# SoK: 5 Years of Neural Differential Cryptanalysis

David Gerault<sup>1</sup>, Anna Hambitzer<sup>1</sup>, Moritz Huppert<sup>2</sup> and Stjepan Picek<sup>3</sup>

<sup>1</sup> Technology Innovation Institute, Abu Dhabi, UAE {name.lastname}@tii.ae

 $^2$  Technical University of Darmstadt, Cryptoplexity <code>moritz.huppert@tu-darmstadt.de</code>

<sup>3</sup> Radboud University, The Netherlands stjepan.picek@ru.nl

**Abstract.** At CRYPTO 2019, A. Gohr introduced Neural Differential Cryptanalysis by applying deep learning to modern block cipher cryptanalysis. Surprisingly, the resulting neural differential distinguishers enabled a new state-of-the-art key recovery complexity for 11 rounds of SPECK32. As of May 2024, according to Google Scholar, Gohr's article has been cited 178 times. The wide variety of targets, 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 178 publications and focus on the 50 that deal with differential neural distinguishers to systematically review and compare them. We then discuss two challenges for the field, namely comparability of neural distinguishers and scaling.

Keywords: Neural Differential Cryptanalysis, Systematization of Knowledge

## 1 Introduction

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 by the introduction of LeNet in 1998 [LBBH98], achieving human-level performance in digit recognition. In the game of Go, advancements in Monte Carlo Tree Search (MCTS) and reinforcement learning led to remarkable achievements, such as Google's AlphaGo surpassing human capabilities [SHM+16, SHS+18, SAH+20]. More recently, transformer-based Large Language Models (LLMs) [VSP+17], like GPT, have revolutionized natural language processing, demonstrating near-human capabilities in tasks like machine translation and language generation.

At CRYPTO'19, Gohr's seminal paper [Goh19a] was "the first to show that neural networks can be used to produce attacks quite competitive to the published state of the art against a round-reduced version of a modern block cipher". A cryptanalytic distinguishing attack discerns encrypted data from random data by detecting distinctive features within the input data, enabling the differentiation of their respective distributions. In the context of differential cryptanalysis, the objective of the distinguisher is to discriminate between ciphertext pairs  $C_0, C_1$  derived from plaintext pairs  $P_0, P_1$  with a fixed input difference ( $\delta$ ) versus those originating from a random input difference (rand). Such a distinguisher can be used to mount a key recovery attack on a block cipher: When partially decrypting an (r + 1)-round ciphertext pair using a guessed round key, the *r*-round distinguisher can differentiate between a valid key guess, which yields the characteristic distribution of the case  $\delta$ , and an invalid key guess, which yields the uniform distribution of the case rand. A higher-accuracy distinguisher enhances the effectiveness of the key recovery attack. In Neural Differential Cryptanalysis, the distinguisher is a deep neural network  $\mathcal{ND}$ . Gohr's work challenged conventional belief by demonstrating the remarkable cryptographic distinguishing capabilities of machine learning techniques. Notably, neural networks applied to SPECK32 reduced to 8 rounds exhibit superior accuracy compared to pure differential distinguishers. Additionally, Gohr achieved significant improvements in the time complexity of the 11-round key recovery attack. Gohr's success has motivated extensive follow-up research in Neural Differential Cryptanalysis over the last five years.

Figure 1 a) shows the basic scheme for a neural differential distinguisher experiment as introduced in [Goh19a].



Figure 1: a) Neural Differential Distinguisher: Basic Pipeline. Start with two plaintext  $P_0, P_1$ , where  $P_0 \oplus P_1 = \delta$  or  $P_0 \oplus P_1 =$  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.

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 differential neural distinguishers have attempted to apply the scheme to **other symmetric primitives** or to use **other neural network architectures**. Variations of the initial experiment have been proposed, for which classification was proposed at FSE 2024 [BGH<sup>+</sup>23], based on the number of input ciphertexts n, the number of used input differences m, the applied feature engineering E and the distinguishing experiment type T. A general open question for deep neural networks is the one of **explainability**, i.e., cryptographic features the neural distinguisher is actually learning. Gohr's work was analyzed in detail by Benamira *et al.* at EUROCRYPT 2021 [BGPT21], and more recently by Bao *et al.* at ASIACRYPT 2023 [BLYZ23]. Finally, we also observe some efforts into making the neural distinguisher process a more automated process [BGH<sup>+</sup>23].

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

1. Comprehensive Field Review: We have conducted an exhaustive survey of the follow-up work (Section 3). 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 Differential Neural Cryptanalysis (Section 3).

2

- 2. **Rigorous Classification and Comparison:** We systematically classified and compared the outcomes of peer-reviewed research on differential neural distinguishers (Section 6). Our analysis reveals that, over the past five years, the research community has investigated differential neural distinguishers spanning diverse symmetric primitives and neural network architectures. We also identify severe methodological issues in some peer-reviewed papers and challenge their results.
- 3. 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).
- 4. Future Challenges: We identify and discuss two major challenges set to shape the next five years of neural cryptanalysis (Section 8).

**Organization.** We start with a short historical overview of works that connect Artificial Intelligence (AI) and Cryptography before Gohr's work [Goh19a] in Section 2. We identify the full body of research and taxonomy of research directions of Neural Differential Cryptanalysis in Section 3. The comparative review of all differential neural distinguishers published to date is provided in Section 6. Based on our observations in the comparative review, we give best practice recommendations for training (and reviewing) neural distinguishers in Section 7. We formulate two future challenges for the field of Neural Differential Cryptanalysis in 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]. 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.

Klimov *et al.* analyzed the security of a key exchange protocol based on mutually learning neural networks [KMS02]. While the authors showcase it is unlikely that the attacker using a similar neural network is unlikely to converge to the same key, they showed it is possible to break the protocol in several ways. Castro *et al.* used evolutionary algorithms to construct a cryptanalytic tool that can distinguish between the two-round TEA algorithm and random permutations [CSIR02]. 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]. Tapiador *et al.* used heuristics to conduct nonlinear cryptanalysis and applied it to the MARS cipher S-box [TCHC07]. Chou *et al.* experimented with machine learning techniques to mount distinguishing attacks and concluded it is not possible to extract useful information from ciphertexts produced by modern ciphers operating in secure modes, nor to distinguish them from random data [CLC12].

On the other hand, Svenda *et al.* used evolutionary algorithms to construct empirical tests for randomness [SSUM14]. Awad and El-Alfy provided a survey of computational intelligence applications in cryptography, focusing on the automated design and cryptanalysis of ciphers [AEA17].

# 3 Neural Differential Cryptanalysis: A Taxonomy of Research Directions

## 3.1 Selected Literature

As of May 13, 2024, a total of 178 works cite Gohr's work [Goh19a] on Google Scholar. Out of these, 25 references are written in languages other than English and are not considered further. Additionally, 30 references are not peer-reviewed, being published only on platforms such as arXiv or ePrint [BBCD20, BBDH21, BBC+23, BGL+21, ElS21, GJS20, GLN22, Goh22, HRC21a, HRC21b, HRC21d, JKM20, JKM21, Jun05, KJL+22, LTJ+20, LSW+23, LRC22, PMC+22, PMK20, SM23a, SWL+24, Sug24, WNB+23, ZL20a, ZL20b, ZW22a, ZW+22b, ZZW24, ZDW+23]. After excluding these, we are left with 123 peer-reviewed references, which we systematically categorize as shown in Figure 2.



**Figure 2:** Taxonomy of the peer-reviewed publications in English that cite Gohr's work [Goh19a] to date.

We consider the following references outside the field of research on Neural Differential Cryptanalysis as they are surveys, overviews, or book chapters that treat the use of "ML in cryptography" [BHR<sup>+</sup>22, Bru21, CDS22, NR23, PJ21, PJ22, Som23, Tan23, Tu22, ZG24], or their research focuses on other topics such as: classical cryptanalysis [Bak21, BCdST<sup>+</sup>23, ELR20, FLW<sup>+</sup>23, GPT21, KS22, KY21, SLL24, WW22, YK21b, YK22], cryptanalysis of

historic or toy ciphers [GZDAL22, KSJS21, KLK<sup>+</sup>23, LMK<sup>+</sup>21, PMDC22, PKM23], deep learning-supported design of cryptographic algorithms [CS21, HLG<sup>+</sup>23, ITYY21, LTJ<sup>+</sup>21, MJBHC22], neural side-channel attacks [GJS21, TDD22, YBBP23, ZZC<sup>+</sup>22], building distinguishers between different ciphers [BPS22, DM23, MPM<sup>+</sup>21, XLC<sup>+</sup>22], neural preimage attacks [JMTD22, LLL<sup>+</sup>21, PTD22a], the introduction of a new tool or library [BGG<sup>+</sup>23, Ess23, Hal22], the proposal of a new cipher [BSL21, DCW23], neural output prediction attacks [KEI<sup>+</sup>22, KEI<sup>+</sup>23], neural integral distinguishers [HLLL, ZL22], neural attacks on protocols [TD21, ZKL20], post-quantum schemes [LWAZ<sup>+</sup>24, WCCL22], or pseudorandom number generators [Boa24, EAZD23], or other unrelated topics [KLJW23, ZZS21, PYW24, AAEK22, MGKMP21, PTD22b, HLZW20, DZF<sup>+</sup>21, AABAA22, DDK<sup>+</sup>23, RRSM22b, So20, TDF<sup>+</sup>22, MKMP21, MLYW22, Kar23].

This leaves us with a total of 49 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 [BB22, BBD<sup>+</sup>23, BGL<sup>+</sup>22, BLYZ23, BR21, BGH<sup>+</sup>23, BGPT21, BBP22, CSY23, CSYY23, DCC23, ERP22, EGP23, HGH<sup>+</sup>23, HRC21c, HRC23, KJL<sup>+</sup>23, KKJ<sup>+</sup>24, LCLH22, LLHC23, LRC23, LRCL23, LLS<sup>+</sup>24, LTZ22a, LTZ22b, MLR<sup>+</sup>23, MPKM<sup>+</sup>22, Pal, PPWR23, PSM23, RLS23, RRSM22a, SM23b, SCL24, SSL<sup>+</sup>24, SZM21, SSL<sup>+</sup>22, TTJ23, TH21, WW21, WWH21, WTZ<sup>+</sup>22, WQW<sup>+</sup>24, YK21a, YW23, ZZY<sup>+</sup>21, ZZ21, ZLWL23, ZWC23].

