo24, TH21, TTJ23, TSL23, WW21, WWH21, WTZ+22, WQW+24, WW24b, WWS24, WW24a, YK21a, YW23, YK24, ZLHH25, ZZY+21, ZZ21, ZLWL23, ZWC23, ZWW24, ZWL24]

## 4.2 Taxonomy of Research Directions

We found contributions to the explainability (or interpretability) of neural distinguishers in the following 17 works [BGPT21, BBP22, BLYZ23, CSY23, DCC23, ERP22, Goh19a, GLP+24, HGH+23, HLF+24, LRC23, LRC24, LJSC24a, SCL24, Seo24, YW23, ZWL24], and will discuss their respective contributions in Section 5.
We found contributions to neural-aided key recovery attacks in the following 22 works [BGL+22, BLYZ23, CSY23, CSYY23, Goh19a, HRC23, HLF+24, LTZ22a, LCLH22, LLHC23, LRC24, LJSC24a, SZM21, Seo24, TH21, TTJ23, WQW+24, YW23, ZWW24, ZLWL23, ZLHH25, ZWL24], and will give an overview of these works in Appendix C.

Most (62/66) of peer-reviewed research on Neural Differential Cryptanalysis involves training neural differential distinguishers. More precisely, neural differential distinguishers are trained in [BGPT21, BR21, BBCD22, BGL+22, BBP22, BGH+23, BBD+23, BLYZ23, BFG+24, BPC24, 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, LJSC24a, MPKM+22, PSM23, RRSM22a, RLS23, SZM21, SSL+22, SM23b, SSL+24, SCL24, Seo24, SBG+24, TH21, TTJ23, TSL23, WW21, WWH21, WTZ+22, WQW+24, WW24b, WWS24, WW24a, YK21a, YW23, YK24, ZZY+21, ZZ21, ZLWL23, ZWW24, ZWC23, ZLHH25, ZWL24]

A comparative review of peer-reviewed neural differential distinguishers is provided in Subsection 6.3. We excluded papers that were inaccessible [BPRC24, PCDC24], focused primarily on explainability [SCL24, GLP+24], lacked concrete accuracy measurements [HRC23], utilized leakage models outside conventional security assumptions [SBG+24, HLZH25], or prioritized input difference compatibility over distinguisher performance in hybrid approaches [YK21a, WW24b].

Two recent papers [HLF+24, LRC24] investigating neural differential attacks on large-state block ciphers predominantly emphasize key recovery methodologies while providing limited insights into their neural network training processes. This methodological opacity presents significant challenges for our comparative review. However, Huang *et al.* [HLF+24] provide their implementation, enabling a more comprehensive evaluation of their approach despite the initial presentation's technical brevity.

Gohr's analysis was performed within the secret key chosen-plaintext attack (SK/CPA) model. We do not consider the work of Phan *et al.* [PPWR23] as it operates under a fundamentally different adversary model, where generative AI techniques are trained in an adaptively chosen ciphertext or known key scenario to distinguish 10-round SPECK32/64, making direct comparison inappropriate.

## 5 Overview: Neural Differential Distinguisher Explainability

Neural distinguishers enabling new attacks, potentially better than manual cryptanalysis, motivated researchers to try to understand what made these attacks so powerful and to learn new properties from these. 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 is an active research field in AI and has resulted in various ways to add some explainability to a neural network, e.g., by pruning, ablation studies, or visualization techniques.

A. Gohr investigated the capabilities of provided neural networks by introducing a differential cryptanalytic task called the real differences experiment [Goh19a]. Then, the author looks at the importance of features and gives some evidence that the neural distinguishers exploit features outside the difference distribution table.

In [BGPT21], Benamira *et al.* studied the properties of pairs that were correctly classified and proposed that Gohr's neural distinguishers learn differential-linear features. In particular, the authors observed that the pairs for which the score of the neural distinguisher at round 5 is high often follow a specific truncated differential pattern at round 3; a similar observation is made for rounds 6 and 4, leading to the authors proposing that the features learned by the neural distinguisher are differential-linear in nature. The authors further modified the neural network to use a Heaviside activation function, which forces its output to be 0 or 1, to study the Boolean functions learned on SPECK. From these, they derived advanced features that could be used to replace the initial 1D convolutions of Gohr's network. Later, the truncated differential observations from [BGPT21] were used by [BGH+23] to identify good input differences for neural distinguishers automatically.

In [BBP22], Bacuieti *et al.* further investigated the structure of the neural network itself. In particular, the authors used the *lottery ticket hypothesis* to prune Gohr's neural network to a minimal working version, on which they used feature visualization techniques to obtain a visual representation of the neural network's behavior. They additionally show that, for the case of SPECK32, there is no significant accuracy difference between the depth 1 neural network and the depth 10 version for Speck reduced to 7 and 8 rounds.

Ablation studies are routinely performed for neural networks to understand their sensitivity and fidelity under small perturbations on either the network itself or its input data. Ablation studies can give insights into the explainability of neural network models, as detailed, for example, in *"BASED-XAI: Breaking Ablation Studies Down for Explainable Artificial Intelligence"* [HSB+22], or *"Logic Rule Guided Attribution with Dynamic Ablation"* [ALH22]. In [YW23], Yue *et al.* performed a data ablation study to observe trade-offs between improved accuracy and overfitting when using multiple ciphertext pairs per sample for neural differential distinguishers.

Seok *et al.* [SCL24] investigated 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 the datasets associated with input differences leading to successful distinguishers tend to have more axes that effectively represent the data compared to other datasets. Similarly, these datasets form multiple high-density clusters compared to only a single cluster in the shape of a sphere. 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 [Goh19a].

Bao *et al.* developed explicit rules to be used alongside a differential distinguisher to enhance its effectiveness and more closely match the performance of advanced neural distinguishers [BLYZ23]. The rules are based on strong correlations between bit values in

the right pairs of XOR-differential propagation through addition modulo $2^n$. The authors also showed that those rules can be closely linked to the previous studies of the multi-bit constraints and the fixed-key differential probability. Finally, the authors concluded that leveraging the value-dependent differential probability makes it possible to add additional knowledge to purely differential distinguishers. In contrast, they demonstrate that neural differential distinguishers inherently utilize these rules. Building on this observation, Lv *et al.* [LJSC24a] trained a neural distinguisher on differential-linear cryptanalysis.

Deng *et al.* introduced the attention mechanism into the differential cryptanalysis on SPECK [DCC23]. The authors used a visualization algorithm to demonstrate the effectiveness of the attention mechanism and further analyze the features extracted from the ciphertext by deep learning. With this visualization technique, the authors evaluate which bits the attention mechanism focuses most, providing interpretability results.

Recent advances in neural distinguishers [ERP22, GLP+24, HLF+24, LRC24, LRC23, HGH+23, CSY23, Seo24, ZWL24] have demonstrated remarkable efficiency by operating on partial ciphertext information rather than complete outputs. These approaches have simultaneously advanced cryptographic interpretability methods through systematic identification of the most influential ciphertext bits. Chen *et al.* [CSY23] introduced "Informative Bits" and Bit Sensitivity Testing, formally defining informative bits as ciphertext bits that effectively distinguish between a cipher and a pseudo-random permutation. They successfully maintained high distinguisher performance for SPECK32/64 while omitting 16 of 32 ciphertext bits through their novel testing methodology.

Hambitzer *et al.*'s [HGH+23] deep learning ensemble (NNBits) provided bit-profiling capabilities specifically designed for evaluating cryptographic (pseudo) random bit sequences. Their work notably contributed to explaining the accuracy obtained by Gohr's depth-1 neural distinguisher in round 6 for SPECK32/64 by providing a detailed bit-level analysis. Liu *et al.* [LRC23] performed a comprehensive interpretability analysis exploring the relationship between neural distinguishers, truncated differentials, and advantage bits. Their advantage bit search algorithm successfully truncated ciphertexts to just 8 bits while leveraging XOR differences to reduce training sample size requirements significantly.

Similarly, Ebrahimi *et al.* [ERP22] presented a Partial Differential (PD) ML-distinguisher for SPECK32/64, achieving nearly identical accuracy with only 8 bits compared to full 32-bit distinguishers for six rounds of the cipher. Goi *et al.* [GLP+24] employed explainable AI techniques (LIME and SHAP) to examine Gohr's neural distinguisher, revealing significant methodological differences: LIME effectively captures individual bit significance, while SHAP uniquely identifies important bit pairings in the ciphertext.

Seok [Seo24] developed a specialized neural distinguisher for HIGHT that focuses exclusively on ciphertext bits produced by one of the two independent operations in the round function, demonstrating the viability of operation-specific analysis. Zhang *et al.* [ZWL24] extended neural cryptanalysis to AES-128, training distinguishers for 2-round reduced cipher and additionally examining specific intermediate states between rounds 2 and 3. Their approach replaced full 16-byte state processing with specialized networks operating on just 2-byte segments while maintaining nearly identical accuracy. Huang *et al.* [HLF+24] train partial neural distinguishers through extended encryption and strategic decryption with zero-set subkey bits, and Li *et al.* [LRC24] develop a sophisticated ensemble approach combining multiple student distinguishers, each strategically trained on input differences producing mostly distinct informative ciphertext bits.

## 6 Comparative Review: Neural Differential Distinguishers

In the following, we provide a comparative review of all trained neural differential distinguishers to date. First, all investigated neural network architectures are reviewed (Subsection 6.1), then we detail the classification scheme (Subsection 6.2) and conclude with

a comparative review of the best neural differential distinguishers for each symmetric primitive (Subsection 6.3), followed by a discussion of the review (Subsection 6.4).

## 6.1 Architectures

In this section, we review the neural network architectures[2] that have been employed in Neural Differential Cryptanalysis.

**$\mathcal{ND}_{\text{Gohr}}$** $\mathcal{ND}_{\text{Gohr}}$ is the original neural network architecture as introduced by Gohr in [Goh19a]. It consists of an initial reshaping that "mirrors the word-oriented structure of the cipher", a single bit-sliced convolution, a residual convolutional tower of different possible depths, most commonly depth-1, depth-5 and depth-10, and, finally, a fully connected prediction head. A staged training approach in combination with an elaborate additional training procedure is required to obtain the 8-round distinguisher for SPECK [Goh19a]. $\mathcal{ND}_{\text{Gohr}}$ has subsequently been used on the majority (14/24) of the primitives. A non-peer-reviewed work by Gohr, Leander, and Neumann [GLN22] provides a thorough investigation of relevant hyperparameters when adapting $\mathcal{ND}_{\text{Gohr}}$ to a new primitive. Variants of Gohr's original network have been created: $\mathcal{ND}^{\text{pruned}}_{\text{Gohr}}$ is a pruned version of $\mathcal{ND}_{\text{Gohr}}$ for SPECK introduced in [BBP22]. $\mathcal{ND}^{\text{attntn.}}_{\text{Gohr}}$ was introduced in [DCC23] and adds an attention mechanism to $\mathcal{ND}_{\text{Gohr}}$ and applies it to SPECK. [HGH+23] uses an ensemble of $\mathcal{ND}_{\text{Gohr}}$ ($\mathcal{ND}^{\text{ensmbl.}}_{\text{Gohr}}$) to explain the accuracy of Gohr's network on SPECK.
A variant of Gohr's network that uses a separable convolution instead of the traditional one ($\mathcal{ND}^{\text{sep.conv.}}_{\text{Gohr}}$) was introduced in [LRC23] and applied to SPECK with the motivation to save training cost. DenseNet is a variant of CNNs in which every convolutional layer is directly connected to *all following* downstream layers. It has been used by [SM23b] on SPECK-32.

**DBitNet** DBitNet was introduced in [BGH+23] as a "cipher-agnostic" neural network that aims to avoid SPECK-dedicated features of $\mathcal{ND}_{\text{Gohr}}$. It is based on *dilated convolutional* layers. In a dilated convolution, the convolution kernel is not learning dependencies between neighboring neurons but between neurons that are farther apart. In this way, DBitNet aims to avoid the input reshaping and bit-slicing convolution of $\mathcal{ND}_{\text{Gohr}}$. Notably, using a simple staged[3] training pipeline, and a simple additional polishing step, the same accuracy as Gohr is obtained for SPECK. It has been employed in [BGH+23] to generate distinguishers for seven primitives automatically (SPECK, SIMON, HIGHT, PRESENT, KATAN, TEA and XTEA, and LEA).

**Inception** In the Inception architecture (INC), a layer inspired by GoogLeNet's Inception module replaces one of the (convolutional) layers of the original $\mathcal{ND}_{\text{Gohr}}$ architecture. The Inception module consists of multiple parallel convolutional layers that process the module input using a variety of kernel sizes. This might allow for extracting features that could not be extracted with one specific kernel size at the cost of increased training times [GLN22].

[2] We introduce key vocabulary for Neural Differential Cryptanalysis architectures: MLPs use *densely* connected layers with full connectivity between neurons, resulting in many parameters. *Convolutional layers* (CNNs) apply filters to detect spatial patterns, requiring more computation but better capturing hierarchical features. *Inception modules* combine parallel convolutions with various kernel sizes for enhanced feature extraction. *Residual connections* (RESNets) create bypass paths to improve information flow during training. *LSTM* (a type of RNN) processes sequential data using memory cells to capture long-term dependencies. *Attention mechanisms* dynamically focus on relevant input portions, forming the basis of transformer networks.

[3] *Staged training* refers to the method to continue training the best $r-1$ round neural differential distinguisher in round $r$.

In [ZWW24], the authors first proposed to use an Inception-like module to train neural distinguisher by replacing the initial convolutional block of $\mathcal{ND}_{\text{Gohr}}$. Some follow up works [ZWC23, ZLWL23] use INC and obtain neural distinguishes for SIMECK, PRESENT, CHASKEY, and DES. [YW23] construct INC by replacing the convolutional layers in the residual blocks of $\mathcal{ND}_{\text{Gohr}}$ and applying this architecture to SPECK.

[BLYZ23] introduced the idea of staged training together with a partially frozen network ($\text{INC}^{\text{freeze}}$). The underlying idea for the freezing of particular layers is that "convolutional layers are viewed as feature extractors" (which can be reused in subsequent rounds and can therefore be frozen), while "fully connected layers are viewed as a classifier" (which have to be updated when training a new round).

**MLP** The MLP (Multi-Layer Perceptron) is a neural network architecture in which subsequent layers are densely connected. MLPs are often outperformed by residual networks and CNNs. However, they are generally computationally more lightweight, which motivates the application of the architecture to 11 of the 24 primitives to investigate their potential as a neural differential distinguisher.

**LSTM and Transformer** Long-short term memory cells (LSTMs) were used in [BBCD22, SSL+22] on GIMLI, TinyJAMBU, and GIFT. In [BPC24], Bose *et al.* build a distinguisher for LEA, PRESENT, and HIGHT by training an Encoder network, implemented either as a *Transformer* or an *LSTM*. For the *LSTM* variant, each ciphertext pair is embedded using a one-hot encoding scheme supplemented with positional encoding.