### 3.2 Taxonomy of Research Directions

We found contributions to the explainability (or interpretability) of neural distinguishers in the following nine works [Goh19a, BGPT21, CSY23, BBP22, DCC23, HGH<sup>+</sup>23, LRC23, YW23, BLYZ23], and will discuss their respective contributions in Section 4. We found contributions to neural-aided key recovery attacks in the following 16 works [Goh19a, BGL<sup>+</sup>21, HRC21a, TH21, ERP22, LTZ22a, BGL<sup>+</sup>22, CSY23, CSYY23, BLYZ23, KLK<sup>+</sup>23, LCLH22, LLHC23, TTJ23, YW23, ZLWL23], and will give an overview of these works in Section 5.

The majority (44/49) of peer-reviewed research on Neural Differential Cryptanalysis involves the training of differential neural distinguishers. More precisely, differential neural distinguishers are trained in [BGPT21, BB22, YK21a, CSY23, CSYY23, BR21, BGL<sup>+</sup>22, SZM21, HRC21c, BBP22, TH21, BGH<sup>+</sup>23, LLS<sup>+</sup>24, WW21, RRSM22a, ERP22, WWH21, ZLWL23, BBD<sup>+</sup>23, LTZ22a, TTJ23, KJL<sup>+</sup>23, LCLH22, HRC23, LTZ22b, BLYZ23, HGH<sup>+</sup>23, LRC23, EGP23, LRCL23, YW23, DCC23, LLHC23, WTZ<sup>+</sup>22, SSL<sup>+</sup>22, RLS23, ZWC23, ZZY<sup>+</sup>21, PSM23, MPKM<sup>+</sup>22, SSL<sup>+</sup>24, WQW<sup>+</sup>24, KKJ<sup>+</sup>24, SM23b]. We will provide a comparative review of all peer-reviewed differential neural distinguishers to date in Subsection 6.3.

#### Research Directions in Neural Differential Cryptanalysis

In the following sections, we will give a survey-like overview of the work-to-date on explainability (Section 4) and neural-aided key recovery (Section 5). The main part of our remaining work will focus on a critical, comparative review of trained differential neural distinguishers (Section 6). Based on this discussion, we will provide best practice recommendations for neural distinguisher training (Section 7) and formulate the future challenges (Section 8) for the field of Neural Differential Cryptanalysis.

# 4 Overview: Explainability of Differential Neural Distinguishers

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 feature importance 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<sup>+</sup>23] to automatically identify good input differences for neural distinguishers.

In [BBP22], Bacuieti *et al.* further investigated the structure of the neural network itself. In particular, they 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 differential neural distinguishers.

Chen *et al.* proposed a new concept named Informative Bit and a test called a Bit Sensitivity Test to identify those [CSY23]. The authors defined an informative bit as the ciphertext bit that is helpful in distinguishing between the cipher and a pseudo-random permutation. As such, this can be seen as an interpretability approach.

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 on which bits the attention mechanism focuses most, which in turn provides interpretability results.

Hambitzer *et al.* introduced deep learning ensemble (NNBits) as a tool for bit-profiling and evaluation of cryptographic (pseudo) random bit sequences [HGH<sup>+</sup>23]. From the bit-level information provided by NNBits, the authors managed to partially explain the accuracy obtained by Gohr's depth-1 neural distinguisher in round 6 for SPECK32/64.

Liu *et al.* performed an interpretability analysis of two new neural distinguishers whose results are mainly reflected in the relationship between the neural distinguishers, truncated differential, and advantage bits [LRC23]. The authors considered the interpretability of the algorithm named the advantage bits search and investigated the types of bit positions preferred by the distinguisher.

Bao *et al.* developed explicit rules to be used alongside DDTs to enhance the effectiveness of 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 show 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 by leveraging the value-dependent differential probability, it is possible to add additional knowledge to DDT-based distinguishers.

## 5 Overview: Neural Aided Key Recovery Attacks

As already stated, the turning point in the development of AI-based distinguishers was the work by A. Gohr that managed to obtain results rivaling (and surpassing) state-of-the-art results with manual cryptanalysis [Goh19a]. More precisely, Gohr obtained high accuracy for 6-round and 7-round neural distinguishers of Speck32/64 and achieved 11-round and 12-round key recovery attacks based on the neural distinguishers.

Bao *et al.* introduced the generalized neutral bits techniques and the framework of conditional differential neural cryptanalysis [BGL<sup>+</sup>21]. The authors improved the success rate of deep learning-assisted key recovery attacks, considering accuracies and the number of rounds of neural distinguishers, as well as the classical differential paths spliced in front of neural distinguishers. Moreover, they explored data complexity for deep learning-assisted key recovery attacks. Finally, they managed successful key recovery attacks on 13-round Speck32/64 and 16-round Simon32/64.

Hou *et al.* trained 9-round and 8-round differential distinguisher of SIMON32 based on deep residual neural networks. The authors managed to construct a last subkey recovery attack on 11-Round SIMON32 with practical data and time complexities [HRC21a].

Tian and Hu trained a residual neural network to get the 7-, 8-, and 9-round neural distinguishers for SIMON32/64 [TH21]. Moreover, they performed a distinguishing attack and key-recovery attack against 15-round SIMON32/64.

Yadav and Kumar used an r-round classical differential distinguisher to build an (r + s)-round ML-based differential distinguisher, where s denotes the rounds covered by the ML-based distinguisher [YK21a]. The authors demonstrated a reduction in data complexities of distinguishers for 9-round SPECK32, 12-round SIMON32, and 8-round GIFT64.

Ebrahimi *et al.* presented a Partial Differential (PD) ML-distinguisher and demonstrated its effectiveness on lightweight cipher SPECK32/64 [ERP22]. With this approach, the authors managed to train a partial ML distinguisher over 8-bits, which is almost as accurate as an equivalent ML distinguisher over the entire 32 bits for six rounds of SPECK32/64.

Lyu *et al.* improved upon Gohr's framework and applied it to Simeck32/64 [LTZ22a]. The authors obtained 8/9/10-round neural differential distinguishers for Simeck32/64 and recovered the penultimate round and last round subkeys for 13/14/15-round Simeck32/64 with low data complexity and time complexity.

Bao *et al.* developed ML-based key-recovery attacks on more extended round-reduced Speck32/64 [BGL+22]. With their approach, the authors achieved an improved 12-round and the first practical 13-round attacks. Moreover, the authors provided the first practical 16-round-based key-recovery attack on Simon32/64.

Chen *et al.* proposed a Neural-Aided Statistical Attack (NASA) running experiments on round reduced Speck32/64, DES, and Speck96/96 [CSY23]. With this approach, the authors break 10-round DES while Gohr's attack breaks 8-round DES.

Chen *et al.* proposed a neural distinguisher that considers multiple ciphertext pairs simultaneously and builds multiple ciphertext pairs from different keys [CSYY23]. The authors consider five round-reduced ciphers: Speck32/64, Chaskey, PRESENT, DES, and SHA3-256, and show reduced data complexity of the attack.

Bao *et al.* provided explicit rules to be used alongside DDTs to enhance the effectiveness of distinguishers compared to pure DDT-based distinguishers [BLYZ23]. The authors successfully conducted a 14-round key recovery attack on Speck32/64.

Kim *et al.* considered lightweight block ciphers (S-DES, S-AES, and S-SPECK) and showed a reduction in the number of parameters required for training and an increase in the average of bit accuracy probability compared with previous state-of-the-art work [KLK<sup>+</sup>23].

Lin *et al.* presented practical key recovery attacks on KATAN ciphers [LCLH22]. More precisely, the authors provided practical key recovery attacks on the 125-round KATAN32, 106-round KATAN48, and 95-round KATAN64.

Lin *et al.* proposed a conditional differential analysis framework that is based on deep learning with the multi-differential neural distinguishers [LLHC23]. The authors presented practical key recovery attacks on the 97-round KATAN32, 82-round KATAN48, and 70-round KATAN64.

Teng *et al.* constructed neural distinguishers for the LBC-IoT and SLIM block ciphers [TTJ23]. The authors showed a practical-time key recovery attack on LBC-IoT for up to 8 rounds.

Yue and Wu showed a better accuracy of the 7-round differential neural distinguisher for Speck32/64 compared to the one from A. Gohr [YW23]. The authors demonstrated a key recovery attack on 8 rounds of Speck32/64.

Zhang *et al.* provided practical key recovery attacks on SIMECK32/64, improved the 15-round attack, and launched the first practical 16-, 17-round key recovery attacks for SIMECK32/64 [ZLWL23].

# 6 Comparative Review: Differential Neural Distinguishers

In the following, we provide a comparative review of all trained differential neural distinguishers to date. First, all investigated neural network architectures to date are reviewed (Subsection 6.1), then we detail the classification scheme (Subsection 6.2) and finalize with a comparative review of the best differential neural distinguishers for each symmetric primitive (Subsection 6.3).

### 6.1 Architectures

In deep learning, various specialized layers and mechanisms<sup>1</sup> optimize neural networks. To date, a number of different neural network architectures have been employed in Neural Differential Cryptanalysis.

 $\mathcal{ND}_{Gohr}$   $\mathcal{ND}_{Gohr}$  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 two different possible depths (depth-1, 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.  $\mathcal{ND}_{Gohr}$  has subsequently been used on ten primitives (CHASKEY, DES, KATAN, LBCIoT, PRESENT, SIMECK, SIMON, SLIM, SPECK, SHA3) by [BGPT21, HRC21c, SZM21, TH21, WW21, BB22, LCLH22, LTZ22b, WTZ<sup>+</sup>22, CSY23, CSYY23, EGP23, HGH<sup>+</sup>23, LLHC23, LRCL23, RLS23, TTJ23, ZWC23, ZLWL23, WQW<sup>+</sup>24]. A non-peer-reviewed work by Gohr, Leander, and Neumann [GLN22] provides a thorough investigation of relevant hyperparameters when adapting  $\mathcal{ND}_{Gohr}$  to a new primitive. Variants of Gohr's original network have been created:  $\mathcal{ND}_{Gohr}^{pruned}$  is a pruned version of  $\mathcal{ND}_{Gohr}$  for SPECK introduced in [BBP22].  $\mathcal{ND}_{Gohr}^{attntn.}$  was introduced in [DCC23] and adds an attention mechanism to  $\mathcal{ND}_{Gohr}$  and applies it to SPECK. [HGH<sup>+</sup>23] uses an ensemble of  $\mathcal{ND}_{Gohr}$  ( $\mathcal{ND}_{Gohr}^{ensmbl.}$ ) 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  $(\mathcal{ND}_{Gohr}^{sep.conv})$  was introduced in [LRC23] and applied to SPECK with the motivation to save training cost.

Convolutional Neural Networks without residual connections (CNNs) have been investigated in [BB22], and it was concluded that "CNNs are not suitable for the purpose of finding a distinguisher". This is contradicted in [BR21]. Moreover, [WWH21] builds CNN-based neural differential distinguishers for PRESENT and SPECK. [MPKM<sup>+</sup>22] compares CNNs and MLPs on GIFT and PRIDE, where MLPs give better accuracies. **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<sup>+</sup>23] as a "cipher-agnostic" neural network that aims to avoid SPECK-dedicated features of  $\mathcal{ND}_{Gohr}$ . 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  $\mathcal{ND}_{Gohr}$ . Notably, using a simple staged<sup>2</sup> training pipeline, and a simple additional polishing step, the same accuracy as Gohr is obtained for SPECK. It has been employed in [BGH<sup>+</sup>23]

<sup>&</sup>lt;sup>1</sup>We give a short introduction to the needed vocabulary to understand the various architectures used for Neural Differential Cryptanalysis: MLPs employ densely connected layers in which every neuron in layer iis connected to all neurons in layer i - 1. This leads to a large number of parameters. A convolutional layer, the basis for CNNs (Convolutional Neural Networks), uses filters to scan the input, requiring more computations (FLOPs or MACs) than densely connected layers but excelling in capturing spatial hierarchies. The *inception module* enhances feature capture by using multiple parallel convolutional operations with different kernel sizes, concatenating the outputs for richer feature extraction. Residual connections allow information to bypass layers in RESNets (Residual Neural Networks), preventing blockages and enhancing information flow during training. LSTM is a type of RNN (recurrent neural network) that captures long-term dependencies in sequential data with memory cells and gating mechanisms and is effective in tasks like time-series prediction. Attention mechanisms are the basis for modern transformer neural networks, dynamically focusing on important input parts and improving tasks like translation or generative AI.

<sup>&</sup>lt;sup>2</sup>Staged training refers to the method to continue training the best r-1 round neural differential distinguisher in round r.

to automatically generate distinguishers for seven primitives (SPECK, SIMON, HIGHT, PRESENT, KATAN, TEA and XTEA, LEA).

**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 many works [BR21, YK21a, ZZY<sup>+</sup>21, BB22, ERP22, RRSM22a, SSL<sup>+</sup>22, BBD<sup>+</sup>23, PSM23, KKJ<sup>+</sup>24, SSL<sup>+</sup>24] on diverse primitives (ASCON, FF1 and FF3, GIFT, GIMLI, KNOT, PRESENT, SIMECK, SIMON, SPECK, TEA and XTEA, TinyJAMBU) to investigate their potential as a differential neural distinguisher.

Long-short term memory cells (LSTMs) were used in [BB22, SSL<sup>+</sup>22] on GIMLI, TinyJAMBU, and GIFT. [BB22] investigated CNNs in comparison to *Long Short-Term Memory Networks* (LSTMs) and MLPs on the GIMLI-PERMUTATION and found that MLPs perform best. In contrast, [SSL<sup>+</sup>22] compared LSTMs and MLPs on TinyJAMBU and GIFT and obtained better neural distinguishers using LSTMs.

**Inception** In the Inception architecture (INC), a layer inspired by GoogLeNet's Inception module replaces at least one of the (convolutional) layers of the original  $\mathcal{ND}_{Gohr}$  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.

[ZWC23, ZLWL23] construct the INC architecture by replacing the initial convolutional block of  $\mathcal{ND}_{Gohr}$  and obtain neural distinguishes for SIMECK, PRESENT, CHASKEY, and DES. [YW23] construct INC by replacing the convolutional layers in the residual blocks of  $\mathcal{ND}_{Gohr}$  and apply this architecture to SPECK.

[BLYZ23] introduces the idea of staged training together with a partially frozen network (INC<sup>freeze</sup>). 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).

**Others.** In [KJL<sup>+23</sup>], 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<sup>+22</sup>]. SENet introduces a new building block for CNN that improves the finding of channel interdependencies at almost no computational cost. [BGL<sup>+22</sup>] applied SENet to SPECK and SIMON. SE-ResNet was first used as differential neural distinguisher by [LLS<sup>+24</sup>], motivated by "the success of  $\mathcal{ND}_{Gohr}$  on SPECK [Goh19a] and SENet on SIMON [BGL<sup>+22</sup>]". [LLS<sup>+24</sup>] 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. Gohr's analysis was performed within the secret key chosen-plaintext attack (SK / CPA) model. We do not consider the following neural distinguisher that was trained under a different attack model: [PPWR23] uses generative AI trained in an adaptively chosen ciphertext or known key scenario to distinguish 10-round SPECK32/64. We report Classical ML results, such as SVM in [BBD<sup>+</sup>23], on the rare occasion that they are competitive with neural distinguishers.

### 6.2 Classification Scheme *n*-*m*-*T*-*E* for Neural Distinguishers

The abundance of different settings used to train neural distinguishers sometimes makes it difficult to compare two results. In  $[BGH^+23]$ , the authors propose a classification of

neural distinguishers based on their distinctive features, labeled n, m, T, and E. We use that classification in our review.

### 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 with the same labels during the key recovery process, effectively 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 propose 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.,  $C_0 \oplus C_1 || C_2 \oplus C_3 \dots$  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. They also propose the data reuse strategy, in which the number of multi-pair samples created from a set of pairs is increased through picking with replacement. In [ZWC23], the authors raise 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 conclude 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. Finally, in e.g., [SSL+22], the authors consider polytopic samples with multiple input differences, where the used plaintexts are  $(P, P \oplus \delta_0, P \oplus \delta_1 \dots)$ , effectively building k relevant pairs from k+1 plaintexts. A similar technique is referred to as mixture differential in  $[WQW^+24]$ .

### 6.2.2 Number of input differences: m

Baksi *et al.* [BB22] explore 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 differential neural network distinguisher. In this model, multiple differences are used, keeping one plaintext fixed among the differences and changing the other. In [WTZ<sup>+</sup>22], propose a multiple input difference scheme called ND<sub>am</sub>, 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.

### 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 differential neural cryptanalysis is to replace the ciphertext pairs (T = CT) by their XOR difference ( $T = \delta$ ). This approach, used by works such as Baksi *et al.* [BB22], Zezhou *et al.* [HRC21c], and Yadav *et al.* [YK21a], simplifies the training process, at the cost of losing some information.

Advanced types of feature engineering (T = 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]. 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. In [YW23], the authors use 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]  $(E = \mathbf{R})$ , 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  $(E = \mathbf{R}_M)$ , 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 [BB22]'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  $i \ (E = D)$ .

In [BR21], the samples are built using modular addition difference, rather than XOR, to analyze the ciphers TEA and RAIDEN ( $E = \mathbb{R}^+$ ).

In [EGP23], the samples are built through rotational-XOR differences rather than XOR, which we denote by  $E = \mathbb{R}^+$ .

### 6.3 Comparative Review

Based on the full body of research in Neural Differential Cryptanalysis (Subsection 3.1), 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 comparative review for all symmetric primitives (ASCON, CHASKEY, DES, FF1 and FF3, GIFT, GIMLI, HIGHT, KATAN, KNOT, LBCIOT, LEA, PRESENT, PRIDE, SHA3, SIMECK, SIMON, SKINNY, SLIM, SPECK, TEA and XTEA, TinyJAMBU) can be found in the appendix. Here, we show several examples.

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 present it as well. ii) In most works, no error margins on the results are provided, preventing us from displaying them. Ideally, the accuracies shown should be test accuracies on sets of several fresh samples. However, in many works, only the validation accuracy is reported. iii) Some works, such aa [YK21a], consider the concatenation of an r-round probabilistic differential distinguisher and an s-round neural distinguisher. In

this table, we focus solely on the neural distinguisher part. 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*.

#### 6.3.1 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 initialisation, 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<sup>+</sup>24] train neural differential distinguishers for the 4-round ASCON-PERMUTATION with an accuracy of 0.5069 in the standard setting (2-1- $\delta$ -R), and can 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 1:** Overview of the Differential Neural Distinguishers for ASCON.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
ASCON-PERM	MLP	$2\text{-}2\text{-}\delta\text{-}D$	1.1M	1.1M		3	0.9861	
	Classical ML	$2-2-\delta-D$	$64 \mathrm{K} \ (2 \cdot 2^{14.96})$	$16 \mathrm{K} \ (2 \cdot 2^{12.96})$		3	1	$[BBD^+23]$
	MLP	$2\text{-}1\text{-}\delta\text{-}R$	20M	20M	-	4	0.5069	$[SSL^{+}24]$
ASCON (Rate)	Classical ML	2-2-δ-D	$64 \mathrm{K} \ (2 \cdot 2^{14.96})$	$16K (2 \cdot 2^{12.96})$	-	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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

### 6.3.2 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  $[ZZY^+21]^{\dagger}$ , the authors claim a distinguisher on 7 rounds because the training accuracy is 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  $[MPKM^+22]^{\dagger}$ , the authors claim a full round distinguisher on GIFT-64 with over 90% accuracy, using 2<sup>20</sup> 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 claim a full round distinguisher on PRIDE with 100% accuracy. Full-round attacks on modern and reputable ciphers are an extraordinary claim and require extraordinary evidence, which the author's manuscript does not provide.

In [RRSM22a], only 10K samples are used for training and testing; as a result, the distinguishers in Table 5 exhibit significant overfitting (e.g., 92% training accuracy and 25% testing accuracy for M1 on 6 rounds).