**Others.** In [KJL+23], the first quantum neural network based distinguisher (Quantum) is built for SPECK. SENet stands for Squeeze-and-Excitation network and was used for the first time as a neural differential distinguisher in [BGL+22]. SENet introduces a new building block for CNN that improves the finding of channel interdependencies at almost no computational cost. [BGL+22] applied SENet to SPECK and SIMON. SE-ResNet was first used as neural differential distinguisher by [LLS+24], motivated by "the success of $\mathcal{ND}_{\text{Gohr}}$ on SPECK [Goh19a] and SENet on SIMON [BGL+22]". [LLS+24] apply SE-ResNet to SIMON and SIMECK. Note that [BGL+22] also investigates DenseNet; it is, however, surpassed by SENet and, therefore, does not appear in the following compilation of best neural distinguishers.

We report Classical ML results, such as SVM in [BBD+23], on the rare occasion that they are competitive with neural distinguishers. For instance, the distinguishers developed by Zhang *et al.* [ZZ21] using classical machine learning methods – including AdaBoost, Random Forest, Extremely Randomized Trees, and Gradient Boosting Decision Trees– achieved accuracy rates that were consistently at least 20% lower than those obtained using convolutional neural networks.

In [ZLHH25], the distinguisher is built using a U-Net architecture consisting of encoding and decoding parts. This architecture is typically applied for image segmentation tasks and was used for the analyses presented in GIFT and PRESENT.

## 6.2 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.* [BGH+23] addressed this challenge by proposing a systematic classification framework based on four distinctive 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. We extend this framework with a taxonomy of symbols representing diverse network architectures and experimental methodologies, enabling a nuanced categorization.

### 6.2.1 Number of ciphertexts per sample: $n$

In [Goh19a], the scores output by a distinguisher trained to recognize single pairs are combined for multiple pairs during the key recovery process, increasing the strength of the signal and resulting classification accuracy. In [GLN22], the authors note that this notion was rediscovered in several papers and propose a score combining formula to transform a single pair classifier into a multiple pair classifier, while other works, such as [BGPT21], used the less effective score averaging. In [SSL+24], the authors proposed to replace scores aggregation with an MLP to classify based on the scores of multiple pairs.

In this classification, we consider the number of ciphertexts per sample used in the distinguisher training, independently of external scores aggregation through averaging or otherwise. This notion was introduced in [BGPT21], who built a neural distinguisher accepting multiple pairs at once. The Multiple Output Difference (MOD) format, introduced in [HRC21c], consists in concatenating not multiple pairs, but their respective differences, i.e., $C_0 \oplus C_1 || C_2 \oplus C_3 \ldots$. In [CSYY23], two different settings are explored: one where the $k$ pairs that form a sample share the same key and one where they do not. The authors note that compared to [GLN22], no additional features seem to be learned by gathering multiple pairs, compared to a single pair distinguisher and score aggregation. Here, whether one uses a unique key for each pair or reuses the same key for all pairs appears also to have no significant effect. In [ZWC23], the authors raised the question of the number of samples to use when the number of pairs per sample increases and consider two scenarios for training: one where the number of pairs is fixed to $10^7$ and one where the number of (multi-pair) samples is set to $10^7$. The authors concluded that fixing the number of pairs to $10^7$ (and hence obtaining a training set with $\frac{10^7}{n}$ entries) leads to overfitting, fluctuations in validation accuracy, and slow convergence of the model. This is confirmed by [ZWW24]. Finally, in e.g., [SSL+22], the authors considered polytopic samples with multiple input differences, where the used plaintexts are $(P, P \oplus \delta_0, P \oplus \delta_1 \ldots)$, effectively building $k$ relevant pairs from $k+1$ plaintexts. A similar technique using plaintext quadruples is referred to as mixture differential in [WQW+24].

### 6.2.2 Number of input differences: $m$

Baksi *et al.* [BBCD22] explored a setting where a set of $m$ input differences are considered. This setting was applied to various permutations: KNOT, ASCON, CHASKEY, and GIMLI, with $m = 2$ for GIMLI. Su *et al.* [SZM21] introduced a model called polytope neural differential network distinguisher. This model uses multiple differences, keeping one plaintext fixed among the differences and changing the other. In [WTZ+22], the authors proposed a multiple input difference scheme called $\text{ND}_{am}$, where the first ciphertext is the encryption of a random plaintext $P_0$, each subsequent ciphertext $C_i$ is the encryption of $P_{i-1} \oplus \Delta_{i-1}$ so that $n = m + 1$. The same scheme was used in [BBCD22]. In [BBD+23], the authors trained neural distinguishers using higher-order differentials.

### 6.2.3 Feature engineering type: $T$

Feature engineering is often used in machine learning to derive advanced features from the raw dataset, e.g., [GBC17]. A natural feature to use for neural differential cryptanalysis is to replace the ciphertext pairs ($T = \text{CT}$) by their XOR difference ($T = \delta$). This approach, used by works such as Baksi *et al.* [BBCD22], Zezhou *et al.* [HRC21c], and Yadav *et al.* [YK21a], simplifies the training process, at the cost of losing some information. Similarly, other works [LRC23, ERP22] truncated the ciphertexts to a few bits and used their XOR difference to significantly reduce the size of the training samples ($T = \delta_{\text{tr}}$).

Advanced types of feature engineering ($T = \text{A}$) include, e.g., partial decryption of the ciphertexts. For instance, in the case of SPECK32, the right half of the previous round state

can be computed without the key by XORing the two halves and rotating. This type of feature engineering was used in [BGPT21, HLF+24, ZWW24]. A similar technique permits the retrieval of the difference in the previous round for SIMON-like ciphers. [BGL+22] showed that this transformation could significantly improve the accuracies of neural distinguishers, and [LLS+24] exhibited even better distinguishers on SIMON by exploiting inferred information from two rounds ahead; their data format is composed of the two ciphertexts, the difference at the previous round, and the difference two rounds before using subkey 0 for decryption. We refer to such types of feature engineering as A for Advanced. Finally, in [LRCL23], two formats labeled by the authors as MRMSD (Multiple Rounds Multiple Splicing Differences) and MRMSP (Multiple Rounds Multiple Splicing Pairs) use partial decryption with a random key for one round; in the first case, the output difference and this estimated previous round difference are given to the neural distinguisher. In the second case, the corresponding ciphertexts are given. Zhu *et al.* modify this data format by performing only partial encryption to some intermediate state of the round function, i.e., the in- and output of the substitution box nonlinear operation [ZLHH25].

In [YW23], the authors used data format $(R_{r-1}, R'_{r-1}, d_l, C_0, C_1)$ for SPECK, where $d_l$ is an estimation of the difference in the left part at round $r-1$, computed as $((L_r \boxminus R_{r-1}) \oplus (L'_r \boxminus R'_{r-1}))$, equivalent to partial decryption with key 0.

### 6.2.4 Type of distinguishing experiment: $E$

In the initial setting [Goh19b] ($E = \mathrm{R}$), the samples are $E_K(P_0)||E_K(P_0 \oplus x)$, and the label is $x \stackrel{?}{=} \delta$. Gohr additionally defines the *real ciphertext* experiment ($E = \mathrm{R_M}$), where the samples are $E_K(P_0) \oplus x||E_K(P_0 \oplus \delta) \oplus x$, and the label is $x \stackrel{?}{=} 0$, i.e., the distinguisher determines whether the ciphertext pair has been XORed with a random mask. The success of neural distinguishers in this experiment shows that information beyond a simple XOR difference is learned.

In [BBCD22]'s model 1, the samples are formed as $(E_K(P) \oplus E_K(P \oplus \delta_i)), i \in [0; m-1]$, and the label is $i$ ($E = \mathrm{D}$). In [BR21], the samples are built using modular addition difference, rather than XOR, to analyze the ciphers TEA and RAIDEN ($E = \mathrm{R}^+$). In [EGP23], the samples are built through rotational-XOR differences rather than XOR ($E = \mathrm{R}^+$). In [LJSC24a], sample construction employs an XOR difference. For each ciphertext pair $(c, c')$, $N$ distinct output masks $(\Gamma_1, \Gamma'_1), \ldots, (\Gamma_N, \Gamma'_N)$ are applied to generate an $N$-bit input vector for the neural distinguisher, where each bit is computed as $x_i = \Gamma_i \cdot c \oplus \Gamma'_i \cdot c'$ ($E = \mathrm{R}^+$). This methodology represents an implementation of differential-linear cryptanalysis, effectively combining differential properties with linear approximations to enhance the distinguisher's performance.

## 6.3 Comparative Review

Based on the full body of research in Neural Differential Cryptanalysis (Subsection 4.1), this section provides a comparative review of all best published neural distinguishers, classified according to the previously introduced scheme, together with their neural network architecture (Subsection 6.1).

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-CT-R or 2-1-$\delta$-R) or an alternative setting (*n-m-T-E*). If additionally a result in the standard setting is given, we will also present it. *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 a 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*.

To date, neural distinguishers have been applied to analyze 24 symmetric primitives. This comparative review enables new research to be seamlessly integrated into the existing body of work. Comprehensive tables compiling these analyses are provided in the appendices; here, we will focus on SIMON as it is one of the most studied ciphers. The complete list of primitives analyzed to date is as follows: AES (Table 2), ARADI (Table 3), ASCON (Table 4), CHASKEY (Table 5), DES (Table 6), FF (Table 7), GIFT (Table 8), GIMLI (Table 9), GOST (Table 10), HIGHT (Table 11), KATAN (Table 12), KNOT (Table 13), LBCIoT (Table 15), LEA (Table 14), PRESENT (Table 16), PRIDE (Table 17), SHA3 (Table 18), SIMECK (Table 19), SIMON (Table 20), SKINNY (Table 21), SLIM (Table 22), SPECK (Table 23), TEA and XTEA (Table 24), and TinyJAMBU (Table 25).

In addition, Bose *et al.* claimed statistically significant distinguishers for 6-round SPARX and 8-round PICCOLO-80. As theirs is the only work targeting these ciphers and their reported improvements for other ciphers challenge fundamental cryptographic principles, we omitted dedicated tables that would lack contextual comparison. Instead, we included their distinguishers for LEA, PRESENT, and HIGHT with a critical discussion.

### 6.3.1 SIMON

**Table 1:** 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] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 20M | 2M | - | 9 | 0.5907 | [HRC21c] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 20M | 2M | - | 9 | 0.6277 | [SZM21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | / | / | - | 9 | 0.6320 | [TH21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 4-3-CT-R | 40M | 4M | - | 9 | 0.6373 | [SZM21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 4-3-CT-R | 40M | 4M | - | 8 | 0.923 | [WQW+24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-$\delta$-R | 640M | 6.4M | - | 10 | 0.6109 | [HRC21c] |
| | SENet | 2-1-A-R | 4852M | 537M | - | 11 | 0.517 | [BGL+22] |
| | DBitNet | 2-1-CT-R | 2020M | 2M | ✓ | 11 | 0.518 | [BGH+23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-A-R | 640M | 64M | - | 11 | 0.6081 | [LRCL23] |
| | DenseNet | 2-2-CT-D | 2020M | 2M | ✓ | 12 | 0.505 | [WWS24] |
| | SE-ResNet | 16-1-A-R | 160M | 16M | - | 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 | 160M | 16M | - | 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-$\delta$-R | 960M | 9.6M | - | 11 | 0.6143 | [HRC21c] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 12 | 0.515 | [WWS24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 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-$\delta$-R | 1280M | 12.8M | - | 12 | 0.6957 | [HRC21c] |
| | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 13 | 0.518 | [BGH+23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-A-R | 1280M | 128M | - | 13 | 0.701 | [LRCL23] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 14 | 0.506 | [WWS24] |
| | SE-ResNet | 16-1-A-R | 1610M | 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 | 160M | 16M | - | 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 | [BGH+23] |
| 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 6.2. 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.
RK Related key setting.

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 20 provides an overview of the differential neural distinguishers developed for the SIMON family of block ciphers. The most extensively studied variant is SIMON-32, with various neural network architectures and settings explored across multiple works. In the standard setting, the best distinguisher achieves round 11 through an automated pipeline [BGH+23]. By using multiple ciphertext pairs ($n = 16, 32, 64$) and employing advanced feature engineering techniques, as in [LRCL23, LLS+24, ZWW24], the distinguisher performance surpasses this result, extending the analysis to round 12 [LLS+24].

For the case of SIMON, some authors experimented with a vast amount of data: [HRC21c] used $k \cdot 10^7$ for $k = 32, 48, 64$ (maximum of 640M) pairs for training, and [BGL+22] obtained an 11-round distinguisher for SIMON32 at the cost of staged trained in four steps, with respectively $10^7$, $2^{28}$, $2 \cdot 2^{30}$ (2426M pairs). In [BGH+23], the authors proposed a polishing step, retraining a neural distinguisher initially trained with $10^7$ pairs with an additional $10^9$ pairs: $10^7$, $310^9$ (1010M pairs). This polishing step was also used by Wang et al. [WWS24]. Similarly, Zhang et al. [ZWW24] used a staged training approach: $4 \cdot 10^7$ samples, each sample with 16 pairs (640M pairs).

In [LLS+24], Lu *et al.* used advanced feature engineering and 80M ciphertext pairs ($10^7$ samples, each composed of 8 pairs) and reached 12 rounds of SIMON32 in the single-key scenario. In the related key scenario, the same authors obtained a 13-round distinguisher, whereas [EGP23] only reached 11 rounds with a rotational XOR distinguisher. The feature engineering proposed in [LLS+24] was also used in [WW24a]. Further, the authors used staged training for a subset of the obtained distinguishers: $3 \cdot 2^{25}$ samples, 8 pairs each (805M pairs).

## 6.4 Discussion

Based on our comprehensive assessment of research in neural differential cryptanalysis (Subsection 4.1), we identify several promising directions and critical challenges that merit further investigation. Our analysis focuses primarily on thoroughly vetted cryptographic primitives – those subjected to substantial cryptanalytic scrutiny (demonstrated by five or more papers that have not been challenged), specifically SIMON, SIMECK, SPECK).

**Network Architectures (N)** Findings on optimal neural network architectures for cryptanalysis remain contradictory. While Baksi *et al.* [BBCD22] concluded CNNs were unsuitable for distinguishers and found MLPs superior on GIMLI-PERMUTATION, Bellini *et al.* [BR21] and Wang *et al.* [WWH21] demonstrated effective CNN-based distinguishers for PRESENT and SPECK. Mishra *et al.* [MPKM+22] reported MLPs outperforming CNNs on GIFT and PRIDE, whereas Sun *et al.* [SSL+22] found LSTMs superior to MLPs on TinyJAMBU and GIFT. Tcydenova *et al.* [TSL23] evaluated various architectures but found no significant improvements over ResNet, though noted overfitting issues. Lv *et al.* [LJSC24a] comprehensively compared multiple techniques for differential-linear cryptanalysis, with MLPs consistently outperforming alternatives, including ELLR, Logistic Regression, and LightGBM.