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GIFT-64	MLP	$2\text{-}1\text{-}\delta\text{-}R$	$67M (2 \cdot 2^{25})$	67M	-	4	0.65	[YK21a]
	LSTM	3-2-CT-R	$17M(2 \cdot 2^{23})$	$4M (2 \cdot 2^{21})$		6	0.5754	$[SSL^+22]$
	MLP	$3-2-\delta-R$	2.2M	0.5M		FULL	0.96	[MPKM <sup>+</sup> 22] <sup>†</sup>
GIFT-128	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	200 K	-	7	0.5002	$[ZZY^+21]^{\dagger}$
GIFT-COFB	MLP	2-4-δ-D	20K	20K	_	4	0.615	[RRSM22a]

**Table 2:** Overview of the Differential Neural Distinguishers for GIFT.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

 $^\dagger$  A critical discussion of these results is provided in the text.

### 6.3.3 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<sup>+</sup>23] reaches statistically significant accuracies up to 69 rounds in an automatically generated distinguisher, and note 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] reaches 51 rounds in the standard setting, and 59 when using 64 pairs. In [LLHC23, LCLH22], the authors prepend a conditional probability 1 differential on r rounds (which holds based on conditions on the equality of some plaintext and key bits) to an s rounds neural distinguisher. For these distinguishers, we write (r + s) to highlight which part is purely neural. 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 are obtained for 125, 106 and 95 rounds respectively, in the related key scenario. Single-key conditional neural distinguishers are also mentioned in [LCLH22] for 85, 72 and 61 rounds respectively, but the r + s decomposition is not given so we omit them in the table.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
KATAN32	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	51	0.533	[LCLH22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$64\text{-}1\text{-}\delta\text{-}R$	64M	$6.4 \mathrm{M}$	_	$84^{*}$ (26+58)	0.602	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$128 \text{-} 1 \text{-} \delta \text{-} R$	1280M	128M	_	59	0.575	[LCLH22]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	69	0.505	$[BGH^+23]$
KATAN32 <sup>RK</sup>	$\mathcal{ND}_{\mathrm{Gohr}}$	$128\text{-}1\text{-}\delta\text{-}\mathrm{R}$	1280M	128M	-	$112^* (66+46)$	0.647	[LCLH22]
KATAN48	$\mathcal{ND}_{\mathrm{Gohr}}$	$64\text{-}1\text{-}\delta\text{-}\mathrm{R}$	64M	$6.4 \mathrm{M}$	_	$72^{*}(25+47)$	0.5820	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	40	0.58	[LCLH22]
	$\mathcal{ND}_{ ext{Gohr}}$	$96-1-\delta-R$	960M	96M	-	50	0.54	[LCLH22]
KATAN48 <sup>RK</sup>	$\mathcal{ND}_{\mathrm{Gohr}}$	$48\text{-}1\text{-}\delta\text{-}\mathrm{R}$	960M	96M	-	$96^{*}(57+39)$	0.625	[LCLH22]
KATAN64	$\mathcal{ND}_{ ext{Gohr}}$	$64\text{-}1\text{-}\delta\text{-}\mathrm{R}$	64M	$6.4\mathrm{M}$	_	$61^{*}(25+26)$	0.6130	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	31	0.718	[LCLH22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	1280M	128M	-	36	0.548	LCLH22
$\rm KATAN64^{RK}$	$\mathcal{ND}_{\mathrm{Gohr}}$	$128\text{-}1\text{-}\delta\text{-}\mathrm{R}$	1280M	128M	-	$86^*$ (54+32)	0.728	[LCLH22]

**Table 3:** Overview of the Differential Neural Distinguishers for KATAN.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

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

(r+s) means an s-round neural distinguisher with r are prepended rounds.

#### 6.3.4 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<sup>+</sup>23], a 9-round distinguisher with an accuracy of 0.5092 is given, which favorably compares to the 7-round distinguishers of [CSYY23], despite [CSYY23] using 4 pairs per sample; on the other hand, [ZW22a] obtains a slightly higher accuracy, at the cost of using 32 ciphertexts per samples. In comparison, the best differential characteristic for PRESENT reduced to 9 rounds has probability  $2^{-36}$  [Wan07].

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
PRESENT	CNN	2-1-CT-R	20M	2M	-	7	0.533	[WWH21]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.563	[GLN22]
	$\mathcal{ND}_{ ext{Gohr}}$	8-1-CT-R	20M	2M	-	7	0.5853	CSYY23
	CNN	$2-2-\delta-D$	20M	2M	-	8	0.515	[WWH21]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.512	$[BGH^+23]$
	INC	32-1-CT-R	320M	32M	-	8	0.5416	[ZWC23]
PRESENT-64/80 <sup>RK</sup>	MLP	$6-1-\delta-R$	$6.3 M^{*}$	$1.6 M^{*}$	-	5	0.614	[PSM23]
	CNN	2-2-δ-D	20M	2M	-	10	0.517	[WWH21]

<sup>RK</sup> Related key setting.

 $^{*}$  [PSM23] uses 2<sup>20</sup> samples, each composed of 3 pairs, i.e., 6.3M ciphertexts for training, and one quarter as many pairs, i.e., 1.6M ciphertexts for validation.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-). The work [GLN22] of Gohr, Leander, and Neumann was not peer-reviewed.

### 6.3.5 SPECK

SPECK is a family of ARX block ciphers, denoted SPECK-B/K, that encrypt blocks of size B with a key of size K. SPECK-32/64, SPECK-48/96, SPECK-64/128, SPECK-96/96, and SPECK-128/256 have 22, 23, 27, 29, and 34 rounds, respectively. Neural differential distinguishers have been built for a versions of SPECK.

For SPECK-32, the best accuracies are reported when multiple the number of ciphertexts n is increased, as is done in [BBD<sup>+</sup>23] (n = 2), [HRC21c, CSYY23] (n = 64), [LRCL23] (n = 128). Currently, the best accuracy of 93.9% in round 8 of SPECK-32 is obtained by [CSYY23] when using n = 64. In the standard setting (2-1-CT-R) [BGH<sup>+</sup>23] reach the same accuracy as [Goh19b] with an automated pipeline that is not dedicated to SPECK<sup>3</sup>. In terms of larger state experiments, two automated pipelines reach 7, respectively 8 rounds of SPECK-64 [WW21, BGH<sup>+</sup>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] obtains the first 7-round distinguisher, while for SPECK-128, [BGH<sup>+</sup>23] obtains the first 10-round neural distinguisher in an automated pipeline.

<sup>&</sup>lt;sup>3</sup>We note that [BLYZ23] states that "the simple training pipeline [of [BGH+23]] did not produce NDs 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 polishing step is indeed needed to achieve the same accuracy, the polishing step is a *highly simplified* version of the 8-round training scheme used by Gohr (in conclusion, it does *not* demand more time or data).

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SPECK-32	MLP	$2-1-\delta$ -R	209M	105M	-	3	0.79	[YK21a]
	Quantum	2-1-CT-R	/	2K	-	5	0.53	[KJL+23] <sup>†</sup>
	CNN	2-2-δ-D	20M	2M		5	0.959	[WWH21]
	$\mathcal{ND}_{\mathrm{Cohr}}^{\mathrm{ensmbl.}}$	2-1-CT-R	20M	2M	-	6	0.781	$[HGH^{+}23]$
	MLP	$2-1-\delta-R$	20M	2M	-	6	0.72	[ERP22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	100-1-A-R	20M	2M		6	1	[BGPT21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{pruned}}$	2-1-CT-R	20M	2M	-	7	0.596	$[BBP22]^{\dagger}$
	DenseNet	$2-1-\delta-R$	2M	2M	-	7	0.531	[SM23b] <sup>†</sup>
	CNN	2-2-δ-D	20M	2M		7	0.599	[WWH21]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.614	[WW21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{attntn.}}$	2-1-CT-R	20M	2M	-	7	0.6169	[DCC23]
	CNN	2-1-CT-R	20M	2M	-	7	0.618	[WWH21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{attntn.}}$	16-1-CT-R	160M	16M		7	0.728	[DCC23]
	$\mathcal{ND}_{Gohr}^{sep.conv.}$	8-1-CT-R	80M	8M		7	0.6939	[LRC23]
	INC	64-1-PD-R	64M	$6.4\mathrm{M}$		7	0.9713	[YW23]
	MLP	2-2-ð-D	21M	12M		8	0.51413	$[BBD^+23]$
	INCIreeze	2-1-CT-R	20M	2M	-	8	0.5135	[BLYZ23]
	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	-	8	0.514	Goh19b
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.514	$ BGH^+23 $
	MLP	2-2-0-D	14M	8M		8	0.515	[BBD <sup>+</sup> 23]
	$\mathcal{N}\mathcal{D}_{Gohr}$	04-1-0-K 199 1 MDMCD D	04IVI 199M	19.01/		0.041/1	0.304	[HRU21C]
	$\mathcal{ND}_{Gohr}$	64.1 CT R	128W 20M	212.81		8	0.0002	CSVV23
SPECK-32 RK	CNN	2_2_δ_D	20M	2M		7	0.559	[US1123] [WWH21]
51 LOR-52	CNN	2-1-CT-B	20M	2M	-	7	0.576	[WWH21]
	INC <sup>freeze</sup>	2-1-CT-R	20M	2M	-	10	0.5562	[BLYZ23]
SPECK-48	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	√	7	0.726	[WW21]
	$\mathcal{ND}_{\mathrm{Gohr}}$	96-1-MRMSD-R	96M	$9.6\mathrm{M}$		8	0.5462	[LRCL23]
SPECK-64	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.632	[WW21]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.537	$[BGH^+23]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	128M	1.28M		8	0.632	[HRC21c]
	$\mathcal{ND}_{\mathrm{Gohr}}$	128-1-MRMSD-R	128M	$12.8\mathrm{M}$	-	8	0.7181	[LRCL23]
SPECK-96	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	-	7	$0.850^{\ddagger}$	[CSY23]
SPECK-128	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	10	0.593	$[BGH^+23]$

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

<sup>RK</sup> Related key setting. / means unknown.

<sup>‡</sup> 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 on the authors' repository; we give the average of 10 runs, each with  $10^6$  samples.

<sup>†</sup> [KJL<sup>+</sup>23]report an accuracy of 53% (round 5) on only 1,000 validation samples. The experimental mean or standard deviation is not given. For a binomial experiment on 1k samples, the statistically expected standard deviation 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]report an accuracy of 53.1% (round 7) on 2M training, respectively validation samples, and provide a comparison in which DenseNet outperforms  $\mathcal{ND}_{Gohr}$ . At such a small number of training samples, both networks show heavy overfitting ([SM23b, Table 2]), and the authors themselves call the result only "marginal".

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

#### 6.3.6 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. For the case of SIMON, some authors experimented with a vast amount of data: [HRC21c] uses  $2^{25}$  (33.6M) pairs for training, and [BGL<sup>+</sup>22] obtain an 11-round distinguisher for SIMON32 at the cost of staged trained in two steps, with respectively  $2^{28}$  (268M) and  $2^{30}$  (1074M) pairs. In [BGH<sup>+</sup>23], the authors propose a polishing step, retraining a neural distinguisher initially trained with  $10^7$  pairs with an additional  $10^9$  pairs. In [LLS<sup>+</sup>24], Lu *et al.* use advanced feature engineering and 320M ciphertexts ( $2 \cdot 10^7$  samples, each composed of 8 pairs), and reach 12 rounds of SIMON32 in the single-key scenario. In the related key scenario, the same authors reach 13 rounds, whereas [EGP23] only reaches 11 rounds with a rotational XOR distinguisher.

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SIMON-32/64	MLP	$2-1-\delta-R$	67M	8M	-	5	0.570	[YK21a]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	9	0.5907	HRC21c
	$\mathcal{ND}_{Gohr}$	2-1-CT-R	20M	2M		9	0.6263	[SZM21]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	/	/	-	9	0.6320	[TH21]
	$\mathcal{ND}_{\mathrm{Gohr}}$	4-3-CT-R	40M	$4\mathrm{M}$		9	0.6373	SZM21
	$\mathcal{ND}_{ ext{Gohr}}$	4-3-CT-R	40M	$4\mathrm{M}$		9	0.923	$[WQW^+24]$
	$\mathcal{ND}_{ ext{Gohr}}$	$64-1-\delta-R$	640M	6.4M			0.6109	[HRC21c]
	SENet	2-1-A-R	2684M	268.435 M		11	0.517	$[BGL^+22]$
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	11	0.518	$[BGH^+23]$
	$\mathcal{ND}_{ ext{Gohr}}$	64-1-MRMSD-R	640M	64M		11	0.6081	[LRCL23]
	SE-ResNet	16-1-A-R	320M	32M		12	0.514	$[LLS^+24]$
$SIMON-32/64^{RK}$	$\mathcal{ND}_{ ext{Gohr}}$	$2-1-CT-R^{RX}$	20M	2M		11	0.5445	[EGP23]
1	SE-ResNet	16-1-A-R	320M	32M		13	0.5262	$[LLS^+24]$
SIMON-48/96	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5789	[HRC21c]
	$\mathcal{ND}_{ ext{Gohr}}$	$96-1-\delta-R$	960M	9.6M		11	0.6143	HRC21c
	$\mathcal{ND}_{\mathrm{Gohr}}$	96-1-MRMSD-R	960M	96M		12	0.6159	[LRCL23]
SIMON-64/128	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	11	0.59.72	[HRC21c]
,	$\mathcal{ND}_{ ext{Gohr}}$	$128-1-\delta-R$	1280M	12.8 M		12	0.6957	HRC21c
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	13	0.518	$[BGH^+23]$
	$\mathcal{ND}_{ ext{Gohr}}$	128-1-MRMSD-R	1280M	128M		13	0.701	LRCL23
	SE-ResNet	16-1-A-R	320M	32M		14	0.519	$[LLS^{+}24]$
SIMON-64/128 <sup><math>RK</math></sup>	$\mathcal{ND}_{\mathrm{Gohr}}$	$2-1-CT-R^{RX}$	20M	2M		13	0.5151	[EGP23]
1	SE-ResNet	16-1-A-R	160M	16M		14	0.5788	$[LLS^{+}24]$
SIMON-128/256	DBitNet	2-1-CT-R	20M	2M	√	20	0.507	$[BGH^+23]$
$SIMON-128/256^{RK}$	$\mathcal{ND}_{Cohr}$	$2-1-CT-R^{RX}$	20M	2M			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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-). / means unknown.

## 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 training outcome is heavily influenced by the chosen hyperparameters and neural network architectures.

To interpret the success of neural network training correctly, it is important to carefully distinguish between training, validation, and test data. 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 performance of the model on previously unseen data during training, *validation data* is used. To accurately measure the model's performance on unseen data, it is crucial never to train it on the validation data. 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 data 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, now the validation data has been used in model optimization, and can no longer be used to characterize the performance on previously unseen data. For the final characterization, fresh *testing data* should be used instead. For example, Gohr first trains on  $10^7$  samples, and after each epoch, the training success is validated on  $10^6$  validation data is tested on several fresh sets of  $10^6$  test samples. The result is a neural distinguisher with minimal overfitting and a robust accuracy including error margins, e.g.,  $0.788 \pm 8.17 \cdot 10^{-4}$  [Goh19a, Table 2], on previously unseen test data.

The number of parameters of a deep neural network *does not* relate to its computational training cost in a straightforward way. 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. 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.

#### Summary of Best Practices for Neural Distinguisher Training

- 1. **Results Reporting I:** Clearly indicate the results obtained on training, validation, and test datasets, as well as the size of each dataset.
- 2. **Results Reporting II:** Denote accuracy (or any other metrics) with error margins on multiple sets of *freshly generated* testing 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.

## 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. In order 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** In a broad sense, a problem can be defined as an n - M - T - E configuration, a primitive, a training pipeline, and a dataset size. A first step would be to

run all the models on the initial SPECK32 problem in the 2-1-CT-R setting to gain a better overview of which architectures perform best when given the same conditions and training data. However, changing the input format of architecture may not be trivial [BGH<sup>+</sup>23], so an effort would be required from developers to provide instructions on running their model with a generic input size. From there, problems with various input sizes and cardinality could be compared.

While neural distinguishers' learning features from multiple pairs have not been demonstrated yet [GLN22, CSYY23], feature engineering techniques have been shown to have a strong impact on the distinguisher's performance [LLS<sup>+</sup>24]. Such features include partial decryption or mixing ciphertext values and difference-related features; standardizing them and building the corresponding benchmarking instances would provide valuable insight.

In addition, the training regime of a neural distinguisher has been known to be critical from the start: in [Goh19a], an advanced training pipeline is presented as required to reach 8 rounds. In this pipeline, a neural distinguisher is pre-trained on the most likely difference after some rounds and then re-trained for the whole number of rounds with 100 times more samples than the other distinguishers. Further research often uses a similar polishing step on the final round to improve accuracy. There is, therefore, a distinction between raw performance, where the neural distinguisher is trained from scratch for each round in the same conditions, and enhanced accuracy, where techniques such as pretraining [Goh19b], freezing layers [GLN22], retraining of the previous rounds distinguishers [BR21], or increased number of samples in the last rounds may be used. For these reasons, the design of a standard, generic pipeline to compare enhanced distinguishers would be beneficial.

In addition to training pipelines, the number of samples is another choice to make. A lot of the literature follows the reasonable choices of [Goh19a] (10<sup>7</sup> samples for training, 10<sup>6</sup> for testing), as lowering these numbers seems to have a dramatic effect on performance. However, neural distinguishers trained on multiple-pair samples (e.g., [BBCD20]) are less easy to compare. On the one hand, one can choose to fix the number of samples; in this case, the neural distinguisher will see more pairs than one with single-pair samples, which provides an unfair advantage. On the other hand, one can fix the number of pairs; in this case, the resulting distinguisher may have an unfair disadvantage, as it is trained on fewer samples. An extreme example would be samples composed of 10<sup>7</sup> pairs; in that case, the second distinguisher is trained on a single sample and has little chance of learning anything. While some works use over 1 billion ciphertexts during training, little has been studied on using that order of magnitude of data in the 2-1-CT-R scenario (as compared to multiple pairs per sample), and it would be interesting to include this axis in a benchmarking study.

**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, and true negative rate. More recently  $[BGH^+23]$ , the number of floating point operations (FLOPS) of a neural network was added as an additional metric. It relates to the complexity of evaluating inputs to the neural network, which itself has an impact on the training time and complexity of the corresponding key recovery attack. What is missing is a metric that relates the accuracy of the neural distinguisher with its performance, which is related to accuracy per flop. This metric should say something about the final complexity of the key recovery and answer questions such as: *Does doubling the FLOPS to gain 1% accuracy improve the attacks?*. Furthermore, an important notion in key recovery attacks is the *wrong key response profile*, which relates the confidence of the distinguisher with the error in the decryption of the last round with the wrong key. It has not been studied how it correlates with other metrics of a neural distinguisher may become slightly better for key recovery due to a better wrong key response profile. Finally, it is

not clear how to weigh the ability of a distinguisher to be extended by probabilistic rounds to reach a longer key recovery.

## 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 when 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.



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

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 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 The Benchmarking Challenge, 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<sup>4</sup> to allow AI researchers and engineers to apply state-of-the-art methods to Neural Differential Cryptanalysis.

## 9 Conclusion

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 that of comparing neural distinguisher results and that of scaling up to much larger and ambitious architecture.

Over the past 5 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 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.
- [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.
- [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.
- [BB22] Anubhab Baksi and Anubhab Baksi. Machine learning-assisted differential distinguishers for lightweight ciphers. *Classical and Physical Security of Symmetric Key Cryptographic Algorithms*, pages 141–162, 2022.

<sup>&</sup>lt;sup>4</sup>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<sup>+</sup>15].