Convolutional neural networks, pioneered as the original distinguisher architecture in [Goh19a], consistently demonstrate excellent performance in neural cryptanalysis, with convolutional architectures ranking among the most effective distinguishers across virtually all cryptographic primitives with a substantial body of neural cryptanalytic research. This is expected as DCNNs have demonstrated remarkable feature extraction capabilities across various disciplines, particularly in image recognition. Once the relevant features have been identified, classical or simpler neural models can often achieve performance comparable to their complex neural counterparts [BGPT21, BLYZ23]. However, meaningful comparison

between concrete approaches remains challenging due to numerous influential factors beyond architecture alone, including the number of ciphertext samples and input differences, the sophistication of feature engineering techniques, and the experimental design variations. This complexity underscores the critical need for comprehensive benchmarking studies to establish definitive conclusions about optimal approaches, a challenge we address in detail in Subsection 8.1.

Unlike natural language processing, neural cryptanalysis has not consistently benefited from the "bigger is better" scaling paradigm described by Kaplan *et al.* [KMH+20]. Research has not conclusively demonstrated that deeper or wider neural architectures reliably improve distinguishing capability in cryptographic contexts. Notably, Gohr [Goh19a] employed shallower architectures for distinguishers targeting near-uniform ciphertext distributions (specifically for 7 and 8 rounds of SPECK encryption). Differential ciphertext distributions contain subtle non-uniform statistical properties that remain challenging to capture. This underscores a fundamental challenge in developing neural networks capable of effectively learning these cryptographic statistics – a problem requiring sophisticated modeling approaches, which we examine thoroughly in Subsection 8.2.

**Multi-Pair Distinguishers ($n > 2$)** Neural distinguishers that process multiple ciphertext pairs simultaneously have historically shown minimal practical advantages over simpler approaches [GLN22, CSYY23]. While these complex multi-pair architectures typically performed equivalently or worse than single-pair distinguishers with basic score aggregation, recent evidence suggests this paradigm is shifting – particularly for lightweight block ciphers. Our comprehensive analysis reveals that for SPECK, SIMON, and SIMECK, multi-pair distinguishers have successfully broken more rounds than their single-pair counterparts. This development closely matches common notions in differential cryptanalysis: as the number of rounds increases, the differential probability decreases, and more data is needed to observe a bias; grouping multiple pairs into a sample artificially increases the chance that a rare but relevant differential propagation will occur within each sample.

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, introduced in [ATH02], where the 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], seem to be promising directions to explore in order to improve multi-pair classifiers. Similarly, the problem of *anomaly detection* has received considerable attention in the machine learning community; if we choose to treat the 'elephant pair' as an anomaly to an otherwise unremarkable distribution, adapting approaches such as Deep One-Class Classification [RVG+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 significant body of work available in the deep learning community, and we believe there is significant room for improvement through incorporating these techniques.

**Using Multiple Input Differences ($m > 1$)** The effectiveness of differential cryptanalysis using multiple input differences has been demonstrated across several cipher families,

including SIMON, SIMECK, and SPECK. Gohr *et al.* [GLN22] established a crucial relationship: neural differential distinguisher accuracy correlates with the statistical distance between the separated ciphertext-difference distributions (particularly when distinguishing from a uniform distribution $E = R$). Distinguishers naturally perform better when targeting input differences that produce more distinguishable output distributions.

Despite this advantage, integrating multiple-difference approaches into practical key recovery attacks presents significant challenges. The current attack framework pioneered by Gohr [Goh19a] fundamentally relies on distinguishing real ciphertext distributions from uniform distributions as its core mechanism. Adapting this framework to leverage the statistical power of multiple input differences would require substantial modifications to the underlying cryptanalytic methodology.

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

**Feature Engineering ($T$)** Feature engineering has demonstrated a significant impact on distinguisher performance [LLS+24], with notable examples including partial decryption and combining ciphertext values with difference-related features. Interestingly, virtually all multi-pair distinguishers obtaining state-of-the-art results utilize advanced feature engineering. Nevertheless, well-designed network architectures can autonomously learn optimal feature representations, as demonstrated by Gohr *et al.* [GLN22] for SIMON, which achieved comparable results to the feature engineering of Bao *et al.* [BGL+22].

The extent to which hand-crafted features enhance neural network learning capabilities remains an open research question. Establishing comprehensive benchmarking frameworks would provide valuable insights into the relative merits of automated versus engineered feature extraction for cryptanalytic applications (Subsection 8.1).

**Alternative Adversarial Models (E)** Paralleling classical cryptanalysis, adversarial models with expanded capabilities consistently outperform against increased cipher rounds, as demonstrated by related-key and conditional approaches extending several rounds beyond chosen plaintext counterparts. Rotational cryptanalysis and other specialized techniques have also shown promising results when adapted to neural frameworks, exploiting structural weaknesses conventional differential approaches might miss.

The critical research question is what additional adversarial models remain unexplored. Classical cryptanalysis offers numerous attack vectors yet to be fully adapted to neural network distinguishers. Systematically mapping these classical techniques to their neural counterparts could reveal new attack classes.

# 7 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 interpret the success of neural network training correctly, it is important to distinguish between training, validation, and test data carefully. Each dataset has an important role: The *training data* is used to calculate the loss of the model and to update the model 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 the training, the model does not learn new generalizable features of the training data but instead uses its parameters to learn the training dataset "by heart". This leads to an increasing validation loss. Instead of using the model that has been trained for the maximum number of epochs, in this case, 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* relate to its computational training cost straightforwardly. 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.[4] 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.

**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.

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.

# 8 Future Challenges

## 8.1 The 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 gain a better understanding of neural distinguisher, 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 on sets of standard problems and compare them in a leaderboard. This objective is, however, not straightforward, and we discuss some friction points below.

**Defining Problems** A problem can be defined as an $n$-$M$-$T$-$E$ configuration, primitive, training pipeline, and dataset size. A logical first step would be evaluating all models on the

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

initial SPECK32 problem in the 2-1-CT-R setting to identify top-performing architectures.

Training regimes are critical: Gohr's work [Goh19a] 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 [Goh19b], layer freezing [GLN22], previous-round distinguisher retraining [BR21], or increased final-round samples). A standardized pipeline for comparing enhanced distinguishers would be beneficial.

Sample quantity also matters. Many works follow Gohr's approach [Goh19a] ($10^7$ training, $10^6$ test samples), as reduction significantly impacts performance. Multiple-pair sample approaches [BBCD20] present comparison challenges: fixing sample count gives unfair advantages to models seeing more pairs, while fixing pair count may disadvantage models trained on fewer samples (extreme case: $10^7$ pairs per sample would mean training on a single sample). Despite some works using over 1 billion ciphertexts, little research explores this data magnitude in the 2-1-CT-R scenario versus multiple-pair approaches – an axis worth including in benchmarking studies.

**Metrics** The first challenge to comparing different models is to define what is to be compared. As of now, the main metrics used to compare neural distinguishers are accuracy, true positive rate, true negative rate, and more recently [BGH+23], 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 a neural network based on inference speed or parameter count constraints, is often used as a baseline comparison for new models. For neural distinguishers, we could similarly use the number of parameters and FLOPs ratio with the original architecture from Gohr, providing context to 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 Appendix B), prepended differentials, and neutral bits.

## 8.2 The AI-$\mathcal{ND}$ Challenge

The neural network architectures currently employed in Neural Differential Cryptanalysis have origins that trace back several years. For instance, the Inception Module by Google researchers was introduced in a seminal paper in 2014 [SLJ+15]. 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 led to a larger parameter count. Figure 3 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) like GPT, which contain billions of parameters. The deep learning models used to date in Neural Differential Cryptanalysis have low parameter counts compared to more modern "Deep Learning Era" models. Challenges when increasing the parameter count of the models 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 for enhancing 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

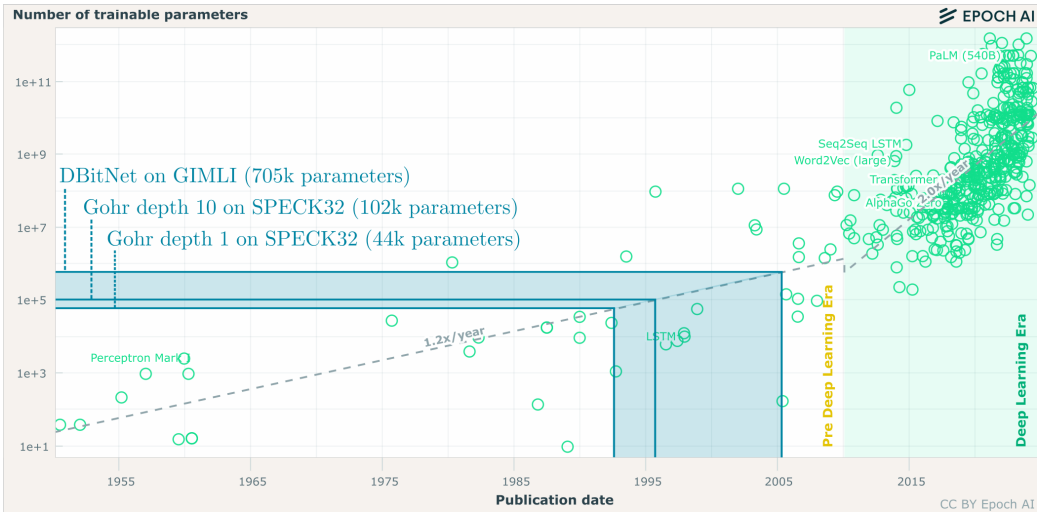


**Figure 3:** Adapted from [Epo24] with added data for Gohr's $\mathcal{ND}_{\text{Gohr}}$ on Table 23, and DBitNet on Table 9 from [BGH+23, Table 5].

through iterative feedback and learning from interactions. These advanced AI methodologies, when applied to cryptographic neural differential distinguishers, can lead to more accurate models. The increased computational power available today allows for training deeper and more complex networks, which 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 8.1, 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[5] to allow AI researchers and engineers to apply state-of-the-art methods to Neural Differential Cryptanalysis.

# 9 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 two challenges, namely comparing neural distinguisher results and scaling up to much larger and more ambitious architectures.

Over the past 6 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 types of feature engineering, particularly through 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, a large portion of the literature has focused on differential-based property

[5]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].

for one pair and one input difference, but many variations could be built, as well as tasks related to different types of cryptanalysis or entirely new distinguishing experiments.

## References


[AABAA22] Abdullah F Al-Aboosi, Matan Broner, and Fadhil Y Al-Aboosi. Bingo: A semi-centralized password storage system. *Journal of Cybersecurity and Privacy*, 2:444–465, 2022.

[AAEK22] Hicham Tahiri Alaoui, Ahmed Azouaoui, and Jamal El Kafi. Artificial neural networks cryptanalysis of merkle-hellman knapsack cryptosystem. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 196–205. Springer, 2022.

[AEA17] Wasan Shaker Awad and El-Sayed M. El-Alfy. *Computational Intelligence in Cryptology*, page 1636–1652. IGI Global, 2017.

[AKVS+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. *Cryptography*, 8(4):55, 2024.

[ALH22] Jianqiao An, Yuandu Lai, and Yahong Han. Logic rule guided attribution with dynamic ablation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 77–85, 2022.

[ATH02] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, 15:561–568, 01 2002.

[Bak21] Anubhab Baksi. Classical and physical security of symmetric key cryptographic algorithms. In *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 1–2. IEEE, 2021.

[BBC+23] Anubhab Baksi, Jakub Breier, Anupam Chattopadhyay, Tomáš Gerlich, Sylvain Guilley, Naina Gupta, Takanori Isobe, Arpan Jati, Petr Jedlicka, Hyunjun Kim, et al. Baksheesh: Similar yet different from gift. *Cryptology ePrint Archive*, 2023.

[BBCD20] Anubhab Baksi, Jakub Breier, Yi Chen, and Xiaoyang Dong. Machine learning assisted differential distinguishers for lightweight ciphers (extended version). *Cryptology ePrint Archive*, 2020.

[BBCD22] Anubhab Baksi, Jakub Breier, Yi Chen, and Xiaoyang Dong. Machine learning-assisted differential distinguishers for lightweight ciphers. *Classical and Physical Security of Symmetric Key Cryptographic Algorithms*, pages 141–162, 2022.

[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*, 2023.

[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.

[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.

[BCdST+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 *International Conference on Applied Cryptography and Network Security*, pages 149–177. Springer, 2023.

[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.

[BGG+23] Emanuele Bellini, David Gerault, Juan Grados, Yun Ju Huang, Rusydi Makarim, Mohamed Rachidi, and Sharwan Tiwari. Claasp: a cryptographic library for the automated analysis of symmetric primitives. In *International Conference on Selected Areas in Cryptography*, pages 387–408. Springer, 2023.

[BGH+23] E Bellini, D Gerault, A Hambitzer, M Rossi, et al. A cipher-agnostic neural training pipeline with automated finding of good input differences. *IACR TRANSACTION ON SYMMETRIC CRYPTOLOGY*, 2023(3):184–212, 2023.

[BGL+21] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Conditional differential-neural cryptanalysis. *IACR Cryptol. ePrint Arch.*, 2021:719, 2021.

[BGL+22] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Enhancing differential-neural cryptanalysis. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 318–347. Springer, 2022.

[BGPT21] Adrien Benamira, David Gerault, Thomas Peyrin, and Quan Quan Tan. A deeper look at machine learning-based cryptanalysis. In *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 40*, pages 805–835. Springer, 2021.

[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.

[BLYZ23] Zhenzhen Bao, Jinyu Lu, Yiran Yao, and Liu Zhang. More insight on deep learning-aided cryptanalysis. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 436–467. Springer, 2023.

[Boa24a] Sara Boanca. 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.

[Boa24b] Sara Boancă. Exploring patterns and assessing the security of pseudorandom 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.

[BPC24] 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.

[BPRC24] Amrita Bose, Debranjan Pal, and Dipanwita Roy Chowdhury. Cryptographic distinguishers through deep learning for lightweight block ciphers. In *International Conference on Applications and Techniques in Information Security*, pages 47–63. Springer, 2024.

[BPS22] Carlo Brunetta and Pablo Picazo-Sanchez. Modelling cryptographic distinguishers using machine learning. *Journal of Cryptographic Engineering*, 12:123–135, 2022.

[BR21] Emanuele Bellini and Matteo Rossi. Performance comparison between deep learning-based and conventional cryptographic distinguishers. In *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3*, pages 681–701. Springer, 2021.

[Bru21] Carlo Brunetta. *Cryptographic Tools for Privacy Preservation*. PhD thesis, Department of Computer Science & Engineering, Chalmers University of Technology, Gothenburg, Sweden, 2021.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *J. Cryptology*, 4:3–72, 1991.

[BSL21] KVL Bhargavi, Chungath Srinivasan, and KV Lakshmy. Panther: a sponge based lightweight authenticated encryption scheme. In *Progress in Cryptology–INDOCRYPT 2021: 22nd International Conference on Cryptology in India, Jaipur, India, December 12–15, 2021, Proceedings 22*, pages 49–70. Springer, 2021.

[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.