- [BBC<sup>+</sup>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.
- [BBD<sup>+</sup>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<sup>+</sup>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.
- [BGG<sup>+</sup>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<sup>+</sup>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<sup>+</sup>21] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Conditional differential-neural cryptanalysis. IACR Cryptol. ePrint Arch., 2021:719, 2021.
- [BGL<sup>+</sup>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<sup>+</sup>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.
- [Boa24] 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.
- [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.
- [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.
- [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.
- [CLC12] Jung-Wei Chou, Shou-De Lin, and Chen-Mou Cheng. On the effectiveness of using state-of-the-art machine learning techniques to launch cryptographic distinguishing attacks. In *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<sup>+</sup>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.
- [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.
- [DZF<sup>+</sup>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€<sup>TM</sup>Arco. On pseudorandomness and deep learning: A case study. Applied Sciences, 13:3372, 2023.
- [EGP23] Amirhossein Ebrahimi, David Gerault, and Paolo Palmieri. Deep learningbased 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.

- [FLW<sup>+</sup>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.
- [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.
- [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 differentialbased distinguishers and forgeries for ascon. *IACR Transactions on Symmetric Cryptology*, 2021.
- [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<sup>+</sup>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<sup>+</sup>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€<sup>TM</sup> Track at the RSA Conference, pages 493–523. Springer, 2023.
- [HLG<sup>+</sup>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<sup>+</sup>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.
- [HLZW20] Botao Hou, Yongqiang Li, Haoyue Zhao, and Bin Wu. Linear attack on roundreduced 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 roundreduced 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 roundreduced simon based on deep learning. *The Computer Journal*, 66:2517–2534, 2023.
- [HSB<sup>+</sup>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.
- [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.
- [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.
- [KEI<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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.
- [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.
- [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. Gradientbased 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.
- [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<sup>+</sup>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.
- [LLS<sup>+</sup>24] Jinyu Lu, Guoqiang Liu, Bing Sun, Chao Li, and Li Liu. Improved (relatedkey) differential-based neural distinguishers for simon and simeck block ciphers. *The Computer Journal*, 67(2):537–547, 2024.
- [LMK<sup>+</sup>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.
- [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<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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.
- [MPKM<sup>+</sup>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<sup>+</sup>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.
- [NR23] Abderrahmane Nitaj and Tajjeeddine Rachidi. Applications of neural network-based ai in cryptography. *Cryptography*, 7:39, 2023.
- [Pal] Paolo Palmieri. Deep learning-based rotational-xor distinguishers for and-rx block ciphers: Evaluations on simeck and simon.
- [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.
- [PMC<sup>+</sup>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.
- [PPWR23] Raphaël C-W Phan, Arghya Pal, KokSheik Wong, and Sailaja Rajanala. CηιDAE: 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.
- [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.

- [RDS<sup>+</sup>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.
- [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.
- [SAH<sup>+</sup>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.
- [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.
- [SHM<sup>+</sup>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<sup>+</sup>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<sup>+</sup>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. *Cryp*tology 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.

- [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.
- [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<sup>+</sup>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.
- [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.
- [Sug24] Nobuyuki Sugio. Implementation of cryptanalytic programs using chatgpt. Cryptology ePrint Archive, 2024.
- [SWL<sup>+</sup>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. Nonlinear 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<sup>+</sup>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.
- [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<sup>+</sup>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.
- [WNB<sup>+</sup>23] Ping Wang, Shishir Nagaraja, Aurélien Bourquard, Haichang Gao, and Jeff Yan. Sok: Acoustic side channels. *arXiv preprint arXiv:2308.03806*, 2023.
- [WQW<sup>+</sup>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<sup>+</sup>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.
- [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.
- [XLC<sup>+</sup>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.
- [YW23] Xiaoteng Yue and Wanqing Wu. Improved neural differential distinguisher model for lightweight cipher speck. *Applied Sciences*, 13:6994, 2023.
- [ZDW<sup>+</sup>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 resourceconstrained environments: A comprehensive survey. *Future Generation Computer Systems*, 2024.
- [ZKL20] Behnam Zahednejad, Lishan Ke, and Jing Li. A novel machine learningbased 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.
- [ZLWL23] Liu Zhang, Jinyu Lu, Zilong Wang, and Chao Li. Improved differentialneural cryptanalysis for round-reduced simeck32/64. Frontiers of Computer Science, 17(6):176817, 2023.
- [ZW22a] Liu Zhang and Zilong Wang. Improving differential-neural distinguisher model for des, chaskey, and present. *arXiv preprint arXiv:2204.06341*, 2022.
- [ZW<sup>+</sup>22b] Liu Zhang, Zilong Wang, et al. Improving differential-neural cryptanalysis with inception. *Cryptology ePrint Archive*, 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.

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

- [ZZC<sup>+</sup>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<sup>+</sup>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 Differential Neural Distinguishers

## A.1 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 initialisation, 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<sup>+</sup>24] train neural differential distinguishers for the 4-round ASCON-PERMUTATION with an accuracy of 0.5069 in the standard setting (2-1- $\delta$ -R), and can 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 7:** Overview of the Differential Neural Distinguishers for ASCON.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
ASCON-PERM	MLP	$2\text{-}2\text{-}\delta\text{-}D$	1.1M	1.1M		3	0.9861	
	Classical ML	$2-2-\delta-D$	$64 \mathrm{K} \ (2 \cdot 2^{14.96})$	$16 \mathrm{K} \ (2 \cdot 2^{12.96})$		3	1	$[BBD^+23]$
	MLP	$2\text{-}1\text{-}\delta\text{-}R$	20M	20M	-	4	0.5069	$[SSL^{+}24]$
ASCON (Rate)	Classical ML	$2\text{-}2\text{-}\delta\text{-}\mathrm{D}$	$64 \mathrm{K} \left( 2 \cdot 2^{14.96} \right)$	$16K (2 \cdot 2^{12.96})$	-	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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.2 CHASKEY

CHASKEY is a 128-bit ARX-based permutation on 8 rounds.

In [CSYY23], the best distinguisher uses 16 pairs per sample, though the authors present a valid single-pair distinguisher for CHASKEY as well.

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
CHASKEY	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	$17M (2 \cdot 2^{23})$	$40 \mathrm{K} \ (2 \cdot 2^{14.3})$	-	4	0.6161	[BB22]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M `	2M	-	4	0.6161	[CSYY23]
	$\mathcal{ND}_{\mathrm{Gohr}}$	32-1-CT-R	320M	32M		4	0.7712	[CSYY23]
	INC	16-1-CT-R	20M	10M		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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.3 DES

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

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
DES	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	5	0.58	[CSY23]
	INC	4-1-CT-R	40M	4M	Ν	6	0.5653	[CSYY23]
	$\mathcal{ND}_{ ext{Gohr}}$	32-1-CT-R	320M	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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.4 FF1 and FF3

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

In [KKJ<sup>+</sup>24], the authors perform neural cryptanalysis of FF1 and FF3, for digits (FFX-D) and lowercase letters (FFX-L). We report the best results in the 2-1-CT-R setting but note that the authors additionally performed experiments in the m-2-CT-D setting, with similar, yet not directly comparable, results. The number of samples for training and testing is not given, nor the source code (/-entries in Table 10).

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
FF1-D	MLP	2-1-CT-R	/	/	-	10	0.85	$[KKJ^+24]$
FF1-L	MLP	2-1-CT-R	/	1	-	2	0.522	$[KKJ^+24]$
FF3-D	MLP	2-1-CT-R	/	/	-	8	0.98	[KKJ <sup>+</sup> 24]
FF3-L	MLP	2-1-CT-R	/	1	-	2	0.55	$[KKJ^+24]$

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-). / means unknown.

## A.5 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  $[ZZY^+21]^{\dagger}$ , the authors claim a distinguisher on 7 rounds because the training accuracy is 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  $[MPKM^+22]^{\dagger}$ , the authors claim a full round distinguisher on GIFT-64 with over 90% accuracy, using 2<sup>20</sup> 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 claim a full round distinguisher on PRIDE with 100% accuracy. Full-round attacks on modern and reputable ciphers are an extraordinary claim and require extraordinary evidence, which the author's manuscript does not provide.

In [RRSM22a], only 10K samples are used for training and testing; as a result, the distinguishers in Table 5 exhibit significant overfitting (e.g., 92% training accuracy and 25% testing accuracy for M1 on 6 rounds).

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GIFT-64	MLP	$2\text{-}1\text{-}\delta\text{-}R$	$67M (2 \cdot 2^{25})$	67M	-	4	0.65	[YK21a]
	LSTM	3-2-CT-R	$17M (2 \cdot 2^{23})$	$4M (2 \cdot 2^{21})$		6	0.5754	$[SSL^+22]$
	MLP	$3-2-\delta-R$	2.2M	0.5M		FULL	0.96	[MPKM <sup>+</sup> 22] <sup>†</sup>
GIFT-128	MLP	$2\text{-}1\text{-}\delta\text{-}R$	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] <sup>†</sup>
GIFT-COFB	MLP	2-4-δ-D	20K	20K	-	4	0.615	[RRSM22a]

**Table 11:** Overview of the Differential Neural Distinguishers for GIFT.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

 $^\dagger$  A critical discussion of these results is provided in the text.

### A.6 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.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
GIMLI	MLP DBitNet	$2-2-\delta$ -D 2-1-CT-R	0.4M 20M	$\begin{array}{c} 0.04\mathrm{M} \\ \mathrm{2M} \end{array}$	- ✓	8 11	$\begin{array}{c} 0.510\\ \textbf{0.524} \end{array}$	$\substack{[\mathrm{BB22}]\\[\mathrm{BGH}^+23]}$
GIMLI-HASH	MLP	2-2-δ-D	$0.4\mathrm{M}$	$0.04 \mathrm{M}$	-	8	0.5219	[BB22]
GIMLI-CIPHER	MLP	2-2-δ-D	0.4M	$0.04 \mathrm{M}$	_	8	0.5099	[BB22]

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

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

### A.7 HIGHT

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
HIGHT	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	10	0.751	$[BGH^+23]$
HIGHT <sup>RK</sup>	$\operatorname{DBitNet}$	2-1-CT-R	20M	2M	$\checkmark$	14	0.563	$[BGH^+23]$

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

<sup>RK</sup> Related key setting.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

### A.8 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<sup>+</sup>23] reaches statistically significant accuracies up to 69 rounds in an automatically generated distinguisher, and note 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] reaches 51 rounds in the standard setting, and 59 when using 64 pairs. In [LLHC23, LCLH22], the authors prepend a conditional probability 1 differential on r rounds (which holds based on conditions on the equality of some plaintext and key bits) to an s rounds neural distinguisher. For these distinguishers, we write (r + s) to highlight which part is purely neural. 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 are obtained for 125, 106 and 95 rounds respectively, in the related key scenario. Single-key conditional neural distinguishers are also mentioned in [LCLH22] for 85, 72 and 61 rounds respectively, but the r + s decomposition is not given so we omit them in the table.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
KATAN32	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	51	0.533	[LCLH22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$64\text{-}1\text{-}\delta\text{-}R$	64M	$6.4 \mathrm{M}$	-	$84^*$ (26+58)	0.602	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$128-1-\delta-R$	1280M	128M	-	59	0.575	[LCLH22]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	69	0.505	$[BGH^+23]$
KATAN32 <sup>RK</sup>	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	1280M	128M	-	$112^* (66+46)$	0.647	[LCLH22]
KATAN48	$\mathcal{ND}_{ ext{Gohr}}$	$64 \text{-} 1 \text{-} \delta \text{-} R$	64M	$6.4\mathrm{M}$	_	$72^{*}(25+47)$	0.5820	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	40	0.58	[LCLH22]
	$\mathcal{ND}_{ ext{Gohr}}$	$96-1-\delta-R$	960M	96M	-	50	0.54	LCLH22
$\rm KATAN48^{RK}$	$\mathcal{ND}_{\mathrm{Gohr}}$	$48\text{-}1\text{-}\delta\text{-}\mathrm{R}$	960M	96M	-	$96^{*}(57+39)$	0.625	[LCLH22]
KATAN64	$\mathcal{ND}_{ ext{Gohr}}$	$64 \text{-} 1 \text{-} \delta \text{-} R$	64M	$6.4 \mathrm{M}$	_	$61^{*}(25+26)$	0.6130	[LLHC23]
	$\mathcal{ND}_{ ext{Gohr}}$	$2\text{-}1\text{-}\delta\text{-}R$	20M	2M	-	31	0.718	[LCLH22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	1280M	128M	-	36	0.548	LCLH22
$\rm KATAN64^{RK}$	$\mathcal{ND}_{\mathrm{Gohr}}$	$128\text{-}1\text{-}\delta\text{-}\mathrm{R}$	1280M	128M	-	$86^*$ (54+32)	0.728	[LCLH22]

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

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

 $^{\rm \scriptscriptstyle RK}$  Related key setting.

\* (r+s) means an s-round neural distinguisher with r are prepended rounds.

### A.9 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 [BB22], the authors use a neural distinguisher to recognize whether a 1 difference is introduced in the first or the second byte.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
KNOT-256	MLP	$2\text{-}2\text{-}\delta\text{-}D$	$1.048 \mathrm{M}$	$1.048 \mathrm{M}$	-	10	0.5912	[BB22]
KNOT-512	MLP	$2\text{-}2\text{-}\delta\text{-}\mathrm{D}$	$1.048 \mathrm{M}$	$1.048\mathrm{M}$	-	12	0.6032	[BB22]

**Table 15:** Overview of the Differential Neural Distinguishers for KNOT.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

### A.10 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.

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
LEA-128	$\operatorname{DBitNet}$	2-1-CT-R	20M	2M	$\checkmark$	11	0.512	$[BGH^+23]$
Class: n-	m- $T$ - $E$ , from	n Subsection (	6.2. Und	er this c	onvention, Go	ohr's initial	experimen	ts are 2-1-CT-
R, and the	results obtai	ned in greyed	d out set	tings $n$ -	m- $T$ - $E$ are n	ot directly	comparab	le. AutoND:
• 1• • • • • • • • • • • • • • • • • •	.1 11	• • • • •					1, 0	

indicates if the neural distinguisher was automatically generated ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.11 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 17:** Overview of the Differential Neural Distinguishers for LBCIoT.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
LBC-IoT	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	2M	$200 \mathrm{K}$	-	7	0.607	[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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.12 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<sup>+</sup>23], a 9-round distinguisher with an accuracy of 0.5092 is given, which favorably compares to the 7-round distinguishers of [CSYY23], despite [CSYY23] using 4 pairs per sample; on the other hand, [ZW22a] obtains a slightly higher accuracy, at the cost of using 32 ciphertexts per samples. In comparison, the best differential characteristic for PRESENT reduced to 9 rounds has probability  $2^{-36}$  [Wan07].

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
PRESENT	CNN	2-1-CT-R	20M	2M	-	7	0.533	[WWH21]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.563	[GLN22]
	$\mathcal{ND}_{ ext{Gohr}}$	8-1-CT-R	20M	2M	-	7	0.5853	[CSYY23]
	CNN	$2-2-\delta-D$	20M	2M	_	8	0.515	[WWH21]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.512	$[BGH^+23]$
	INC	32-1-CT-R	320M	32M	-	8	0.5416	[ZWC23]
PRESENT-64/80 <sup>RK</sup>	MLP	$6\text{-}1\text{-}\delta\text{-}R$	$6.3 \mathrm{M}^*$	$1.6 M^*$	-	5	0.614	[PSM23]
,	CNN	$2-2-\delta-D$	20M	2M	_	10	0.517	[WWH21]

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

Related key setting.

<sup>\*</sup> [PSM23] uses 2<sup>20</sup> samples, each composed of 3 pairs, i.e., 6.3M ciphertexts for training, and one quarter as many pairs, i.e., 1.6M ciphertexts for validation.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-). The work [GLN22] of Gohr, Leander, and Neumann was not peer-reviewed.

## A.13 PRIDE

PRIDE is a 20-round SPN cipher using 64-bit blocks and 128-bit keys. In [MPKM<sup>+</sup>22], the authors claim 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, especially considering the lack of evidence provided in the paper.

**Table 19:** Overview of the Differential Neural Distinguishers for PRIDE.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
	MLP	$2\text{-}1\text{-}\delta\text{-}R$	$734 \mathrm{K}$	$157 \mathrm{K}$	-	20	1	[MPKM <sup>+</sup> 22] <sup>†</sup>

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

 $^\dagger$  A critical discussion of these results is provided in the text.

## A.14 SHA3

\_

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

Table 20: Overview of the Differential Neural Distinguishers for SHA3.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SHA3-256	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	-	3	0.7228	[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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.15 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 use an inception-based architecture, and m samples per pair (between 1 and 256), with 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 training is done on  $2 \cdot 10^7$  samples. Their best distinguisher reaches 13 rounds of SIMECK32.

In [WTZ<sup>+</sup>22], the authors investigate 2 variations of a multiple input differences scenario, where the samples are the concatenations of pairs with differences  $\delta_i$ . In ND<sub>rm</sub>, a sample is the concatenation of a pair of ciphertexts for each difference (resulting in n = 2m); in ND<sub>am</sub>, 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 are trained on  $2^{24}$  (16.8M) samples, 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.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SIMECK-32	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-D	20M	2M	-	10	0.5407	[LTZ22b]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5438	[LTZ22b]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$4-3-ND_{am}-R$	67M	1M		11	0.5042	$[WTZ^{+}22]$
	INC	16-1-A-R	320M	32M		12	0.5161	[ZLWL23]
	INC	512-1-A-R	1024M	124M		13	0.5086	[ZLWL23]
SIMECK-32 <sup>(Unkeyed)</sup>	MLP	2-2-δ-D	66k	66k		9	0.526	[BBD+23]
SIMECK-32/64 <sup>RK</sup>	SE-ResNet	16-1-A-R	160M	16M		15	0.5467	$[LLS^+24]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	$2\text{-}1\text{-}\text{CT-}\text{R}^{RX}$	20M	2M		15	0.5134	[EGP23]
SIMECK-48/96 <sup>RK</sup>	$\mathcal{ND}_{\mathrm{Gohr}}$	$2\text{-}1\text{-}\text{CT-}\text{R}^{RX}$	20M	2M	_	17	0.5206	[EGP23]
SIMECK-64 <sup>(Unkeyed)</sup>	MLP	2-2-δ-D	33k	33k	-	14	0.55	[BBD+23]
SIMECK-64/128 <sup>RK</sup>	SE-ResNet	16-1-A-R	160M	16M		22	0.5180	$[LLS^{+}24]$
,	$\mathcal{ND}_{\mathrm{Gohr}}$	$2\text{-}1\text{-}\text{CT-}\text{R}^{RX}$	20M	2M		20	0.5212	[EGP23]