[CDJ+21] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas-López, and Mridul Nandi. thyena: Making hyena even smaller. 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 26–48. Springer, 2021.

[CdOAB+18a] 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.

[CdOAB+18b] 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.

[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.

[CLC12a] 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.

[CLC12b] 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 *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence*, AISec '12, page 105–110, New York, NY, USA, 2012. Association for Computing Machinery.

[CS21] Bang Yuan Chong and Iftekhar Salam. Investigating deep learning approaches on the security analysis of cryptographic algorithms. *Cryptography*, 5:30, 2021.

[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.

[CSY23] Yi Chen, Yantian Shen, and Hongbo Yu. Neural-aided statistical attack for cryptanalysis. *The Computer Journal*, 66:2480–2498, 2023.

[CSYY23] Yi Chen, Yantian Shen, Hongbo Yu, and Sitong Yuan. A new neural distinguisher considering features derived from multiple ciphertext pairs. *The Computer Journal*, 66:1419–1433, 2023.

[DCC23] Haoran Deng, Xianghui Cao, and Yu Cheng. Attention in differential cryptanalysis on lightweight block cipher speck. In *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, pages 1–9. IEEE, 2023.

[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.

[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 *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 98–127. Springer, 2023.

[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.

[DM23] Shivank Kumar Dadhwal and Girish Mishra. Machine learning-based classification between block cipher and stream cipher. In *Inventive Computation and Information Technologies: Proceedings of ICICIT 2022*, pages 531–542. Springer, 2023.

[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.

[DZF+21] Ming Duan, Rui Zhou, Chaohui Fu, Sheng Guo, and Qianqiong Wu. Vulnerability testing on the key scheduling algorithm of present using deep learning. In *International Conference on Security and Privacy in New Computing Environments*, pages 307–318. Springer, 2021.

[EAZD23] Zahra Ebadi Ansaroudi, Rocco Zaccagnino, and Paolo Dâ€™Arco. On pseudorandomness and deep learning: A case study. *Applied Sciences*, 13:3372, 2023.

[EGP23] Amirhossein Ebrahimi, David Gerault, and Paolo Palmieri. Deep learning-based rotational-xor distinguishers for and-rx block ciphers: Evaluations on simeck and simon. In *International Conference on Selected Areas in Cryptography*, pages 429–450. Springer, 2023.

[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 *Progress in Cryptology–INDOCRYPT 2020: 21st International Conference on Cryptology in India, Bangalore, India, December 13–16, 2020, Proceedings 21*, pages 345–369. Springer, 2020.

[ElS21] Muhammad ElSheikh. *MILP-aided Cryptanalysis of Some Block Ciphers*. PhD thesis, Concordia University, 2021.

[Epo24] Epoch AI. Parameter, compute and data trends in machine learning, 2024. Accessed: 2024-05-31.

[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 *International Symposium on Foundations and Practice of Security*, pages 123–141. Springer, 2022.

[Ess23] Bernhard Esslinger. *Learning and Experiencing Cryptography with CrypTool and SageMath*. Artech House, 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), 2024.

[FLW+23] Zhuohui Feng, Ye Luo, Chao Wang, Qianqian Yang, Zhiquan Liu, and Ling Song. Improved differential cryptanalysis on speck using plaintext structures. In *Australasian Conference on Information Security and Privacy*, pages 3–24. Springer, 2023.

[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.

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

[GBC17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning: The MIT Press*, volume 19. The MIT Press, 2017.

[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.

[GJS20] 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 february 12, 2020. *challenge*, 2:2, 2020.

[GJS21] Aron Gohr, Sven Jacob, and Werner Schindler. Subsampling and knowledge distillation on adversarial examples: New techniques for deep learning based side channel evaluations. In *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27*, pages 567–592. Springer, 2021.

[GLN22] Aron Gohr, Gregor Leander, and Patrick Neumann. An assessment of differential-neural distinguishers. *Cryptology ePrint Archive*, 2022.

[GLP+24] Yue-Tian Goi, Shu-Min Leong, Raphaël C-W Phan, Shangqi Lai, and Ana Sălăgean. Unveiling the black box: Neural cryptanalysis with xai. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1951–1956. IEEE, 2024.

[Goh19a] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 150–179. Springer, 2019.

[Goh19b] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 150–179, Cham, 2019. Springer International Publishing.

[Goh22] Aron Gohr. Brute force cryptanalysis. *Cryptology ePrint Archive*, 2022.

[GPT21] David Gerault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ascon. *IACR Transactions on Symmetric Cryptology*, 2021.

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

[GVWT21] David Gunning, Eric Vorm, Yunyan Wang, and Matt Turek. Darpa's explainable ai (xai) program: A retrospective. *Authorea Preprints*, 2021.

[GZDAL22] Hicham Grari, Khalid Zine-Dine, Ahmed Azouaoui, and Siham Lamzabi. Deep learning-based cryptanalysis of a simplified aes cipher. *International Journal of Information Security and Privacy (IJISP)*, 16:1–16, 2022.

[Hal22] Roger A Hallman. Poster evegan: Using generative deep learning for cryptanalysis. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3355–3357, 2022.

[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. *Cognitive Computation*, 16(1):45–74, 2024.

[HGH+23] Anna Hambitzer, David Gerault, Yun Ju Huang, Najwa Aaraj, and Emanuele Bellini. Nnbits: Bit profiling with a deep learning ensemble based distinguisher. In *Cryptographersâ€™ Track at the RSA Conference*, pages 493–523. Springer, 2023.

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

[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.

[HLG+23] Ying Huang, Lang Li, Ying Guo, Yu Ou, and Xiantong Huang. An efficient differential analysis method based on deep learning. *Computer Networks*, 224:109622, 2023.

[HLK+14] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. Lea: A 128-bit block cipher for fast encryption on common processors. In Yongdae Kim, Heejo Lee, and Adrian Perrig, editors, *Information Security Applications*, pages 3–27, Cham, 2014. Springer International Publishing.

[HLLL] Ying Huang, Lang Li, Di Li, and Yongchao Li. Iabc: A neural integral distinguisher for and-rx ciphers. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–15.

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

[HLZW20] Botao Hou, Yongqiang Li, Haoyue Zhao, and Bin Wu. Linear attack on round-reduced des using deep learning. In *Computer Security–ESORICS*

*2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II 25*, pages 131–145. Springer, 2020.

[HRC21a] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Cryptanalysis of round-reduced simon32 based on deep learning. *Cryptology ePrint Archive*, 2021.

[HRC21b] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Improve neural distinguisher for cryptanalysis. *Cryptology ePrint Archive*, 2021.

[HRC21c] ZeZhou Hou, JiongJiong Ren, and ShaoZhen Chen. Improve neural distinguishers of simon and speck. *Security and Communication Networks*, 2021:1–11, 2021.

[HRC21d] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Sat-based method to improve neural distinguisher and applications to simon. *Cryptology ePrint Archive*, 2021.

[HRC23] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. Practical attacks of round-reduced simon based on deep learning. *The Computer Journal*, 66:2517–2534, 2023.

[HSB+22] Isha Hameed, Samuel Sharpe, Daniel Barcklow, Justin Au-Yeung, Sahil Verma, Jocelyn Huang, Brian Barr, and C. Bayan Bruss. Based-xai: Breaking ablation studies down for explainable artificial intelligence, 2022.

[HSS18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[ITW18] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2127–2136. PMLR, 10–15 Jul 2018.

[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.

[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.

[JKM20] Aayush Jain, Varun Kohli, and Girish Mishra. Deep learning based differential distinguisher for lightweight cipher present. *Cryptology ePrint Archive*, 2020.

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

[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.

[JMTD22] Dushica Jankovikj, Hristina Mihajloska Trpceska, and Vesna Dimitrova. Cryptanalysis of round-reduced ascon powered by ml. In *The 19th International Conference on Informatics and Information Technologies – CIIT*, 2022.

[Jun05] Pascal Junod. Statistical cryptanalysis of block ciphers. Technical report, EPFL, 2005.

[Kar23] SK Karthika. Check for theoretical and deep learning based analysis of biases in salsa 128 bits sk karthika) and kunwar singh department of of computer science and engineering, national institute of. In *Mobile Internet Security: 6th International Symposium, MobiSec 2022, Jeju, South Korea, December 15–17, 2022, Revised Selected Papers*, page 147. Springer Nature, 2023.

[KATT20] Shivam Kalra, Mohammed Adnan, Graham Taylor, and H. 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*, pages 677–693, Cham, 2020. Springer International Publishing.

[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.

[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 *International Conference on Applied Cryptography and Network Security*, pages 248–276. Springer, 2022.

[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. *Journal of Information Processing*, 31:550–561, 2023.

[KJL+22] Hyunji Kim, Kyungbae Jang, Sejin Lim, Yeajun Kang, Wonwoong Kim, and Hwajeong Seo. Quantum neural network based distinguisher for differential cryptanalysis on simplified block ciphers. *Cryptology ePrint Archive*, 2022.

[KJL+23] Hyunji Kim, Kyungbae Jang, Sejin Lim, Yeajun Kang, Wonwoong Kim, and Hwajeong Seo. Quantum neural network based distinguisher on speck-32/64. *Sensors*, 23:5683, 2023.

[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):1196, 2024.

[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*, 32:225–236, 2023.

[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*, 25:986, 2023.

[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. *arXiv preprint arXiv:2001.08361*, 2020.

[KMS02] Alexander Klimov, Anton Mityagin, and Adi Shamir. *Analysis of Neural Cryptography*, page 288–298. Springer Berlin Heidelberg, 2002.

[KS22] SK Karthika and Kunwar Singh. Theoretical and deep learning based analysis of biases in salsa 128 bits. In *International Symposium on Mobile Internet Security*, pages 147–164. Springer, 2022.

[KSJS21] Hyun-Ji Kim, Gyeong-Ju Song, Kyung-Bae Jang, and Hwa-Jeong Seo. Cryptanalysis of caesar using quantum support vector machine. In *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–5. IEEE, 2021.

[KVD+25] Benjamin D Kim, Vipindev Adat Vasudevan, Rafael GL D'Oliveira, Alejandro Cohen, Thomas Stahlbuhk, and Muriel Médard. Cryptanalysis via machine learning based information theoretic metrics. *arXiv preprint arXiv:2501.15076*, 2025.

[KY21] Manoj Kumar and Tarun Yadav. Milp based differential attack on round reduced warp. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 42–59. Springer, 2021.

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

[LCLH22] Dongdong Lin, Shaozhen Chen, Manman Li, and Zezhou Hou. The construction and application of (related-key) conditional differential neural distinguishers on katan. In *International Conference on Cryptology and Network Security*, pages 203–224. Springer, 2022.

[LF24] Ronniel D Labio and Enrique Festijo. Neural network–based cryptanalysis of present and d-present block ciphers. In *Cryptology and Information Security Conference 2024*, page 110, 2024.

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

[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.

[LLHC23] Dongdong Lin, Manman Li, Zezhou Hou, and Shaozhen Chen. Conditional differential analysis on the katan ciphers based on deep learning. *IET Information Security*, 17:347–359, 2023.

[LLL+21] Guozhen Liu, Jingwen Lu, Huina Li, Peng Tang, and Weidong Qiu. Preimage attacks against lightweight scheme xoodyak based on deep learning. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pages 637–648. Springer, 2021.

[LLL+22] Jinyu Lu, Guoqiang Liu, Yunwen Liu, Bing Sun, Chao Li, and Li Liu. Improved neural distinguishers with (related-key) differentials: Applications in SIMON and SIMECK. *CoRR*, abs/2201.03767, 2022.

[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. *The Computer Journal*, 67(2):537–547, 2024.

[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 *Data Mining: 19th Australasian Conference on Data Mining, AusDM 2021, Brisbane, QLD, Australia, December 14-15, 2021, Proceedings 19*, pages 152–164. Springer, 2021.

[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.

[LRC22] Jiashuo Liu, Jiongjiong Ren, and Shaozhen Chen. Effective network parameter reduction schemes for neural distinguisher. *Cryptology ePrint Archive*, 2022.

[LRC23] JiaShuo Liu, JiongJiong Ren, and ShaoZhen Chen. A deep learning aided differential distinguisher improvement framework with more lightweight and universality. *Cybersecurity*, 6:47, 2023.

[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.

[LRCL23] JiaShuo Liu, JiongJiong Ren, ShaoZhen Chen, and ManMan Li. Improved neural distinguishers with multi-round and multi-splicing construction. *Journal of Information Security and Applications*, 74:103461, 2023.

[LSW+23] Cathy Li, Jana Sotakova, Emily Wenger, Zeyuan Allen-Zhu, Francois Charton, and Kristin Lauter. Salsa verde: a machine learning attack on learning with errors with sparse small secrets. *arXiv preprint arXiv:2306.11641*, 2023.

[LTJ+20] Ting Rong Lee, Je Sen Teh, Norziana Jamil, Jasy Liew Suet Yan, and Jiageng Chen. Assessing lightweight block cipher security using linear and nonlinear machine learning classifiers. *Cryptology ePrint Archive*, 2020.

[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.

[LTZ22a] Lijun Lyu, Yi Tu, and Yingjie Zhang. Deep learning assisted key recovery attack for round-reduced simeck32/64. In *International Conference on Information Security*, pages 443–463. Springer, 2022.

[LTZ22b] Lijun Lyu, Yi Tu, and Yingjie Zhang. Improving the deep-learning-based differential distinguisher and applications to simeck. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 465–470. IEEE, 2022.

[LWAZ+24] Cathy Li, Emily Wenger, Zeyuan Allen-Zhu, Francois Charton, and Kristin E Lauter. Salsa verde: a machine learning attack on lwe with sparse small secrets. *Advances in Neural Information Processing Systems*, 36, 2024.

[MGKMP21] Girish Mishra, Indivar Gupta, SVSSNVG Krishna Murthy, and SK Pal. Deep learning based cryptanalysis of stream ciphers. *Defence Science Journal*, 71, 2021.

[MJBHC22] Luca Mariot, Domagoj Jakobovic, Thomas Bäck, and Julio Hernandez-Castro. Artificial intelligence for the design of symmetric cryptographic primitives. In *Security and Artificial Intelligence: A Crossdisciplinary Approach*, pages 3–24. Springer, 2022.

[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*, pages 1–21, 2021.

[MLR+23] Isabella Martínez, Valentina López, Daniel Rambaut, Germán Obando, Valérie Gauthier-Umaña, and Juan F Pérez. Recent advances in machine learning for differential cryptanalysis. In *Colombian Conference on Computing*, pages 45–56. Springer, 2023.

[MLYW22] Pingchuan Ma, Zhibo Liu, Yuanyuan Yuan, and Shuai Wang. Neurald: Detecting indistinguishability violations of oblivious ram with neural distinguishers. *IEEE Transactions on Information Forensics and Security*, 17:982–997, 2022.

[MP88] Marvin L Minsky and Seymour A Papert. Perceptrons: expanded edition, 1988.

[MPKM+22] Girish Mishra, SK Pal, SVSSNVG Krishna Murthy, Ishan Prakash, and Anshul Kumar. Deep learning-based differential distinguisher for lightweight ciphers gift-64 and pride. In *Machine Intelligence and Smart Systems: Proceedings of MISS 2021*, pages 245–257. Springer, 2022.

[MPM+21] Girish Mishra, SK Pal, SVSSNVG Krishna Murthy, Kanishk Vats, and Rakshak Raina. Distinguishing lightweight block ciphers in encrypted images. *Defence Science Journal*, 71:647–655, 2021.

[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.

[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.

[NR23] Abderrahmane Nitaj and Tajjeeddine Rachidi. Applications of neural network-based ai in cryptography. *Cryptography*, 7:39, 2023.

[PCDC24] Debranjan Pal, Mainak Chaudhury, Abhijit Das, and Dipanwita Roy Chowdhury. Deep learning-based differential distinguishers for nist standard authenticated encryption and permutations. In *International Conference on Mathematics and Computing*, pages 1–13. Springer, 2024.

[PJ21] Stjepan Picek and Domagoj Jakobovic. Evolutionary computation and machine learning in cryptology. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1089–1118, 2021.

[PJ22] Stjepan Picek and Domagoj Jakobovic. Evolutionary computation and machine learning in security. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1572–1601, 2022.

[PKM23] Seonghwan Park, Hyunil Kim, and Inkyu Moon. Automated classical cipher emulation attacks via unified unsupervised generative adversarial networks. *Cryptography*, 7:35, 2023.

[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.

[PMC+22] Debranjan Pal, Upasana Mandal, Mainak Chaudhury, Abhijit Das, and Dipanwita Roy Chowdhury. A deep neural differential distinguisher for arx based block cipher. *Cryptology ePrint Archive*, 2022.

[PMDC22] Debranjan Pal, Upasana Mandal, Abhijit Das, and Dipanwita Roy Chowdhury. Deep learning based differential classifier of pride and rc5. In *International Conference on Applications and Techniques in Information Security*, pages 46–58. Springer, 2022.

[PMK20] Manan Pareek, Girish Mishra, and Varun Kohli. Deep learning based analysis of key scheduling algorithm of present cipher. *Cryptology ePrint Archive*, 2020.

[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.

[PPWR23] Raphaël C-W Phan, Arghya Pal, KokSheik Wong, and Sailaja Rajanala. C$\eta\iota$DAE: Cryptographically distinguishing autoencoder for cipher cryptanalysis. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pages 4467–4472. IEEE, 2023.

[PSM23] Pooja, Shantanu, and Girish Mishra. Related-key neural distinguisher for round-reduced present cipher. In *International Conference on Advances in Data-driven Computing and Intelligent Systems*, pages 393–405. Springer, 2023.

[PTD22a] Milena Gjorgjievska Perusheska, Hristina Mihajloska Trpceska, and Vesna Dimitrova. Deep learning-based cryptanalysis of different aes modes of operation. In *Future of Information and Communication Conference*, pages 675–693. Springer, 2022.

[PTD22b] Milena Gjorgjievska Perusheska, Hristina Mihajloska Trpceska, and Vesna Dimitrova. Deep learning-based cryptanalysis of different aes modes of operation. In *Future of Information and Communication Conference*, pages 675–693. Springer, 2022.

[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.

[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.

[RD20] Mark Randolph and William Diehl. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptogr.*, 4(2):15, 2020.

[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.

[Riv91] Ronald L. Rivest. Cryptography and machine learning. In *Proceedings of the International Conference on the Theory and Applications of Cryptology: Advances in Cryptology*, ASIACRYPT '91, page 427–439, Berlin, Heidelberg, 1991. Springer-Verlag.

[RLS23] Vignesh Rajakumar, KV Lakshmy, and Chungath Srinivasan. Deep learning based cryptanalysis on slim cipher. In *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pages 1–6. IEEE, 2023.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Ros24] Matteo Rossi. *Automatic differential cryptanalysis of symmetric ciphers*. PhD thesis, Polytechnic University of Turin, Italy, 2024.

[RRSM22a] Reshma Rajan, Rupam Kumar Roy, Diptakshi Sen, and Girish Mishra. Deep learning-based differential distinguisher for lightweight cipher gift-cofb. In *Machine Intelligence and Smart Systems: Proceedings of MISS 2021*, pages 397–406. Springer, 2022.

[RRSM22b] Reshma Rajan, Rupam Kumar Roy, Diptakshi Sen, and Girish Mishra. Gift-cofb. *Machine Intelligence and Smart Systems: Proceedings of MISS 2021*, page 397, 2022.

[RVG+18] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018.

[SAH+20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[SBG+24] Arpita Sarkar, Malay Bhattacharyya, Utpal Garain, Saibal Kumar Pal, Shantanu Shantanu, Sanghamitra Bandyopadhyay, and Nikhil R Pal. Leveraging synergy to design neural differential distinguishers for lightweight block ciphers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

[SCL24] Byoungjin Seok, Donghoon Chang, and Changhoon Lee. A novel approach to construct a good dataset for differential-neural cryptanalysis. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[Seo24] Byoungjin Seok. Truncated differential-neural key recovery attacks on round-reduced hight. *Electronics*, 13(20):4053, 2024.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. J.*, 28(4):656–715, 1949.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[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.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[SLL24] Jiali Shi, Chao Li, and Guoqiang Liu. Differential attack with constants on $\mu 2$ block cipher. *The Computer Journal*, 67:195–209, 2024.

[SM23a] Ayan Sajwan and Girish Mishra. Comparative analysis of resnet and densenet for differential cryptanalysis of speck 32/64 lightweight block cipher. *Cryptology ePrint Archive*, 2023.

[SM23b] Ayan Sajwan and Girish Mishra. Comparative analysis of resnet and densenet for differential cryptanalysis of speck 32/64 lightweight block cipher. In *International Conference on Cryptology & Network Security with Machine Learning*, pages 495–504. Springer, 2023.

[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*, volume 14925 of *Lecture Notes in Computer Science*, pages 37–71. Springer, 2024.

[So20] Jaewoo So. Deep learning-based cryptanalysis of lightweight block ciphers. *Security and Communication Networks*, 2020:1–11, 2020.

[Som23] Åvald Åslaugson Sommervoll. *Machine learning for offensive cyber operations*. PhD thesis, Institute for Informatics, University of Oslo, 2023.

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

[SSL+22] Tao Sun, Dongsu Shen, Saiqin Long, Qingyong Deng, and Shiguo Wang. Neural distinguishers on tinyjambu-128 and gift-64. In *International Conference on Neural Information Processing*, pages 419–431. Springer, 2022.

[SSL+24] Dongsu Shen, Yijian Song, Yuan Lu, Saiqin Long, and Shujuan Tian. Neural differential distinguishers for gift-128 and ascon. *Journal of Information Security and Applications*, 82:103758, 2024.

[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.

[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.

[Sug24a] 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.

[Sug24b] Nobuyuki Sugio. Implementation of cryptanalytic programs using chatgpt. *Cryptology ePrint Archive*, 2024.

[SWL+24] Samuel Stevens, Emily Wenger, Cathy Li, Niklas Nolte, Eshika Saxena, François Charton, and Kristin Lauter. Salsa fresca: Angular embeddings and pre-training for ml attacks on learning with errors. *arXiv preprint arXiv:2402.01082*, 2024.

[SZM21] Heng-Chuan Su, Xuan-Yong Zhu, and Duan Ming. Polytopic attack on round-reduced simon32/64 using deep learning. In *Information Security and Cryptology: 16th International Conference, Inscrypt 2020, Guangzhou, China, December 11–14, 2020, Revised Selected Papers*, pages 3–20. Springer, 2021.

[Tan23] Quan Quan Tan. *Cryptanalysis of lightweight symmetric-key cryptographic algorithms*. PhD thesis, Nanyang Technical University, Singapore, 2023.

[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.

[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.

[TDD22] Zakaria Tolba, Makhlouf Derdour, and Nour El Houda Dehimi. Machine learning based cryptanalysis techniques: perspectives, challenges and future directions. In *2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–7. IEEE, 2022.

[TDF+22] Zakaria Tolba, Makhlouf Derdour, Mohamed Amine Ferrag, SM Muyeen, and Mohamed Benbouzid. Automated deep learning black-box attack for multimedia p-box security assessment. *IEEE Access*, 10:94019–94039, 2022.

[TH21] Wenqiang Tian and Bin Hu. Deep learning assisted differential cryptanalysis for the lightweight cipher simon. *KSII Transactions on Internet & Information Systems*, 15, 2021.

[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.

[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.

[TTJ23] Wei Jian Teng, Je Sen Teh, and Norziana Jamil. On the security of lightweight block ciphers against neural distinguishers: Observations on lbc-iot and slim. *Journal of Information Security and Applications*, 76:103531, 2023.

[Tu22] Yi Tu. *Machine learning-aided and SAT-aided cryptanalysis of symmetric-key primitives*. PhD thesis, Nanyang Technical University, Singapore, 2022.

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

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wan07] Meiqin Wang. Differential cryptanalysis of present. *IACR Cryptol. ePrint Arch.*, 2007:408, 2007.

[WCCL22] Emily Wenger, Mingjie Chen, Francois Charton, and Kristin E Lauter. Salsa: Attacking lattice cryptography with transformers. *Advances in Neural Information Processing Systems*, 35:34981–34994, 2022.

[Wea47] W Weaver. Letter to norbert wiener, 4 March 1947. https://aclanthology.org/1952.earlymt-1.1.pdf.

[WG24] Wanqing Wu and Mingyu Guo. Improved integral neural distinguisher model for lightweight cipher PRESENT. *Cybersecur.*, 7(1):65, 2024.

[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.

[WNB+23] Ping Wang, Shishir Nagaraja, Aurélien Bourquard, Haichang Gao, and Jeff Yan. Sok: Acoustic side channels. *arXiv preprint arXiv:2308.03806*, 2023.

[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.

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

[WW21] Gao Wang and Gaoli Wang. Improved differential-ml distinguisher: machine learning based generic extension for differential analysis. In *International Conference on Information and Communications Security*, pages 21–38. Springer, 2021.

[WW22] Feifan Wang and Gaoli Wang. Improved differential-linear attack with application to round-reduced speck32/64. In *International Conference on Applied Cryptography and Network Security*, pages 792–808. Springer, 2022.

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

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

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

[WWS24] 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.

[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.

[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 Transactions on Cryptographic Hardware and Embedded Systems*, pages 24–53, 2023.

[YK21a] Tarun Yadav and Manoj Kumar. Differential-ml distinguisher: Machine learning based generic extension for differential cryptanalysis. In *International Conference on Cryptology and Information Security in Latin America*, pages 191–212. Springer, 2021.

[YK21b] Tarun Yadav and Manoj Kumar. Miles: Modeling large s-box in milp based differential characteristic search. *IACR Cryptol. ePrint Arch.*, 2021:1388, 2021.

[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 *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 3–27. Springer, 2022.

[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.

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

[YW24] Xue Yuan and Qichun Wang. Improving differential-neural distinguisher for simeck family. *IACR Cryptol. ePrint Arch.*, page 2002, 2024.

[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. *Cryptology ePrint Archive*, 2023.

[ZG24] Yue Zhong and Jieming Gu. Lightweight block ciphers for resource-constrained environments: A comprehensive survey. *Future Generation Computer Systems*, 2024.

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

[ZL20a] Behnam Zahednejad and Jin Li. An improved integral distinguisher scheme based on deep learning. Technical report, EasyChair, Technical report, 2020.

[ZL20b] Behnam Zahednejad and Jin Li. An improved integral distinguisher scheme based on deep learning. 2020.

[ZL22] Behnam Zahednejad and Lijun Lyu. An improved integral distinguisher scheme based on neural networks. *International Journal of Intelligent Systems*, 37:7584–7613, 2022.

[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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[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*, 2025.

[ZLWL23] Liu Zhang, Jinyu Lu, Zilong Wang, and Chao Li. Improved differential-neural cryptanalysis for round-reduced simeck32/64. *Frontiers of Computer Science*, 17(6):176817, 2023.

[ZW22] Liu Zhang and Zilong Wang. Improving differential-neural distinguisher model for des, chaskey, and present. *arXiv preprint arXiv:2204.06341*, 2022.

[ZWC23] Liu Zhang, Zilong Wang, and Yindong Chen. Improving the accuracy of differential-neural distinguisher for des, chaskey, and present. *IEICE TRANSACTIONS on Information and Systems*, 106:1240–1243, 2023.

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

[ZWW24] Liu Zhang, Zilong Wang, and Baocang Wang. Improving differential-neural cryptanalysis. *IACR Commun. Cryptol.*, 1(3):13, 2024.

[ZZ21] Wenyu Zhang and Yaqun Zhao. Ensemble learning-based differential distinguishers for lightweight cipher. In *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, pages 28–34, 2021.

[ZZC+22] Qingqing Zhang, Hongxing Zhang, Xiaotong Cui, Xing Fang, and Xingyang Wang. Side channel analysis of speck based on transfer learning. *Sensors*, 22:4671, 2022.

[ZZS21] Zimin Zhang, Wenying Zhang, and Hongfang Shi. Genetic algorithm assisted state-recovery attack on round-reduced xoodyak. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part II 26*, pages 257–274. Springer, 2021.

[ZZW24] Weixi Zheng, Liu Zhang, and Zilong Wang. Theoretical explanation and improvement of deep learning-aided cryptanalysis. *Cryptology ePrint Archive*, 2024.

[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 *2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*, pages 529–534. IEEE, 2021.

# A Comparative Review of all Neural Differential Distinguishers

## A.1 AES

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

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.

**Table 2:** 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-$\text{CT}_{\text{tr}}$-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 6.2. 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.

Bellini et al. [BFG+24] used the automatic analysis tool CLAASP [BGG+23] to identify suitable input differences and to subsequently obtain (related-key) differential neural distinguisher for the ARADI block cipher.

**Table 3:** 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 | [BBCD22] |
| ARADI[RK] | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 20M | 2M | ✓ | 6 | 0.5631 | [BBCD22] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.

## A.3 ASCON

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 and 12 rounds in the initialization, and respectively 64 and 128-bit messages, and 6 and 8 rounds in the encryption process. The hash function ASCON-hash, also based on sponge construction, hashes 64-bit messages over 12 rounds. ASCON was announced as the winner of the NIST Lightweight Cryptography Competition in February 2023.

[SSL+24] trained neural differential distinguishers for the 4-round ASCON-PERMUTATION with an accuracy of 0.5069 in the standard setting (2-1-$\delta$-R) and were able to improve the accuracy to 0.6925 by training another neural network to classify based on the distribution of multiple scores. We do not include this result in the table, as it is a system where the neural distinguisher part is run separately on single pairs rather than a neural distinguisher accepting multiple pairs.

**Table 4:** Overview of the Neural Differential Distinguishers for ASCON.

| Primitive | Arch. | Class | Trn. | Val. | AutoND | Rounds | Acc. | Ref. |
|---|---|---|---|---|---|---|---|---|
| ASCON | MLP | 3-2-$\delta$-D | 1.1M | 1.1M | - | 3 | 0.9861 | [BBCD22] |
| | MLP | 2-1-$\delta$-R | 17M | 2M | - | 4 | 0.502 | [YK24] |
| | MLP | 2-1-$\delta$-R | 20M | 20M | - | 4 | 0.5069 | [SSL+24] |
| ASCON[Unkeyed] | Classical ML | 2-2-$\delta$-D | 64K | 16K | - | 3 | 0.916 | [BBD+23][‡] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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 (-).
[‡] The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.4 CHASKEY

CHASKEY is a 128-bit ARX-based permutation on 8 rounds.
In [CSYY23], the best distinguisher used 16 pairs per sample, though the authors presented a valid single-pair distinguisher for CHASKEY and other ciphers as well. In several ciphers, the authors observed decreasing accuracy as $n$ increases, which starkly contrasts with established findings in the literature.

**Table 5:** 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 | [BBCD22][‡] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 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 6.2. 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 (-).
[‡] The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.5 DES

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

**Table 6:** 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 6.2. 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] used 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.

In [KKJ+24], the authors performed 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 but note that the authors additionally performed experiments in the $m$-2-CT-D setting with similar, yet not directly comparable, results. Experiments were conducted for the classification of up to 15 input differences. However, it is not immediately clear which results are the best. The number of samples for training and test was not given, nor is the source code (/-entries in Table 7).

**Table 7:** 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 6.2. 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.7 GIFT

GIFT is a PRESENT-inspired SPN cipher, using 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.

In [ZZY+21]†, the authors claimed 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 rounds distinguisher as it is the best one claimed by the authors, but also their 6-round distinguisher, which has a significant validation accuracy.

In [MPKM+22]†, the authors claimed a full round distinguisher on GIFT-64 with over 90% accuracy, using $2^{20}$ polytopic samples (composed of 3 ciphertexts each) in total, of which 15% are kept for validation, respectively testing, and a simple MLP architecture; they also claimed 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 [RRSM22a], only 10K samples were used for training and test; as a result, the distinguishers in Table 5 exhibit significant overfitting (e.g., 92% training accuracy and 25% test accuracy for M1 on 6 rounds).

**Table 8:** 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]‡ |
| | LSTM | 3-2-CT-R | 17M | 4M | - | 6 | 0.5754 | [SSL+22] |
| | MLP | 3-2-$\delta$-R | 2.2M | 500K | - | FULL | 0.96 | [MPKM+22]† |
| GIFT-128 | MLP | 2-1-$\delta$-R | 17M | 2M | - | 7 | 0.55 | [YK24] |
| | MLP | 2-1-$\delta$-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-$\delta$-D | 20K | 20K | - | 4 | 0.615 | [RRSM22a]‡ |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## 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.

**Table 9:** 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 | [BGH+23] |
| GIMLI-HASH | MLP | 3-2-$\delta$-D | 400K | 40K | - | 8 | 0.5219 | [BBCD22]‡ |
| GIMLI-CIPHER | MLP | 3-2-$\delta$-D | 400K | 40K | - | 8 | 0.5099 | [BBCD22]‡ |

**Class:** *n-m-T-E*, from Subsection 6.2. 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 (-).
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## 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 10:** 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 6.2. 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.

## A.10 HIGHT

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

Seok et al. [Seo24] achieveded a distinguishing accuracy of 0.5707 on 10-round HEIGHT by analyzing only half of the ciphertext difference (4 out of 8 bytes). These specific bytes were selected through their analysis of the HIGHT round function.

Bose et al. [BPC24] † claimed advancements in distinguishing additional encryption rounds through sequential model training on ciphertext pairs. However, these findings contradict established cryptographic theory, which shows distinguishability decreases monotonically with increasing rounds – a pattern absent in their results. Notably, Bellini et al. [BGH+23] reported superior distinguishers for rounds 9 and 10 of HIGHT, challenging the purported architecture's effectiveness in detecting differential patterns. Given these theoretical and empirical inconsistencies, independent verification is necessary before accepting these anomalous results. We report the best distinguishers with statistical significance at the 95% confidence level (z-scores $> 1.96$, $p < 0.05$), corresponding to accuracies above 0.5011 on a test set of $5 \times 10^6$ samples.

**Table 11:** 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-$\delta_{\text{tr}}$-R | 20M | 2M | - | 10 | 0.5707 | [Seo24] |
| | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 10 | 0.751 | [BGH+23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 2M | 200K | - | 11 | 0.7472 | [TSL23] |
| | LSTM | 2-1-A-R | 10M | 500K | - | 15 | 0.5015 | [BPC24] † |
| HIGHT$^{\text{RK}}$ | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 14 | 0.563 | [BGH+23] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 14 | 0.640 | [WWS24] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.

## A.11 KATAN

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

For KATAN32, [BGH+23] 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 $0.5034 \pm 0.0002$ accuracy using their simple polishing step. In contrast, [LCLH22] reached 51 rounds in the standard setting and 59 when using 64 pairs.

In [LLHC23, LCLH22], the authors enhanced their neural distinguishers by prepending an $r$-round conditional differential that holds with probability 1 to an $s$-round neural distinguisher. While we focus solely on pure differential distinguishers in our analysis, it is worth noting that in [LLHC23], the neural distinguishers were specifically trained under the assumption that the conditional differential holds. This constraint on the input distribution enabled the distinguishers to achieve higher accuracy.

In [LLHC23], these distinguishers lead to practical key recovery on 97, 82, 70 rounds of KATAN32, 48 and 64 in the single key model. In [LCLH22], practical key recoveries were obtained for 125, 106 and 95 rounds respectively, in the related key scenario. Single-key conditional neural distinguishers were also mentioned in [LCLH22] for 85, 72 and 61 rounds respectively, but the $r + s$ decomposition was not explicitly mentioned so we omit them in the table.

**Table 12:** Overview of the Neural Differential Distinguishers for KATAN.

| Primitive | Arch. | Class | Trn. | Val. | AutoND | Rounds | Acc. | Ref. |
|---|---|---|---|---|---|---|---|---|
| KATAN32 | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-$\delta$-R | 20M | 2M | - | 51 | 0.533 | [LCLH22] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 59 | 0.575 | [LCLH22] |
| | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 69 | 0.505 | [BGH+23] |
| KATAN32 $^{C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-$\delta$-R | 64M | 6.4M | - | 58 | 0.602 | [LLHC23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 85 | 0.570 | [LCLH22] |
| KATAN32 $^{RK,C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 112 | 0.647 | [LCLH22] |
| KATAN48 | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-$\delta$-R | 20M | 2M | - | 40 | 0.58 | [LCLH22] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 96-1-$\delta$-R | 960M | 96M | - | 50 | 0.54 | [LCLH22] |
| KATAN48 $^{C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-$\delta$-R | 64M | 6.4M | - | 47 | 0.582 | [LLHC23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 96-1-$\delta$-R | 960M | 96M | - | 72 | 0.582 | [LCLH22] |
| KATAN48 $^{RK,C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 48-1-$\delta$-R | 960M | 96M | - | 96 | 0.625 | [LCLH22] |
| KATAN64 | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-$\delta$-R | 20M | 2M | - | 31 | 0.718 | [LCLH22] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 36 | 0.548 | [LCLH22] |
| KATAN64 $^{C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-$\delta$-R | 64M | 6.4M | - | 26 | 0.613 | [LLHC23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 61 | 0.613 | [LCLH22] |
| KATAN64 $^{RK,C}$ | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-R | 1280M | 128M | - | 86 | 0.728 | [LCLH22] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.

## 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.

In [BBCD22], the authors useed a neural distinguisher to recognize whether a 1 difference is introduced in the first or the second byte.

**Table 13:** Overview of the Neural Differential Distinguishers for KNOT.

| Primitive | Arch. | Class | Trn. | Val. | AutoND | Rounds | Acc. | Ref. |
|---|---|---|---|---|---|---|---|---|
| KNOT-256 | MLP | 3-2-$\delta$-D | 1.6M | 1.6M | - | 10 | 0.5912 | [BBCD22] |
| KNOT-512 | MLP | 3-2-$\delta$-D | 1.6M | 1.6M | - | 12 | 0.6032 | [BBCD22] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.13 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. For LEA, [BGH+23] propose the first neural differential distinguisher, reaching 11 rounds with accuracy 0.5109. In comparison, the proposal of LEA [HLK+14] presents a differential characteristic with probability $2^{-98}$ for 11 rounds, and $2^{-128}$ for 12 rounds.

Bose et al. [BPC24] † claimed advancements in distinguishing additional encryption rounds through sequential model training on ciphertext pairs. However, these findings contradict established cryptographic theory, which shows distinguishability decreases monotonically with increasing rounds – a pattern absent in their results. Notably, Bellini et al. [BGH+23] reported superior distinguishers for rounds 9 and 10 of HIGHT, challenging the purported architecture's effectiveness in detecting differential patterns. Given these

theoretical and empirical inconsistencies, independent verification is necessary before accepting these anomalous results. We report the best distinguishers with statistical significance at the 95% confidence level (z-scores $> 1.96$, $p < 0.05$), corresponding to accuracies above 0.5011 on a test set of $5 \times 10^6$ samples.

**Table 14:** 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 | [BGH+23] |
| | Transformer | 2-1-A-R | 10M | 500K | - | 13 | 0.5012 | [BPC24] † |

**Class:** *n-m-T-E*, from Subsection 6.2. 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 LBCIoT

LBCIoT is a 32-round block cipher encrypting 32-bit plaintexts with an 80-bit key. In [TTJ23], the authors propose a neural distinguisher on 7 rounds and build a practical key recovery attack for 8 rounds.

**Table 15:** 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 6.2. 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 (-).
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## 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.

In [BGH+23], a 9-round distinguisher with an accuracy of 0.5092 was given, which favorably compares to the 7-round distinguishers of [CSYY23], despite [CSYY23] using four pairs per sample. On the other hand, [ZW22] obtained a slightly higher accuracy at the cost of using 32 ciphertexts per sample. In comparison, the best differential characteristic for PRESENT reduced to 9 rounds has probability $2^{-36}$ [Wan07].

Bose et al. [BPC24] † claimed advancements in distinguishing additional encryption rounds through sequential model training on ciphertext pairs. However, these findings contradict established cryptographic theory, which shows distinguishability decreases monotonically with increasing rounds – a pattern absent in their results. Notably, Bellini et al. [BGH+23] reported superior distinguishers for rounds 9 and 10 of HIGHT, challenging the purported architecture's effectiveness in detecting differential patterns. Given these theoretical and empirical inconsistencies, independent verification is necessary before accepting these anomalous results. We report the best distinguishers with statistical significance at the 95% confidence level (z-scores $> 1.96$, $p < 0.05$), corresponding to accuracies above 0.5011 on a test set of $5 \times 10^6$ samples.

**Table 16:** 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 | [BGH+23] |
| | CNN | 2-2-$\delta$-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 | [BPC24] † |
| PRESENT-64/80 RK | MLP | 6-1-$\delta$-R | 4.2M* | 1.9M* | - | 5 | 0.614 | [PSM23] |
| | CNN | 2-2-$\delta$-D | 20M | 2M | - | 10 | 0.517 | [WWH21] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.16 PRIDE

PRIDE is a 20-round SPN cipher using 64-bit blocks and 128-bit keys.
In [MPKM+22], the authors claimed a full-round distinguisher on the cipher with 100% accuracy, which seems likely to be attributed to a methodology issue than an actual break, as a perfect accuracy is often a sign of overfitting, especially considering the lack of evidence provided in the paper.

**Table 17:** Overview of the Neural Differential Distinguishers for PRIDE.

| Primitive | Arch. | Class | Trn. | Val. | AutoND | Rounds | Acc. | Ref. |
|---|---|---|---|---|---|---|---|---|
| PRIDE | MLP | 2-1-$\delta$-R | 734K | 157K | - | 20 | 1 | [MPKM+22]‡ |

**Class:** *n-m-T-E*, from Subsection 6.2. 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 (-).
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.17 SHA3

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

**Table 18:** 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 6.2. 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 SIMECK

SIMECK is a variant of SIMON using a key schedule similar to that of SPECK. SIMECK-32/64, SIMECK 48/96, and SIMECK-128/256 have 32, 36, and 44 rounds, respectively.

In [ZLWL23], the authors used an inception-based architecture and utilized 8 pairs per sample, with the special format $(\Delta x_r, \Delta y_r, x_r, y_r, x'_r, y'_r, \Delta y_{r-1}, p\Delta y_{r-2})$; the authors used the staged training approach proposed by Gohr in [Goh19a]. Their best distinguisher reached 12 rounds of SIMECK32.

In [WTZ+22], the authors investigated two variations of a multiple input differences scenario, where the samples are the concatenations of pairs with differences $\delta_i$. In $\text{ND}_{rm}$, a sample is the concatenation of a pair of ciphertexts for each difference (resulting in $n = 2m$); in $\text{ND}_{am}$, the first ciphertext is the encryption of a random plaintext $P_0$, each subsequent ciphertext $C_i$ is the encryption of $P_{i-1} \oplus \Delta_{i-1}$ so that $n = m + 1$. The distinguishers were trained on $2^{24}$ (16.8M) samples, each containing 4 ciphertexts, and tested on $2^{18}$ (0.3M). The accuracy of 50.42% may not be statistically significant and should be indicated with a mean and standard deviation on fresh sets of test samples.

In [LTZ22b], the authors proposed training a neural distinguisher multiple times independently and selecting the model with the highest test accuracy. Notably, they reported successfully obtaining a single 10-round Simeck distinguisher in one out of 20 independent training attempts. Furthermore, they employed a Mixed-Integer Linear Programming (MILP) model to identify a highly probable differential, which was then combined with the neural distinguisher.

In [WW24a], Wang et al. constructed related-key neural distinguishers to distinguish two differences ($E = D$). They proposed a greedy and exhaustive search for optimal input difference combination based on the bias score proposed in [GLN22].

**Table 19:** 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-A-R | 67M | 1M | - | 11 | 0.5042 | [WTZ+22] |
| | DenseNet | 2-2-CT-D | 2020M | 2M | ✓ | 12 | 0.505 | [WWS24] |
| | SE-ResNet | 16-1-A-R | 1394M | 134M | - | 12 | 0.5146 | [LLS+24] |
| | INC | 16-1-A-R | 32480M | 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-$\delta$-D | 66K | 66K | - | 9 | 0.526 | [BBD+23]‡ |
| SIMECK-48/96 | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 15 | 0.505 | [WWS24] |
| 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 | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 18 | 0.507 | [WWS24] |
| | 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-$\delta$-D | 66K | 66K | - | 14 | 0.55 | [BBD+23]‡ |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.19 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 20:** 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] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 20M | 2M | - | 9 | 0.5907 | [HRC21c] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 20M | 2M | - | 9 | 0.6277 | [SZM21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | / | / | - | 9 | 0.6320 | [TH21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 4-3-CT-R | 40M | 4M | - | 9 | 0.6373 | [SZM21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 4-3-CT-R | 40M | 4M | - | 8 | 0.923 | [WQW+24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-$\delta$-R | 640M | 6.4M | - | 10 | 0.6109 | [HRC21c] |
| | SENet | 2-1-A-R | 4852M | 537M | - | 11 | 0.517 | [BGL+22] |
| | DBitNet | 2-1-CT-R | 2020M | 2M | ✓ | 11 | 0.518 | [BGH+23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 64-1-A-R | 640M | 64M | - | 11 | 0.6081 | [LRCL23] |
| | DenseNet | 2-2-CT-D | 2020M | 2M | ✓ | 12 | 0.505 | [WWS24] |
| | SE-ResNet | 16-1-A-R | 160M | 16M | - | 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 | 160M | 16M | - | 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-$\delta$-R | 960M | 9.6M | - | 11 | 0.6143 | [HRC21c] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 12 | 0.515 | [WWS24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 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-$\delta$-R | 1280M | 12.8M | - | 12 | 0.6957 | [HRC21c] |
| | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 13 | 0.518 | [BGH+23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-A-R | 1280M | 128M | - | 13 | 0.701 | [LRCL23] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 14 | 0.506 | [WWS24] |
| | SE-ResNet | 16-1-A-R | 1610M | 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 | 160M | 16M | - | 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 | [BGH+23] |
| 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 6.2. 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.
RK Related key setting.

Table 20 provides an overview of the differential neural distinguishers developed for the SIMON family of block ciphers. The most extensively studied variant is SIMON-32, with various neural network architectures and settings explored across multiple works. In the standard setting, the best distinguisher achieves round 11 through an automated pipeline [BGH+23]. By using multiple ciphertext pairs ($n = 16, 32, 64$) and employing advanced feature engineering techniques, as in [LRCL23, LLS+24, ZWW24], the distinguisher performance surpasses this result, extending the analysis to round 12 [LLS+24].

For the case of SIMON, some authors experimented with a vast amount of data: [HRC21c] used $k \cdot 10^7$ for $k = 32, 48, 64$ (maximum of 640M) pairs for training, and [BGL+22] obtained an 11-round distinguisher for SIMON32 at the cost of staged trained in four steps, with respectively $10^7$, $2^{28}$, $2 \cdot 2^{30}$ (2426M pairs). In [BGH+23], the authors proposed a polishing step, retraining a neural distinguisher initially trained with $10^7$ pairs with an additional $10^9$ pairs: $10^7$, $310^9$ (1010M pairs). This polishing step was also used by Wang et al. [WWS24]. Similarly, Zhang et al. [ZWW24] used a staged training approach: $4 \cdot 10^7$ samples, each sample with 16 pairs (640M pairs).

In [LLS+24], Lu *et al.* used advanced feature engineering and 80M ciphertext pairs ($10^7$ samples, each composed of 8 pairs) and reached 12 rounds of SIMON32 in the single-key scenario. In the related key scenario, the same authors obtained a 13-round distinguisher, whereas [EGP23] only reached 11 rounds with a rotational XOR distinguisher. The feature engineering proposed in [LLS+24] was also used in [WW24a]. Further, the authors used staged training for a subset of the obtained distinguishers: $3 \cdot 2^{25}$ samples, 8 pairs each

(805M pairs).

## A.20 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.
In [BBD+23], the authors reach 7 rounds of SKINNY-128; however, this result is obtained on an unkeyed version of the cipher and using a classical machine learning algorithm rather than deep learning.

**Table 21:** Overview of the Neural Differential Distinguishers for SKINNY.

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

**Class:** *n-m-T-E*, from Subsection 6.2. 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 (-).
‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.21 SLIM

SLIM is a 32-round block cipher encrypting 32-bit plaintexts with an 80-bit key.
In [RLS23], 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.

In [TTJ23]†, the reported accuracy is 0.5036 on $10^5$ samples, which corresponds to less than 3 standard deviations and has a probability over 1% of occurring for distinguisher making predictions at random; we question the relevance of this result, as testing on more data is required to prove statistical significance.

**Table 22:** 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 | [RLS23] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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.

## A.22 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. Neural differential distinguishers have been developed for all versions of SPECK, and a comprehensive overview of these is presented in Table 23.

In the standard setting (2-1-CT-R) for SPECK-32, Gohr's original analysis on 8 rounds remains unmatched, in which the author applied a staged training: $2 \cdot 10^7$, $2 \cdot 10^9$

**Table 23:** 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]† |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{sep.conv.}}$ | 2-1-$\delta_{\text{tr}}$-$R$ | 10M | 1M | - | 6 | 0.673 | [LRC23] |
| | MLP | 2-1-$\delta_{\text{tr}}$-R | 20M | 2M | - | 6 | 0.688 | [ERP22] |
| | MLP | 2-1-$\delta$-R | 20M | 2M | - | 6 | 0.72 | [ERP22] |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{ensmbl.}}$ | 2-1-CT-R | 20M | 2M | - | 6 | 0.781 | [HGH+23] |
| | CNN | 100-1-A-R | 20M | 2M | - | 6 | 1 | [BGPT21] |
| | DenseNet | 2-1-CT-R | 2M | 2M | - | 7 | 0.531 | [SM23b]† |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{pruned}}$ | 2-1-CT-R | 20M | 2M | - | 7 | 0.596 | [BBP22]† |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-$\delta$-R | 20M | 2M | - | 7 | 0.583 | [BGPT21] |
| | CNN | 2-2-$\delta$-D | 20M | 2M | - | 7 | 0.599 | [WWH21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 2M | / | ✓ | 7 | 0.614 | [WW21] |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{attntn.}}$ | 2-1-CT-R | 20M | 2M | - | 7 | 0.6169 | [DCC23] |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{sep.conv.}}$ | 8-1-CT-R | 80M | 8M | - | 7 | 0.6939 | [LRC23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 16-1-CT-R | 20M | 2M | - | 7 | 0.7009 | [CSYY23] |
| | $\mathcal{ND}_{\text{Gohr}}^{\text{attntn.}}$ | 16-1-CT-R | 160M | 16M | - | 7 | 0.728 | [DCC23] |
| | INC | 64-1-A-R | 64M | 6.4M | - | 7 | 0.9713 | [YW23] |
| | INC$^{\text{freeze}}$ | 2-1-CT-R | 20M | 2M | - | 8 | 0.5135 | [BLYZ23] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 4040M | 2M | - | 8 | 0.514 | [Goh19b] |
| | DBitNet | 2-1-CT-R | 2020M | 2M | ✓ | 8 | 0.514 | [BGH+23] |
| | DenseNet | 2-2-CT-D | 2020MM | 2M | ✓ | 8 | 0.519 | [WWS24] |
| | MLP | 2-1-CT-$R^+$ | 20M | 2M | - | 8 | 0.5208 | [LJSC24a] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-A-R | 1280M | 128M | - | 8 | 0.6502 | [LRCL23] |
| | MLP | 512-1-CT-$R^+$ | 20M | 2M | - | 8 | 0.8866 | [LJSC24a] |
| | INC | 32-1-$A$-$R$ | 1280M | 2M | - | 9 | 0.5045 | [ZWW24] |
| SPECK-32 RK | CNN | 2-2-$\delta$-D | 20M | 2M | - | 7 | 0.559 | [WWH21] |
| | CNN | 2-2-CT-R | 20M | 2M | - | 7 | 0.576 | [WWH21] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 2M | 200K | - | 9 | 0.5932 | [TSL23] |
| | INC$^{\text{freeze}}$ | 2-1-CT-R | 20M | 2M | - | 10 | 0.5562 | [BLYZ23] |
| SPECK-32$^{\text{Unkeyed}}$ | MLP | 2-2-$\delta$-D | 66K | 66K | - | 8 | 0.515 | [BBD+23]‡ |
| SPECK-48 | $\mathcal{ND}_{\text{Gohr}}$ | 2-1-CT-R | 2M | / | ✓ | 6 | 0.726 | [WW21] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 8 | 0.506 | [WWS24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 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 | [BGH+23] |
| | DenseNet | 2-2-CT-D | 20M | 2M | ✓ | 8 | 0.559 | [WWS24] |
| | $\mathcal{ND}_{\text{Gohr}}$ | 128-1-$\delta$-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 | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 10 | 0.593 | [BGH+23] |

**Class:** $n$-$m$-$T$-$E$, from Subsection 6.2. 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.

RK Related key setting.

‡ The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

‡ 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.

(2020M pairs). However, [BGH+23] achieved comparable accuracy to [Goh19b] using an automated, generic pipeline that is not specifically tailored to SPECK[6] and lacks the

[6]We note that [BLYZ23] stated that "the simple training pipeline [of [BGH+23]] 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" which is not entirely correct. While in [BGH+23], a

complex training scheme essential for high accuracy on 8 rounds. More precisely, the authors proposed a polishing step, retraining a neural distinguisher initially trained with $10^7$ pairs with an additional $10^9$ pairs: $10^7$, $3 \cdot 10^9$ (3010M pairs in total). Enhanced accuracy over Gohr's results on 8-round SPECK-32 can be achieved by employing a larger dataset, advanced feature engineering, and using multiple ciphertext pairs (e.g., $n = 128$) as shown in [LRCL23]. Similarly, for SPECK-32 with fewer than 8 rounds, higher accuracy is reported when using multiple ciphertext pairs: [CSYY23] uses $n = 16$, [HRC21c] uses $n = 64$, and [LRCL23] uses $n = 128$.

In terms of larger state experiments, two automated pipelines reached 7, respectively 8 rounds of SPECK-64 [WW21, BGH+23]. The 8-round accuracies can be improved when increasing the number of ciphertext pairs to $n = 128$, respectively $n = 256$, and using MRMSD feature engineering [HRC21c, LRCL23]. For SPECK-96, [CSY23] obtained the first 7-round distinguisher, while for SPECK-128, [BGH+23] obtained the first 10-round neural distinguisher within an automated pipeline.

In [ERP22], the authors reported an accuracy of 0.688 while using only 8 bits of the 32-bit ciphertext difference, identified through a bit scoring algorithm. Building on this, [LRC23] proposed a novel advantage bit search algorithm that incorporates symmetric and differential conditions. This algorithm led to an accuracy of 0.673 on 6 rounds of SPECK-32/64 encryption while still utilizing just 8 bits of the 32-bit ciphertext difference. Further, the authors reported a 50% reduction in training parameters without any loss in network accuracy using separable convolutions. Although the partial output difference neural distinguisher achieved slightly lower accuracy, it significantly reduced the amount of training data required. Liu *et al.* claimed that the result in [ERP22] was obtained achieved using 16 bits, not 8, and assert that their method offers a significant improvement in data complexity.

In [KJL+23]†, the author reported an accuracy of 53% (round 5) on only 1,000 validation samples. The experimental mean or standard deviation was not given. The statistically expected standard deviation for a binomial experiment on 1k samples is $1/(2\sqrt{n}) = 1.6\%$. Therefore, the reported result is only $1.9\sigma$ away from random and is likely not statistically significant.

[SM23b]† reported an accuracy of 53.1% (round 7) on 2M training, respectively validation samples and provides a comparison in which DenseNet outperforms $\mathcal{ND}_{\text{Gohr}}$. At such a small number of training samples, both networks show heavy overfitting ([SM23b, Table 2]), and the authors themselves called the result only "marginal."

Wang and Wang [WWS24] developed a neural distinguisher to differentiate between output distributions generated by two distinct input differences in the related-key setting. Their approach aimed to maximize the average absolute distance between these output distributions—a key metric for distinguisher performance, as established by Gohr [GLN22]. While they extend their analysis to multiple-pair distinctions, we focus on their single-pair distinguishers, which form the core of their experimental work. They further built upon the automation proposed in [BGH+23] to automate the training of neural distinguishers to differentiate between output distributions generated by two distinct input differences. As the size of the validation set has not been explicitly mentioned by the authors, we assume that they follow the size in [BGH+23].

Lv et al. [LJSC24a] demonstrated that differential-linear cryptanalysis can produce neural distinguishers surpassing the state-of-the-art distinguishers based solely on differential cryptanalysis. Their methodology involved an exhaustive search over differential-linear approximations with low Hamming weights, filtering out the most influential approximations using importance metrics from the Light Gradient Boosting Machine (LGBM) classification algorithm. However, the authors incorrectly asserted that multi-pair neural

polishing step is indeed needed to achieve the same accuracy, the polishing step is a *highly simplified and automated* version of the 8-round training scheme used by Gohr (in conclusion, it does *not* demand more time or data).

differential distinguishers necessitate architectural modifications. This assertion directly contradicts the work of Gohr [GLN22], which explicitly presented a concrete methodology for constructing multi-pair distinguishers from single-pair architectures without requiring structural changes. Notably, the input difference employed in their 8-round distinguisher exhibits significant divergence from conventional patterns established in current literature, raising interesting questions about optimal difference selection in neural cryptanalysis.

Huang et al. [HLF+24] used 24 out of 64 ciphertext bits for their partial distinguishers ($T = A$ and $T = \mathrm{CT}_{\mathrm{tr}}$) and an advanced feature engineering that partially inverts 7-round ciphertexts to 6 rounds and combines the information to a sample. For SPECK64, their partial distinguisher was trained in a staged fashion: $4 \cdot 10^7$, 16 pairs each (640M pairs).

Zhang et al. [ZWW24] used a staged training approach: $4 \cdot 10^7$ samples, each sample with 16 pairs (640M pairs).

## A.23 TEA and XTEA

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

In [BR21], the authors considered modular addition-based differentials, where the input difference is injected by modular addition, which we denote by $R^+$ as the experiment. [BGH+23] automatically found distinguishers for both TEA and XTEA for 5 cycles (10 rounds), respectively, with accuracies 0.5634 and 0.5984; the authors noted that they interestingly share the same input difference. For TEA, [BGH+23] reached two more rounds than [BR21].

**Table 24:** 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 | [BGH+23] |
| XTEA | DBitNet | 2-1-CT-R | 20M | 2M | ✓ | 10 | 0.598 | [BGH+23] |

**Class:** *n-m-T-E*, from Subsection 6.2. 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 (-).
[‡] The use of a small validation set raises concerns about the statistical robustness, reproducibility, and generalizability of the results, as such datasets are prone to high variance and may not reliably reflect model performance.

## A.24 TinyJAMBU

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

**Table 25:** Overview of the Neural Differential Distinguishers for TinyJAMBU.

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

**Class:** *n-m-T-E*, from Subsection 6.2. 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.

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[7]: 'When nonce is reused, an attacker can decrypt the ciphertext since the encryption of TinyJAMBU is somehow similar to the Cipher Feedback mode.'.

# B Optimizations for Neural Aided Key Recovery

This section discusses the various optimizations proposed in [Goh19a] to neural aided key recovery, including round extension via probabilistic differentials, computational cost reduction through Bayesian Optimization and Upper Confidence Bounds, and performance enhancement with an additional verification step.

**Extending the Rounds Covered by the Distinguisher.** For ciphers like Speck or Simon, neural distinguishers can gain a free round when initial subkey addition follows the first nonlinearity. This is achieved by selecting plaintext pairs that deterministically yield ciphertext differences matching the neural distinguisher's trained input difference $\delta$.

Additionally, prepending a classical $s$-round differential transition to an $r$-round neural distinguisher creates an $(s+r)$-round differential distinguisher. Note that the choice of differential transitions involves a fundamental tradeoff between data complexity and rounds. In [Goh19a], the selected 2-round differential transition has probability $\frac{1}{64}$. While this requires (on average) 64 times more encryptions than the original attack, it extends the 7-round neural distinguisher to a 9-round combined distinguisher.

When prepending a classical differential to a distinguisher, aggregating scores across multiple ciphertext pairs fails due to non-conforming pairs introducing random noise. *(Probabilistic) Neutral Bits* (PNBs) improve the signal-to-noise ratio. For a neutral bit $i$, pairs $p_1, p_2$ satisfying differential $\delta \to \Delta$ imply $p_1 \oplus (1 \ll i), p_2 \oplus (1 \ll i)$ also follow the differential with probability 1 (or high probability for PNBs). Hence, with $j$ PNBs, a structure of $2^j$ plaintext pairs can be constructed, such that either all pairs in the structure conform to the prepended differential or none do, ensuring consistent scoring. This concept was later extended to conditional simultaneous neutral bit-sets and switching bits for adjacent differentials [BGL+21].

**Reducing Computational Cost.** In [Goh19a], the author reduces computational complexity using Bayesian Optimization for key search. The approach first builds a *Wrong Key Response Profile* (WKRP): For a ciphertext pair $(c_1, c_2)$ from plaintexts $p$ and $p \oplus \delta$, with correct $r$-round key $k$ and candidate key $k' = k \oplus \gamma$, the distinguisher response is modeled as:

$$R_{D,\gamma}(c_1, c_2) = D\left(f_{k'}^{-1}(c_1), f_{k'}^{-1}(c_2)\right).$$

$R_{D,\gamma}$ follows a normal distribution with mean $\mu_\gamma$ and standard deviation $\sigma_\gamma$.

The key search iteratively queries the neural network using an acquisition function based on the WKRP to select a set of candidates. Given observed responses $R_1, \ldots, R_n$ for key candidates $k'_1, \ldots, k'_n$, the function chooses new candidates $k$ that minimize:

$$\sum_{i=0}^{n-1} \frac{(R_i - \mu_{k \oplus k'_i})^2}{\sigma^2_{k \oplus k_i}}.$$

This aligns the precomputed wrong key response profile optimally with the observed values and typically requires only a few iterations until key candidates with high scores are identified, compared to the basic attack enumerating over all possible round keys.

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

To prevent unnecessary computation, Upper Confidence Bounds (UCB) serve as stopping conditions. For $t$ independent encryption oracles $F_K^{r+1}$, the attack prioritizes the instance with the highest value:

$$sk = w_{\max}^i + \alpha \cdot \sqrt{\frac{\log_2(j)}{n_i}},$$

where $w_{\max}^i$ is the highest distinguisher score for instance $i$, $n_i$ is iterations spent on instance $i$, $j$ is the current iteration, and $\alpha = 10$. This tradeoff helps balancing exploration and exploitation, focusing more on instances that are either insufficiently explored or show high key candidate scores.

**Additional Verification.** Incorrect key candidates with Hamming distance 1-2 from the correct key often produce high responses. Searching this narrow Hamming neighborhood for the key candidate output by the key searching algorithm typically reveals the correct key with marginally higher scores, prompting search restarts with each improved candidate.

Gohr [Goh19a] jointly recovered $(r+1)$-round and $r$-round keys by combining distinguishers $D_r$ and $D_{r-1}$. When an $(r+1)$-round key guess exceeds threshold $t_1$, the $r$-round key is searched using $D_{r-1}$, with both keys returned if an $r$-round guess surpasses threshold $t_2$. This creates inherent verification, as incorrect $(r+1)$-round keys rarely produce high scores for any $r$-round key guess.

# C Overview: Neural Aided Key Recovery Attacks

Gohr's work [Goh19a] marked a breakthrough in ML-based cryptanalysis, achieving high-accuracy neural distinguishers for 7-round SPECK32/64 and developing key recovery attacks for 11 and 12 rounds that rivaled or surpassed state-of-the-art manual techniques.

Since then, research has progressed in multiple directions, including applying the proposed key recovery algorithm to various cryptographic primitives [TH21, WQW+24, TTJ23, YW23, ZLHH25, ZWW24], proposing enhancements to the original algorithm [LTZ22a, CSY23, BGL+22, ZLWL23, SZM21, HRC23, CSYY23, LJSC24a], exploring key recovery in alternative adversarial settings [LCLH22, LLHC23, BLYZ23], and reducing the complexity of the attack by truncating the ciphertexts observed by the distinguishers [CSY23, HLF+24, LRC24, Seo24, ZWL24]. We specifically highlight (†) papers that implement a (full) Bayesian attack compared to those employing a simplified basic attack.

### C.0.1 Key Recovery on Different Cryptographic Primitives

While the first neural-aided key recovery was performed on SPECK32/64, subsequent works applied the same or a simplified version of the attack to SPECK [ZWW24], SIMON [TH21, WQW+24, ZWW24], LBC-IoT [TTJ23], SLIM [TTJ23], SPECK [YW23], and PRESENT [ZLHH25].

Zhang *et al.* [ZWW24]† achieved significant breakthroughs in differential-neural cryptanalysis by performing key recovery attacks on 13- and 14-round SPECK32/64, with the 14-round attack brute-forcing the final round's subkey. They also executed the first 17-round key recovery attack on SIMON32/64. Building upon Gohr's foundational work [Goh19a], the authors implemented knowledge distillation to create dramatically smaller student networks featuring fixed-size convolutions and GlobalAveragePooling layers. These streamlined architectures substantially reduced computational demands during key recovery while maintaining attack effectiveness.

Tian and Hu [TH21] developed 7-9 round neural distinguishers for SIMON32/64 and achieved 15-round key recovery using a prepended differential with probabilistic neutral bits, followed by brute-force subkey search.

Teng *et al.* [TTJ23] demonstrated practical 8-round key recovery attacks on LBC-IoT using their 6-round neural distinguisher.

Wu *et al.* [WQW+24]† introduced a mixed-neural differential network for 12-round SIMON32/64 key recovery, achieving higher accuracy with increased complexity.

Yue and Wu [YW23] improved upon Gohr's work with a novel data format exploiting SPECK32/64's round function structure, enabling 8-round key recovery.

Zhu *et al.* [ZLHH25] successfully executed an 8-round key recovery attack on PRESENT by extracting non-linear S-box features using randomly generated subkeys, demonstrating that neural networks trained on ciphertext differences substantially outperform those trained on raw ciphertext pairs for distinguishing and key recovery tasks.

### C.0.2 Advancements of the Key Recovery

Several studies aimed to advance neural-aided key recovery by focusing on parameter selection [LTZ22a, CSY23, BGL+22], exploring variants of neutral bits in the prepended classical differential [BGL+22, ZLWL23, SZM21], and reducing data complexity through precomputation [HRC23, LJSC24a] and reducing encryption queries [CSYY23].

Lyu *et al.* exhaustively explored neural distinguishers for Bayesian key search and applied them to SIMECK32/64 [LTZ22a]†. They obtained 8/9/10-round neural differential distinguishers and recovered penultimate and last round subkeys for 13/14/15-round SIMECK32/64 with low data and time complexity. Their findings revealed that key response profile regularity plays a crucial role (varying greatly among distinguishers), as does the number of neutral bits available for the distinguisher's prepended differential. Interestingly, the most accurate neural distinguisher did not necessarily achieve the best key recovery performance.

Chen *et al.* proposed a Neural-Aided Statistical Attack (NASA) with experiments on reduced-round SPECK32/64, DES, and Speck96/96 [CSY23]†. Their theoretical estimates suggest breaking 10-round DES, surpassing Gohr's 8-round attack. When combined with a novel early stopping technique, neutral bits, and a Bayesian algorithm in the lines of [Goh19a], their method reduces both computational and data complexity compared to the original key recovery in [Goh19a].

Bao *et al.* introduced generalized neutral bits techniques and conditional neural differential cryptanalysis [BGL+22]†. They improved the success rate of deep learning-assisted key recovery attacks by considering neural distinguisher accuracies, round numbers, and classical differential paths spliced in front of neural distinguishers. They also explored data complexity aspects and achieved successful key recovery attacks on 13-round SPECK32/64 and 16-round SIMON32/64.

In [ZLWL23]†, the authors improved SIMECK-32 attacks, enhancing the 15-round attack and launching the first practical 16- and 17-round key recovery attacks for SIMECK32/64. They extended their 12-round neural distinguisher with a 3-round differential and associated 14 deterministic NBs and 2 SNBSs (simultaneously complementing up to 4 bits) identified through exhaustive search.

In [SZM21]†, the authors implemented full key recovery on Simon32/64 using a distinguisher trained for polytopic differences. Unlike Gohr's attack [Goh19a], their approach doesn't rely on neutral bits but instead filters $(r+1)$-round ciphertext pairs conforming to the initial differential using an $(r+1)$-round neural distinguisher, selecting pairs producing the highest scores.

Hou *et al.* in [HRC23]† leveraged key response profile periodicity to achieve key recovery using only a partial profile. This approach is particularly necessary for block ciphers with round key sizes significantly larger than 16 bits, such as SIMON64/128, ensuring feasible key response profile generation.

In [CSYY23], the authors proposed a data reuse strategy for distinguishers processing input sets of $n > 2$ ciphertext pairs. Their approach generates a large ciphertext set and

forms subsets where each ciphertext pair appears in a limited number of subsets while maintaining sufficient distinction between subsets. Using this strategy, they applied neural distinguishers to perform 10-rounds and 11-rounds key recovery on Speck32/64 using NASA [CSY23] and Bayesian Key Recovery, respectively.

Lv *et al.* [LJSC24a]† used super-neutral bits to decrease attack data complexity and a lookup table strategy to eliminate real-time neural distinguisher invocations, performing a practical 13-round key recovery on Speck using their novel differential-linear neural distinguishers.

### C.0.3 Bit-Level Ciphertext Analysis

Recent work on neural distinguishers [CSY23, HLF+24, LRC24, Seo24, ZWL24] offers promising approaches for reducing both computational and data requirements in key recovery attacks. These distinguishers can operate on partial ciphertext bits, suggesting that complete decryption may not be necessary for successful key recovery. For detailed explanations of approaches involving distinguishers on partial ciphertexts, see Section 5 on interpretability. Here, we focus specifically on key recovery applications.

Chen *et al.* [CSY23]† trained student distinguishers using only subsets of ciphertext bits for DES and SPECK while maintaining high accuracy. For SPECK32/64, they omitted 6 of the 32 ciphertext bits, identified through their novel Bit Sensitivity Test. These optimized distinguishers enabled subkey recovery in smaller subspaces, reducing attack complexity. The authors demonstrated a practical attack on SPECK32/64 and provided theoretical estimates for attacking 10-round DES and 14-round Speck96/96.

Li *et al.* [LRC24] extended this work by developing an ensemble of student distinguishers, each trained on distinct input differences and ciphertext bit combinations. Their key insight revealed that varying input differences cause different ciphertext bits to become critical for distinction, affecting relevant key bits. Through their novel key sensitivity test, they partitioned the subkey space into independently solvable components, enabling practical key recovery against previously resistant large-state block ciphers: 18-round SIMON128, 14-round SIMON96, 14-round SIMON64, 12-round SPECK128, 10-round SPECK96, and 9-round SPECK64.

Huang *et al.* [HLF+24] introduce a novel neural differential cryptanalysis framework that substantially mitigates computational complexity in large-state block cipher key recovery. By implementing a parallelizable multi-stage approach with strategically trained neural distinguishers, the researchers demonstrate improvements in attacking SPECK. The proposed methodology leverages partial neural distinguishers (PNDs) executed in parallel, followed by a full neural distinguisher (FND) for key selection. The partial distinguishers are trained to recover independent key bits through an innovative whitening key decryption technique. Experimental validation on 10-round SPECK64 and 10-round SPECK96 reveals computational efficiency gains. Their SPECK64 attack employed a customized ResNet architecture using multiple ciphertext pairs generated via neutral bits and an advanced staged training protocol.

Seok *et al.* [Seo24] attempted partial key recovery in the final transformation of 15-round HEIGHT, claiming to recover portions of the last round key. However, their analysis relied on an assumed differential characteristic with probability $2^{-31}$ without addressing practical implementation details, particularly regarding the use of this prepended differential characteristic without neutral bits. While they theorized about a divide-and-conquer strategy, the proposal lacked concrete implementation details.

Zhang *et al.* [ZWL24] developed neural distinguishers targeting a 2-round reduced version of AES-128, specifically analyzing pairs of bytes from the ciphertext. These byte-wise distinguishers were leveraged to mount key recovery attacks on 3-round AES-128 using a divide-and-conquer strategy where different key segments were recovered independently.

### C.0.4 Key Recovery for Related-Key and Conditional Adversaries

Conditional and related-key differential cryptanalysis enhances adversarial capabilities by slowing diffusion of differences, enabling attacks on additional rounds of ciphers. Following trends in classical cryptanalysis, this approach has extended key recovery attacks in the related-key setting for SPECK [BLYZ23] and in conditional and related-key settings for KATAN [LCLH22, LLHC23].

Lin *et al.* demonstrated practical key recovery attacks on KATAN ciphers [LCLH22]† by combining conditional and related-key differential cryptanalysis. They successfully attacked 125-round KATAN32, 106-round KATAN48, and 95-round KATAN64, while proposing parallelization of the Wrong Key Response Profile calculation to enhance efficiency.

In subsequent work, Lin *et al.* developed attacks targeting 97-round KATAN32, 82-round KATAN48, and 70-round KATAN64 [LLHC23]†. Their method integrates neural distinguishers with conditional prepended differentials that constrain specific plaintext and key bits. By identifying optimal conditions and neutral bit sets, they improved the attack effectiveness.

Bao *et al.* [BLYZ23] successfully executed a 14-round key recovery attack on SPECK32/64 in the related-key setting.

### C.0.5 Limitations of Hybrid Distinguisher Models for Key Recovery

Recent works [YK21a, WWS24, WW24b] have explored hybrid approaches combining classical differential transitions with neural distinguishers. Rather than optimizing standalone $s$-round neural distinguishers, these models are constrained to input differences matching the output difference of a classical $r$-round differential transition. This hybridization aims to achieve near-perfect distinction for $(r + s)$-encrypted ciphertexts with minimal data complexity.

A significant limitation of these hybrid approaches stems from their dependence on extended classical differential paths. This dependence precludes the use of neutral bits, making it impossible to construct the plaintext structures necessary for key recovery methods like those in [Goh19a]. While some studies, including [YK21a], have claimed breakthrough results in key recovery, these claims lack experimental validation.

In a different approach, Yadav *et al.* [YK24] constructed high-accuracy neural distinguishers from low-accuracy ones without prepending classical differentials, though at the cost of increased data complexity.