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

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

### A.16 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. For the case of SIMON, some authors experimented with a vast amount of data: [HRC21c] uses  $2^{25}$  (33.6M) pairs for training, and [BGL<sup>+</sup>22] obtain an 11-round distinguisher for SIMON32 at the cost of staged trained in two steps, with respectively  $2^{28}$  (268M) and  $2^{30}$  (1074M) pairs. In [BGH<sup>+</sup>23], the authors propose a polishing step, retraining a neural distinguisher initially trained with  $10^7$  pairs with an additional  $10^9$  pairs. In [LLS<sup>+</sup>24], Lu *et al.* use advanced feature engineering and 320M ciphertexts ( $2 \cdot 10^7$  samples, each composed of 8 pairs), and reach 12 rounds of SIMON32 in the single-key scenario. In the related key scenario, the same authors reach 13 rounds, whereas [EGP23] only reaches 11 rounds with a rotational XOR distinguisher.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SIMON-32/64	MLP	$2-1-\delta-R$	67M	8M	-	5	0.570	[YK21a]
,	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	9	0.5907	HRC21c
	${\cal ND}_{ m Gohr}$	2-1-CT-R	20M	2M		9	0.6263	[SZM21]
	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	/	/	-	9	0.6320	[TH21]
	$\mathcal{ND}_{\mathrm{Gohr}}$	4-3-CT-R	40M	4M		9	0.6373	
	$\mathcal{ND}_{\mathrm{Gohr}}$	4-3-CT-R	40M	4M		9	0.923	$[WQW^+24]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	$64-1-\delta-R$	640M	6.4M			0.6109	
	SENet	2-1-A-R	2684M	268.435 M		11	0.517	$[BGL^+22]$
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	11	0.518	$[BGH^+23]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	64-1-MRMSD-R	640M	64M		11	0.6081	[LRCL23]
	SE-ResNet	16-1-A-R	320M	32M		12	0.514	$[LLS^+24]$
$SIMON-32/64^{RK}$	$\mathcal{ND}_{\mathrm{Gohr}}$	$2-1-CT-R^{RX}$	20M	2M		11	0.5445	[EGP23]
	SE-ResNet	16-1-A-R	320M	32M	-	13	0.5262	$[LLS^{+}24]$
SIMON-48/96	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	10	0.5789	[HRC21c]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$96-1-\delta-R$	960M	9.6M		11	0.6143	[HRC21c]
	$\mathcal{ND}_{ ext{Gohr}}$	96-1-MRMSD-R	960M	96M	-	12	0.6159	[LRCL23]
SIMON-64/128	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	11	0.59.72	[HRC21c]
	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	1280M	12.8M		12		[HRC21c]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	13	0.518	$[BGH^+23]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	128-1-MRMSD-R	1280M	128M		13	0.701	LRCL23
	SE-ResNet	16-1-A-R	320M	32M		14	0.519	$[LLS^+24]$
$SIMON-64/128^{RK}$	$\mathcal{ND}_{ ext{Gohr}}$	$2-1-CT-R^{RX}$	20M	2M		13	0.5151	[EGP23]
	SE-ResNet	16-1-A-R	160M	16M	-	14	0.5788	$[LLS^{+}24]$
SIMON-128/256	DBitNet	2-1-CT-R	20M	2M	√	20	0.507	$[BGH^+23]$
$SIMON-128/256^{RK}$	$\mathcal{ND}_{\mathrm{Gohr}}$	$2\text{-}1\text{-}\text{CT-}\text{R}^{RX}$	20M	2M		16	0.5062	[EGP23]

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

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-). / means unknown.

## A.17 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<sup>+</sup>23], the authors reach 6 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 23: Overview of the Differential Neural Distinguishers for SKINNY.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SKINNY128 <sup>(Unkeyed)</sup>	Classical ML	2-2-δ-D	2M	$1.2\mathrm{M}$	-	6	0.546	[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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.18 SLIM

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

In [RLS23], the authors perform 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  $[TTJ23]^{\dagger}$ , 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.

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SLIM	$\mathcal{ND}_{ ext{Gohr}} \ \mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R 2-1-CT-R	2M 2M	200K 2M	-	/ 5	$\begin{array}{c} 0.5036\\ 0.814\end{array}$	$[\mathrm{TTJ23}]^{\dagger}$ [RLS23]

**Table 24:** Overview of the Differential Neural Distinguishers for SLIM.

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

<sup>†</sup> A critical discussion of these results is provided in the text.

## A.19 SPECK

SPECK is a family of ARX block ciphers, denoted SPECK-B/K, that encrypt blocks of size B with a key of size K. SPECK-32/64, SPECK-48/96, SPECK-64/128, SPECK-96/96, and SPECK-128/256 have 22, 23, 27, 29, and 34 rounds, respectively. Neural differential distinguishers have been built for a versions of SPECK.

For SPECK-32, the best accuracies are reported when multiple the number of ciphertexts n is increased, as is done in [BBD<sup>+</sup>23] (n = 2), [HRC21c, CSYY23] (n = 64), [LRCL23] (n = 128). Currently, the best accuracy of 93.9% in round 8 of SPECK-32 is obtained by [CSYY23] when using n = 64. In the standard setting (2-1-CT-R) [BGH<sup>+</sup>23] reach the same accuracy as [Goh19b] with an automated pipeline that is not dedicated to SPECK<sup>5</sup>. In terms of larger state experiments, two automated pipelines reach 7, respectively 8 rounds of SPECK-64 [WW21, BGH<sup>+</sup>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] obtains the first 7-round distinguisher, while for SPECK-128, [BGH<sup>+</sup>23] obtains the first 10-round neural distinguisher in an automated pipeline.

<sup>&</sup>lt;sup>5</sup>We note that [BLYZ23] states that "the simple training pipeline [of [BGH+23]] did not produce NDs 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 polishing step is indeed needed to achieve the same accuracy, the polishing step is a *highly simplified* version of the 8-round training scheme used by Gohr (in conclusion, it does *not* demand more time or data).

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
SPECK-32	ECK-32 MLP 2-1-δ-R		209M	105M	-	3	0.79	[YK21a]
	Quantum	2-1-CT-R	/	2K	-	5	0.53	[KJL+23] <sup>†</sup>
	CNN	2-2-δ-D	20M	2M		5	0.959	[WWH21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{ensmbl.}}$	2-1-CT-R	20M	2M	-	6	0.781	$[HGH^+23]$
	MLP	$2-1-\delta-R$	20M	2M	-	6	0.72	[ERP22]
	$\mathcal{ND}_{\mathrm{Gohr}}$	100-1-A-R	20M	2M		6	1	[BGPT21]
	$\mathcal{ND}_{Cohr}^{\mathrm{pruned}}$	2-1-CT-R	20M	2M	-	7	0.596	$[BBP22]^{\dagger}$
	DenseNet	$2-1-\delta-R$	2M	2M	-	7	0.531	[SM23b] <sup>†</sup>
	CNN	2-2-δ-D	20M	2M		7	0.599	[WWH21]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.614	[WW21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{attntn.}}$	2-1-CT-R	20M	2M	-	7	0.6169	[DCC23]
	CNN	2-1-CT-R	20M	2M	-	7	0.618	[WWH21]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{attntn.}}$	16-1-CT-R	160M	16M		7	0.728	[DCC23]
	$\mathcal{ND}_{\mathrm{Gohr}}^{\mathrm{septconv.}}$	8-1-CT-R	80M	8M		7	0.6939	[LRC23]
	INC	64-1-PD-R	64M	6.4M		7	0.9713	[YW23]
	MLP	$2-2-\delta-D$	21M	12M		8	0.51413	$[BBD^{+}23]$
	INC <sup>freeze</sup>	2-1-CT-R	20M	2M	-	8	0.5135	[BLYZ23]
	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	-	8	0.514	[Goh19b]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.514	$[BGH^+23]$
	MLP	$2-2-\delta-D$	14M	8M		8	0.515	$[BBD^+23]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	64-1-∂-R	64M	/		0.64M	0.564	[HRC21c]
	$\mathcal{ND}_{Gohr}$	128-1-MRMSD-R	128M	12.8M		8	0.6502	LRCL23
CDD CIV as PK	$\mathcal{ND}_{\text{Gohr}}$	64-1-CT-R	20M	2M		8	0.939	[CSYY23]
SPECK-32 <sup>rex</sup>	CNN	2-2-0-D	20M	2M		7	0.559	WWH21
	CININ	2-1-CT-R	20M	2M	-	1	0.576	[WWH21]
	INCheeze	2-1-CT-R	20M	2M	-	10	0.5562	[BLYZ23]
SPECK-48	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.726	[WW21]
	$\mathcal{ND}_{ ext{Gohr}}$	96-1-MRMSD-R	96M	9.6M	-	8	0.5462	[LRCL23]
SPECK-64	$\mathcal{ND}_{ ext{Gohr}}$	2-1-CT-R	20M	2M	$\checkmark$	7	0.632	[WW21]
	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	8	0.537	$[BGH^+23]$
	$\mathcal{ND}_{\mathrm{Gohr}}$	$128-1-\delta-R$	128M	1.28M		8	0.632	[HRC21c]
	$\mathcal{ND}_{\mathrm{Gohr}}$	128-1-MRMSD-R	128M	$12.8\mathrm{M}$	-	8	0.7181	[LRCL23]
SPECK-96	$\mathcal{ND}_{\mathrm{Gohr}}$	2-1-CT-R	20M	2M	-	7	$0.850^{\ddagger}$	[CSY23]
SPECK-128	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	10	0.593	$[BGH^+23]$

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

<sup>RK</sup> Related key setting. / means unknown.

<sup>‡</sup> 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 on the authors' repository; we give the average of 10 runs, each with  $10^6$  samples.

<sup>†</sup> [KJL<sup>+</sup>23]report an accuracy of 53% (round 5) on only 1,000 validation samples. The experimental mean or standard deviation is not given. For a binomial experiment on 1k samples, the statistically expected standard deviation 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]report an accuracy of 53.1% (round 7) on 2M training, respectively validation samples, and provide a comparison in which DenseNet outperforms  $\mathcal{ND}_{Gohr}$ . At such a small number of training samples, both networks show heavy overfitting ([SM23b, Table 2]), and the authors themselves call the result only "marginal".

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

### A.20 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 consider modular addition-based differentials, where the input difference is injected by modular addition, which we denote by  $R^+$  as the experiment. [BGH<sup>+</sup>23] automatically finds distinguishers for both TEA and XTEA for 5 cycles (10 rounds), respectively, with accuracies 0.5634 and 0.5984; the authors note that they interestingly share the same input difference. For TEA, [BGH<sup>+</sup>23] reaches two more rounds than [BR21].

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
TEA	MLP	$2\text{-}1\text{-}CT\text{-}R^+$	2M	10k	-	8	0.545	[BR21]
	$\operatorname{DBitNet}$	2-1-CT-R	20M	2M	$\checkmark$	10	0.563	$[BGH^+23]$
XTEA	DBitNet	2-1-CT-R	20M	2M	$\checkmark$	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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

## A.21 TinyJAMBU

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

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

Primitive	Arch.	Class	Trn.	Val.	AutoND	Rounds	Acc.	Ref.
TinyJAMBU-128	MLP	$2\text{-}1\text{-}\delta\text{-}R$	2.097 M $(2\cdot 2^{20})$	262K $(2\cdot 2^{17})$	-	FULL	0.9958	$[SSL^+22]^\dagger$

**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 ( $\checkmark$ ) or is the result of an elaborate, manually designed training procedure (-).

<sup>†</sup> A critical discussion of these results is provided in the text.

In  $[SSL^+22]^{\dagger}$ , the authors claim 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<sup>6</sup>: 'When nonce is reused, an attacker is able to decrypt the ciphertext since the encryption of TinyJAMBU is somehow similar to the Cipher Feedback mode.'

 $<sup>^{6} \</sup>tt https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf$