Hindawi
Security and Communication Networks
Volume 2021, Article ID 9288229, 11 pages
https://doi.org/10.1155/2021/9288229

Research Article

Improve Neural Distinguishers of SIMON and SPECK

ZeZhou Hou, JiongJiong Ren, and ShaoZhen Chen

Information Engineering University, Zhengzhou, Henan 450000, China

Correspondence should be addressed to JiongJiong Ren; jiongjiong_fun@163.com

Received 8 August 2021; Revised 26 October 2021; Accepted 30 November 2021; Published 31 December 2021

Academic Editor: AnMin Fu

Deep learning has played an important role in many fields, which shows significant potential for cryptanalysis. Although these existing works opened a new direction of machine learning aided cryptanalysis, there is still a research gap that researchers are eager to fill. How to further improve neural distinguishers? In this paper, we propose a new algorithm and model to improve neural distinguishers in terms of accuracy and the number of rounds. First, we design an algorithm based on SAT to improve neural distinguishers. With the help of SAT/SMT solver, we obtain new effective neural distinguishers of SIMON using the input differences of high-probability differential characteristics. Second, we propose a new neural distinguisher model using multiple output differences. Inspired by the existing works and data augmentation in deep learning, we use the output differences to exploit more derived features and train neural distinguishers, by splicing output differences into a matrix as a sample. Based on the new model, we construct neural distinguishers of SIMON and SPECK with round and accuracy promotion. Utilizing our neural distinguishers, we can distinguish reduced-round SIMON or SPECK from pseudorandom permutation better.

1. Introduction

Deep learning has brought about significant improvement in many fields [1–3], and it enlightened cryptanalysis. As early as 1991, Ronald Rivest [4] discussed the similarities and differences between machine learning and cryptography and analysed the application of machine learning in the field of cryptography. In recent years, deep learning has also been applied to side channel analysis [5, 6], and it was pointed out that the sensitive information on embedded devices can be effectively extracted by training neural networks.

At Crypto 2019, Gohr [7] showed that deep learning can produce very powerful cryptographic distinguishers and indicated that the neural distinguisher was better than the distinguisher obtained by traditional approach. He used an input difference to train neural distinguishers of SPECK32/64 [8] based on the deep residual neural networks (ResNets) [9]. If the accuracy of a neural distinguisher exceeds 0.5, the neural distinguisher can distinguish target cipher E from pseudorandom permutation. Gohr's work is a giant leap in differential cryptanalysis based on deep learning. However, his work actually opened many questions.

Why are neural distinguishers effective? How to improve neural distinguishers in terms of accuracy and the number of rounds?

In Eurocrypt 2021, Benamira et al. [10] proposed a detailed analysis and thorough explanations of the inherent workings of Gohr's distinguishers. They showed that Gohr's neural distinguisher was in fact inherently building a very good approximation of the differential distribution table (DDT). Based on this, Benamira et al. also constructed an 8-round distinguisher of SIMON32/64. In [10], Benamira et al. answered the first question. Similarly, Chen and Yu [11] bridged machine learning and cryptanalysis via the extended differential-linear connectivity table. The first question is answered in [10, 11]. In addition to these works related to the inherent workings of neural distinguishers, there are some works related to the improvement of the neural distinguishers. In [12], Chen and Yu designed a new neural distinguisher model using multiple ciphertext pairs instead of single ciphertext pair. The new neural distinguisher can be used to improve the key recovery attack on 11-round SPECK32/64. But Chen et al. did not explore improving the accuracy from the perspective of input difference or output difference, which is not conducive to finding a longer-round

neural distinguisher. In [13], Su et al. constructed polytopic neural distinguisher of round-reduced SIMON32/64. Their work partially answered the second question, yet the second question is still worth studying, especially in selecting the input differences and data format. Not limited to the neural distinguishers, there are also some works related to the neural aided key recovery attack [14–16].

It is not difficult to find that further improvement of the neural distinguishers is still worth studying, especially in accuracy and the number of rounds, because if the distinguishing accuracy is promoted, the complexity of key search can be reduced; and if the number of rounds is increased, the key recovery attack can be improved. However, unfortunately, there are few works to explore how to improve neural distinguishers from the perspective of the input difference. Besides, the neural distinguishers can be improved by using other distinguisher models. Inspired by these existing works, our core target is to answer the second question, that is, to further improve neural distinguishers in terms of accuracy and the number of rounds.

In this paper, our contributions are as follows.

An algorithm is designed based on SAT to improve neural distinguishers and apply to SIMON. In [7], Gohr chose $(0x40, 0x0)$ as the input difference to train his distinguisher because it transitioned deterministically to the low-weight output difference. But such input differences are hard to find, which makes it difficult to find effective distinguishers. To solve this problem, we propose an algorithm based on SAT to improve neural distinguishers. With the help of this automatic search tool, we search for the exact nr-round differential characteristics with probability $[2^{-(n/4)} \times P_{\max}, P_{\max}]$ and choose their input differences to train nr-round neural distinguishers, where $P_{\max}$ is the optimal probability and n is the block size. Utilizing the algorithm, we obtain some neural distinguishers of 9-round SIMON32/64, 10-round SIMON48/96, and 11-round SIMON64/128 with the accuracy exceeding 57% for the first time. Compared with the choice of input difference presented in [10], our algorithm obtains higher-accuracy neural distinguishers. Our results are shown in Table 1.

A new neural distinguisher model is proposed using multiple output differences and neural distinguishers of SIMON and SPECK are improved. In image recognition based on deep learning, a deep learning researcher will enhance some objective features of pictures so that the neural network can learn more effective features, which will improve the accuracy of the network. In [10], Benamira et al. explored the connection between Gohr's distinguisher and DDT, which enlightens us that the output difference is helpful to improve neural distinguishers. This also implies that we can selectively enhance certain features from output difference to improve neural distinguishers. Unlike [7, 12] using ciphertext pairs as training data, we use the output differences to train neural distinguishers by splicing output differences into a matrix as a sample. For a matrix, we treat it as an image and each output difference of the matrix is treated as an objective feature. Our goal is not only to learn each objective feature but also to learn the connections between output differences. If all output differences of the matrix are from the same input difference, the matrix will be labeled 1; otherwise, it will be labeled 0. Thanks to the new model learning more features than using ciphertext pairs, we improve neural distinguishers of SIMON32/64, SIMON48/96, and SIMON64/128. Besides, we obtain new neural distinguishers of 8-round SPECK32/64, 7-round SPECK48/96, and 8-round SPECK64/128, which are better than the existing neural distinguishers. Using our improved neural distinguishers, we can distinguish reduced-round SIMON or SPECK from pseudorandom permutation better. As a footnote, we show with experiments where the improvement in the accuracy of distinguishers is not due to the increase in the number of plaintexts but learning more features from the relationship between the output differences. The summary of our neural distinguishers together with other neural distinguishers is shown in Table 1.

SIMON $2n/mn$	SIMON acting on $2n$-bit plaintext blocks and using a mn-bit key
SPECK $2n/mn$	SPECK acting on $2n$-bit plaintext blocks and using a mn-bit key
$\oplus$	Bitwise XOR
$\odot$	Bitwise AND
$\vee$	Bitwise OR
$+$	Addition modulo 2^n
S^j	Left circular shift by j bits
K	Master key
k_i	i-round subkey $k_i = k_i^{n-1} \ldots k_i^0$

The remainder of this paper is organised as follows. In Section 2, we introduce the basic notations and review Gohr's distinguishers. In Section 3, we design an algorithm based on SAT to help us find high-accuracy neural distinguishers. In Section 4, we propose a new neural distinguisher model to ulteriorly improve neural distinguishers. Conclusions are drawn in Section 5 where we also suggest further work.

2. Preliminaries

To make it easier to read this paper, we first list the main notations. Then an overview of Gohr's work is given.

2.1. Notations

2.2. Overview of Gohr's Distinguisher Mode. Given a fixed input difference $\triangle = (0x40, 0x0)$ and a plaintext pair (P^0, P^1), the resulting ciphertext pair (C^0, C^1) is regarded as s sample. Each sample will be attached a label Y:

$$Y = \begin{cases} 1, & \text{if } P^0 \oplus P^1 = \Delta, \\ 0, & \text{else.} \end{cases} \tag{1}$$

A neural network is trained over enough samples labeled 1 and 0. In addition, half the training data comes from ciphertext pairs labeled 1 and the other half from ciphertext pairs labeled 0. For the samples with label 1, their ciphertext pairs are from a specific distribution related to the fixed input difference. For the samples with label 0, their

Table 1: Comparison of neural differential distinguishers.

Block cipher	Source of neural distinguisher/input difference	Round	Accuracy (%)
SIMON32/64	Reference [10][1]	8	82.2
	Reference [17][2]	9	59.07
	Reference [13][3]	9	63.73
	Section 3	9	59.77
	Section 4	9	82.27
	Section 4	10	61.09
SIMON48/96	Reference [17][2]	9	50.22
	Reference [10][1]	10	53.49
	Section 3	10	57.89
	Section 4	10	81.40
	Section 4	11	61.43
SIMON64/128	Reference [17][2]	10	58.61
	Section 3	11	59.72
	Section 4	11	73.79
	Section 4	12	69.57
SPECK32/64	Reference [7]	7	61.6
	Reference [12][3]	7	70.74
	Section 4	7	88.19
	Reference [7]	8	51.40
	Section 4	8	56.49
SPECK48/96	Reference [15]	5	—[4]
	Section 4	7	63.43
SPECK64/128	Section 4	8	63.20

[1]We train neural distinguishers using Benamira et al. 's method presented in [10]. [2]In [17], Abed et al. constructed differential characteristics of SIMON. We train neural distinguishers by choosing the input differences in [17]. [3]We choose the highest-accuracy neural distinguisher in [12, 13]. [4] Chen et al. used 5-round neural distinguisher to attack SPECK48/x, but the accuracy was not presented in [15].

ciphertext pairs are from a uniform distribution due to their random input difference. If a neural network can obtain a stable distinguishing accuracy higher than 50\% on a testing set, we call the trained neural network a neural distinguisher. What is particularly noteworthy is that each sample is encrypted by a random key. By this method, the neural distinguisher will work whether the key is changed or not. In [7], Gohr chose the deep residual neural networks to train neural distinguisher and obtained effective neural distinguishers of 5-round, 6-round, and 7-round SPECK32/64.

In traditional differential attack, it is pivotal to distinguish encryption function from a pseudorandom permutation, which is done with the help of the differential characteristic. For an nr-round optimal characteristic $\Delta\alpha \longrightarrow^{2^{-t}} \Delta\beta$ of a block cipher with block size n bits, we calculate the output difference given the fixed input difference $\Delta\alpha$. If the ratio of the output difference to $\Delta\beta$ is about 2^{-t}, then we can distinguish the block cipher from a pseudorandom permutation. This is called distinguishing attack for block ciphers.

For Gohr's neural distinguisher, we can obtain N ciphertext pairs encrypted by the input difference $(0x40, 0x0)$. We input the N ciphertext pairs, and the neural distinguisher will predict their labels. If the ratio of samples labeled 1 exceeds 0.5, we can distinguish the block cipher and pseudorandom permutation and the neural distinguisher is effective. This is called a distinguishing attack based on the neural distinguisher. In addition, it is obvious that the higher the accuracy of the neural distinguisher, the better the effect of the distinguishing attack; and the complexity of key search can also be reduced if the distinguishing accuracy is greatly promoted. So, it is necessary to improve neural distinguisher.

In [7], Gohr explained the reason for choosing $(0x40, 0x0)$ as the input difference that it transitioned deterministically to the low-weight difference $(0x8000, 0x8000)$. But it is pretty hard to find such input differences unless the full differential distribution table is used. Moreover, it is a time-consuming task to calculate the full DDT, especially for large-size block ciphers.

3. An Approach Based on SAT to Improve Neural Distinguisher

In traditional differential cryptanalysis, it is a primary task to find a high-probability differential characteristic, which takes advantage of the unevenness of the differential distribution. The distribution of output differences is different for different input differences. For a neural distinguisher, it actually learns the distribution of output difference given a fixed input difference. Therefore, the input difference directly affects the accuracy of the neural distinguisher.

In [7], Gohr chose $(0x40, 0x0)$ as the input difference to train the distinguisher because it transitioned deterministically to a low-weight output difference. But such input differences are hard to find, which makes it difficult to find effective distinguishers. In [10], Benamira et al. chose the input difference from $nr-1$-round or $nr-2$-round optimal differential characteristics for nr-round neural distinguishers.

In this section, we will introduce our algorithm for improving nr-round neural distinguishers by searching for the nr-round differential characteristics. With the help of SAT/SMT solver, we search for high-probability differential characteristics with probability in $[2^{-(b_s/4)} \times P_{\max}, P_{\max}]$, where $P_{\max}$ is the optimal probability and b_s is the block size. Using our algorithm, we can obtain high-accuracy neural distinguishers for 9-round SIMON32/64, 10-round SIMON48/96, and 11-round SIMON64/128.

3.1. Generic Network Architecture. Gohr converted the distinguisher of ciphertext pairs into a binary classification problem. His method is not only applicable to SPECK but also applicable to SIMON. With his method, we can construct a generic network architecture for other ciphers. We refer to [7] for the description of the method of constructing the network architecture.

There are multiple neural networks available to train neural distinguishers, such as MIP and ResNets. We choose the ResNets to train a neural distinguisher.

Our networks comprise three main components: input layer, iteration layer, and predict layer, shown in Figure 1. n in Figure 1 refers to the word size of SIMON $2n/mn$. The input layer receives training data with fixed length. In the iteration layer, we use 5 residual blocks. In each residual block, we use two Conv1D layers, and each Conv1D layer is followed by a batch normalization layer and an activation layer. After flattening data from iteration layer, data will be sent into a fully connected layer. The fully connected layer consists of a hidden layer and an output unit.

In our network, we choose the kernel size of the first Conv1D layer as 1 and the kernel size of the other Conv1D layer is 3. In addition, the number of filters in each convolutional layer is $2n$ and the padding method is SAME. At last, we train our network based on L2 weights regularization to avoid overfitting. The other details of the hyperparameters used are given in Table 2. In Table 2, we choose hyperparameters similar to those in Gohr's choice, so we can ignore the influence of the neural network and its parameters. After the neural distinguisher is trained, we can use it to distinguish the output of target cipher with a given input difference from random data. The higher its accuracy on the test set is, the better it distinguishes ciphertext data.

3.2. An Algorithm Based on SAT to Improve Neural Distinguisher. SAT is the Boolean Satisfiability Problem. It is an NP-complete problem and considers whether there is a valid assignment to Boolean variables satisfying a given set of Boolean conditions. As the key issue of computer science and artificial intelligence, SAT solvers have gained a lot of attention since it was proposed. It has great advantages of open source, good interface, high efficiency, and perfect compatibility. There are many cryptanalysis results based on SAT [18–20].

At present, there are two main ways to select the input differences of neural distinguishers. One way is to directly choose an existing optimal differential characteristic [12], and the other is to choose $nr-1$-round or $nr-2$-round optimal differential characteristics for nr-round neural distinguishers [10]. But these methods cannot effectively promote the distinguishing accuracy.

Taking into account the unevenness of the distribution of output differences for different input differences, we decide to choose the input differences of high-probability differential characteristics as the candidate differences. We search for high-probability differential characteristic by a SAT-based automatic search tool and train neural distinguishers with these input differences of differential characteristics. Based on this, we design an algorithm to help us search for neural distinguishers with higher accuracy, which is shown in Algorithm 1. In Algorithm 1, we expand the search space of input difference by expanding the range of the probability. We choose $[2^{-(b_s/4)} \times P_{\max}, P_{\max}]$ as the lower bound of the probability, where $P_{\max}$ is the probability of the optimal differential characteristics and b_s refers to the block size of the target cipher. By experimental experience, we find that if the differential probability is lower than $2^{-(b_s/4)} \times P_{\max}$, there are almost no high-accuracy neural distinguishers. So there is nearly no need to spend time on the differential characteristics with the probability lower than $2^{-(b_s/4)} \times P_{\max}$.

Using Theorem 3 in [19] and open-source SAT/SMT solver Z3 [21], we search for high-probability differential characteristics of SIMON. Then, with Algorithm 1, we get the 9-, 10-, and 11-round neural distinguishers of SIMON32/64, SIMON48/96, and SIMON64/128, respectively. This is the first time that there is a neural distinguisher of 11-round SIMON64/128. The results of the neural distinguishers are shown in Table 3.

In order to show that Algorithm 1 is effective, we use the other two methods [10, 12] of selecting the input difference to train the neural distinguishers with the same rounds. For the method presented in [12], we choose $(0x10100, 0x44040)$ in [17] as input difference to train 9-round neural distinguisher. Besides, for the other method presented in [10], we train 10-round neural distinguishers of SIMON48/96 using 9-round and 8-round optimal differential characteristics, and the specific results are shown in Tables 4 and 5.

In Table 3, we choose the same data format as that of Gohr's distinguisher, which is single ciphertext pair. Other hyperparameters are posted in Table 2. As shown in Table 3, we show the comparison of the accuracy from three methods of selecting the input difference. As we can see, compared with selecting the input difference in [10, 17], the accuracy of neural distinguishers obtained by Algorithm 1 has been significantly promoted, which can be used to reduce the complexity of key recovery attack. Although both methods select the input difference from differential characteristics, Algorithm 1 selects the exact rounds of the differential characteristics according to the rounds of neural distinguisher.

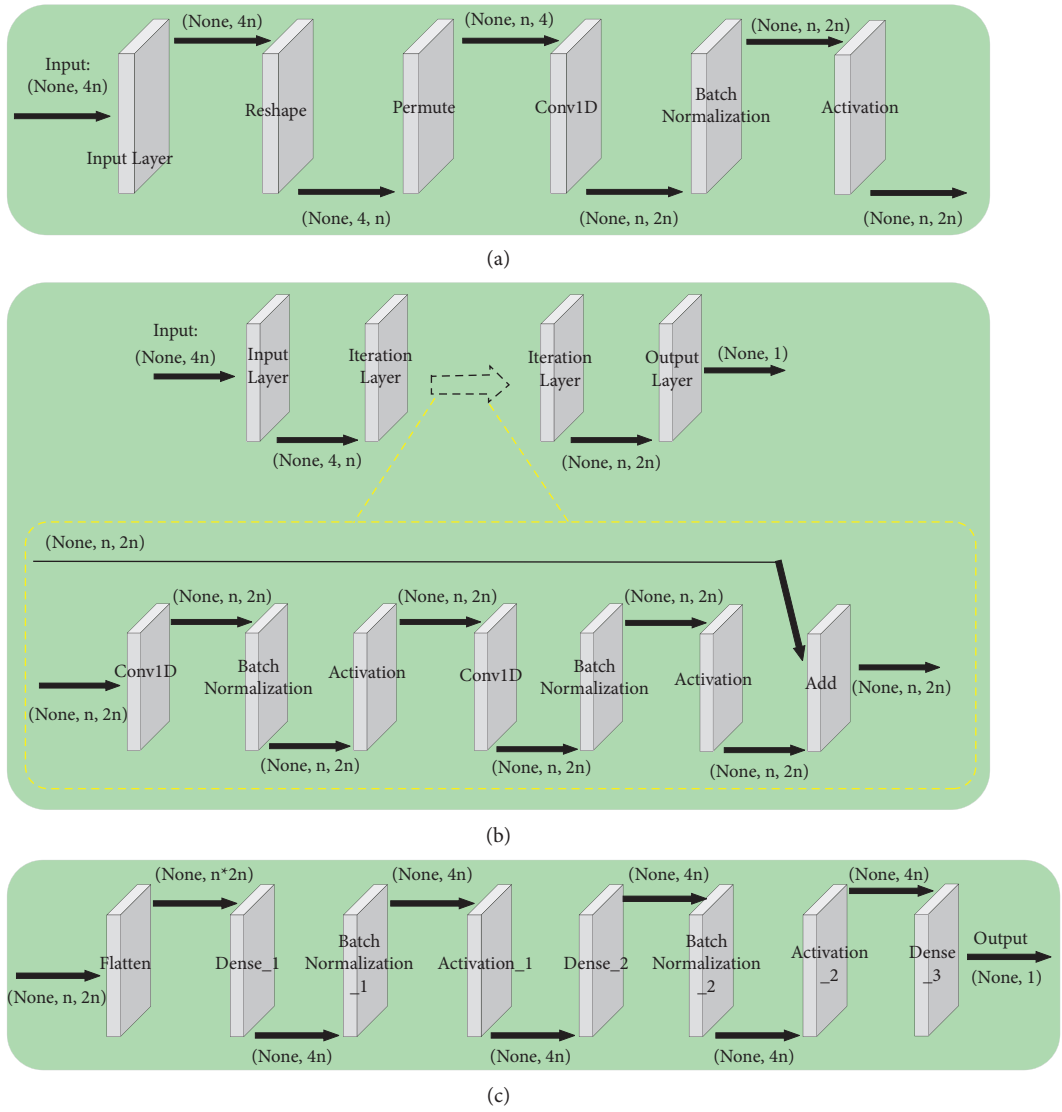

Figure 1: Network architecture. (a) Input Layer. (b) Iteration Layer. (c) Output Layer.

Table 2: List of hyperparameters.

Hyperparameters	Value
Batch size	10000
Epochs	100
Train size	10^7
Validation size	10^5
Regularization parameter	10^{-4}
Optimizer	Adam
Loss function	MSE (mean squared error)

We also try to search for neural distinguishers for more rounds. Unfortunately, as the number of rounds increases, the nonrandom features of the ciphertext pairs become weaker and weaker. So it is difficult for us to find a neural distinguisher with longer round, even if using Algorithm 1. In addition, the higher the Hamming weight of the input difference, the weaker the nonrandom feature of the ciphertext pair. So we should firstly search for input differences with lower Hamming weight adopting Algorithm 1 if time is limited.

4. A New Neural Distinguisher Model Using Multiple Output Differences

In [10], Benamira et al. show that the neural distinguisher generally relies on not only the differential distribution of

Table 3: Comparison with Algorithm 1 and other methods.

Block cipher	Source of neural distinguisher (input difference)	Rounds	Input difference	Accuracy (%)
SIMON32/64	Reference [17][1]	9	$(0x0, 0x40)$	59.07
	Algorithm 1	9	$(0x0, 0x80)$	59.77
SIMON48/96	Reference [17][1]	9	$(0x10100, 0x44040)$	50.22
	Reference [10][2]	10	$(0x80000, 0x222000)$	53.49
	Algorithm 1	10	$(0x0, 0x100000)$	57.89
SIMON64/128	Reference [17][1]	10	$(0x100, 0x440)$	58.61
	Algorithm 1	11	$(0x0, 0x10)$	59.72

[1]In [17], Abed et al. constructed differential characteristics of SIMON. We train neural distinguishers by choosing the input differences in [17]. [2]We train neural distinguishers using Benamira et al.'s method presented in [10].

Input: Network Architecture Net, Cipher C with block size b_s bits, Round R.
Output: Neural distinguisher ND, Input difference of distinguisher I_d.

(1) Search for the optimal probability as $P_{\max}$
(2) Search for the differential characteristics with probability in $[2^{-(b_s/4)} \times P_{\max}, P_{\max}]$, and save their input differences as DIFF
(3) ND = [ ]
(4) I_d = [ ]
(5) **for** d in DIFF **do**
(6) $S = C(d, R)$ *#Generate train set using d as the input difference*
(7) $V = C(d, R)$ *#Generate test set using d as the input difference*
(8) $D_d = \text{Net}(S)$ *#Training using train data*
(9) $\text{acc}_d = \text{Evaluate}(D_d, V)$*#Get the accuracy of the model* D_d
(10) **if** $\text{acc}_d > 0.51$ **then**
(11) $\text{ND} = \text{ND} \| D_d$
(12) $I_d = I_d \| d$
(13) **end if**
(14) **end for**
(15) return (ND, I_d)

Algorithm 1: Search for neural distinguisher based on SAT.

Table 4: 10-round neural distinguishers using 9-round optimal characteristics.

Input difference	Accuracy	Input difference	Accuracy	Input difference	Accuracy
$(0x2, 0x880008)$	0.5258	$(0x100, 0x444)$	0.5279	$(0x800, 0x2220)$	0.5281
$(0x8000, 0x22200)$	0.5286	$(0x4000, 0x11100)$	0.5286	$(0x80000, 0x222000)$	0.5288
$(0x100000, 0x444000)$	0.5290	$(0x20000, 0x88800)$	0.5293	$(0x20000, 0x88800)$	0.5293
$(0x20, 0x800088)$	0.5301	$(0x200000, 0x888000)$	0.5301	$(0x200, 0x888)$	0.5303
$(0x2000, 0x8880)$	0.5303	$(0x4, 0x100011)$	0.5308	$(0x80, 0x222)$	0.5310
$(0x400, 0x1110)$	0.5311	$(0x40, 0x111)$	0.5312	$(0x10, 0x400044)$	0.5312
$(0x400000, 0x110001)$	0.5312	$(0x8, 0x200022)$	0.5315	$(0x1, 0x440004)$	0.5327
$(0x800000, 0x220002)$	0.5327	$(0x1000, 0x4440)$	0.5330	$(0x40000, 0x111000)$	0.5334

Table 5: 10-round neural distinguishers using 8-round optimal characteristics.

Input difference	Accuracy	Input difference	Accuracy	Input difference	Accuracy
$(0x400004, 0x10)$	0.5264	$(0x100, 0x444)$	0.5282	$(0x8, 0x200022)$	0.5283
$(0x88, 0x200)$	0.5285	$(0x2000, 0x8880)$	0.5286	$(0x400, 0x1110)$	0.5287
$(0x40, 0x111)$	0.5289	$(0x440, 0x1000)$	0.5290	$(0x40000, 0x111000)$	0.5291
$(0x100001, 0x4)$	0.5294	$(0x400000, 0x110001)$	0.5296	$(0x20000, 0x88800)$	0.5298
$(0x800000, 0x220002)$	0.5298	$(0x110000, 0x400000)$	0.5298	$(0x11000, 0x40000)$	0.5300
$(0x88000, 0x200000)$	0.5301	$(0x220, 0x800)$	0.5301	$(0x800, 0x2220)$	0.5302
$(0x800, 0x2220)$	0.5302	$(0x2200, 0x8000)$	0.5304	$(0x8800, 0x20000)$	0.5304
$(0x200, 0x888)$	0.5305	$(0x880000, 0x2)$	0.5306	$(0x2, 0x880008)$	0.5306
$(0x100000, 0x444000)$	0.5307	$(0x1, 0x440004)$	0.5307	$(0x10000, 0x44400)$	0.5308
$(0x1100, 0x4000)$	0.5309	$(0x44000, 0x100000)$	0.5311	$(0x110, 0x400)$	0.5313

Table 5: Continued.

Input difference	Accuracy	Input difference	Accuracy	Input difference	Accuracy
$(0x200002, 0x8)$	0.5313	$(0x8000, 0x22200)$	0.5314	$(0x220000, 0x800000)$	0.5314
$(0x22, 0x80)$	0.5318	$(0x2, 0x880008)$	0.5319	$(0x200000, 0x888000)$	0.5320
$(0x80, 0x222)$	0.5322	$(0x1000, 0x4440)$	0.5323	$(0x10, 0x400044)$	0.5325
$(0x4400, 0x10000)$	0.5326	$(0x4, 0x100011)$	0.5327	$(0x44, 0x100)$	0.5327
$(0x440000, 0x1)$	0.5329	$(0x20, 0x800088)$	0.5339	$(0x4000, 0x11100)$	0.5340
$(0x22000, 0x80000)$	0.5341	$(0x11, 0x40)$	0.5341	$(0x80000, 0x222000)$	0.5349

ciphertext pairs but also the differential distribution in penultimate and antepenultimate rounds. This enlightens us whether we can directly use the output differences to train neural distinguishers. Unlike [7, 12] using ciphertext pairs as samples, we design a new neural distinguisher model with multiple output differences as a sample. Using the new model, we obtain the high-accuracy neural distinguishers for 10-round SIMON32/64, 11-round SIMON48/96, 12-round SIMON64/128, 8-round SPECK32/64, 7-round SPECK48/96, and 8-round SPECK64/128. Additionally, we show with experiments that the promotion in the accuracy of distinguishers is not due to the increase of the number of plaintexts but learning more features from the relationship between the output differences.

4.1. New Neural Distinguisher Model. Neural networks and deep learning currently provide the best solutions to many problems in image recognition, speech recognition, and natural language processing. As we know, the deep learning is data-driven, and the quality of the data determines the quality of the model to some extent. For neural distinguishers, the choice of ciphertext pairs directly affects the accuracy of the neural distinguishers, which has been solved in Section 3. In deep learning field, the format of training data also affects the quality of the trained model to some extent. This enlightens us that we can improve neural distinguishers from the perspective of data format. In image recognition, the deep learning researchers currently rotate the image or crop it to enhance some objective features, which has been experimentally proven to be effective. Inspired by Benamira et al.'s work and data augmentation in deep learning, we use the output differences to train neural distinguishers by splicing output differences into a matrix as a sample. For a matrix, we treat it as an image and each output difference of the matrix is treated as an objective feature. Our goal is not only to learn each objective feature but also to learn the connections between output differences.

As shown in Figure 2, the k plaintext pairs $((P_1^1, P_2^1), (P_1^2, P_2^2), \ldots, (P_1^k, P_2^k))$ are encrypted by a random master key. The k ciphertext pairs $((C_1^1, C_2^1), (C_1^2, C_2^2), \ldots, (C_1^k, C_2^k))$ are converted to output differences, where n is the block size of ciphers. We splice multiple output differences into a matrix as a sample, which is described as $O_1 \| O_2 \| \ldots$. Similar to Gohr's method, given an input difference I_d, each sample will be attached a label Y according to the following equation:

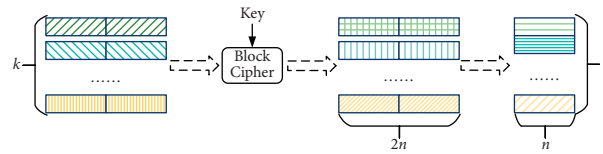

Figure 2: A new data format using k different output differences.

$$Y\left(O_1 \| O_2 \| \ldots \| O_k\right) = \begin{cases} 1, & \text{if } P_1^i \oplus P_2^i = I_d, \quad i \in [1, k], \\ 0, & \text{else.} \end{cases} \tag{2}$$

If the label is 1, the matrix is denoted as a positive sample. Otherwise, it is denoted as a negative sample. We call the new data format $\mathrm{DF}_{\mathrm{different}}$. By randomly generating plaintext and key, we make our distinguishers learn the features of target block cipher instead of the features of the plaintext or key. In the experiment, we make the neural network learn more features by using more output differences in a matrix. As we can see, the new data format needs more ciphertext pairs. For the same number of training sets, the new model requires k times more data than Gohr's model.

Because only the channel dimension is changed, we refer to Figure 1 for the description of network architecture.

4.2. Applications to SIMON and SPECK

4.2.1. Application to SIMON. We choose the input difference in Table 3 to train new neural distinguishers. Other hyperparameters are posted in Table 2. The accuracy comparison is presented in Table 6. For 11-round SIMON48/96, we do not obtain an effective neural distinguisher using the input difference in Table 3. So we research other high-probability differential transmissions.

In Table 6, the "SCP" refers to the data format of Gohr's neural distinguisher, and the "MOD" refers to the data format shown in Figure 2. As shown in Table 6, compared with using ciphertext pairs, the number of rounds and accuracy of new neural distinguishers are greatly promoted. In addition, the new distinguishers can be further promoted by increasing k, which shows that the superposition of output difference can help the neural network to learn more unknown features.

4.2.2. Application to SPECK. The new format is not limited to the neural distinguisher of SIMON, as it can also be found to be effective in SPECK. In [7, 12], $(0x40, 0x0)$ is used to train neural distinguisher of 7-round SPECK32/64. Using the

Table 6: Comparison of SIMON using different data format.

Cipher	Data format	k	Rounds	Input difference	Accuracy (%)
SIMON32/64	SCP	—	9	$(0x0, 0x80)$	59.07
	MOD	4	9	$(0x0, 0x80)$	62.27
	MOD	32	9	$(0x0, 0x80)$	82.27
	MOD	32	10	$(0x0, 0x80)$	61.09
SIMON48/96	SCP	—	10	$(0x0, 0x100000)$	57.89
	MOD	4	10	$(0x0, 0x100000)$	61.15
	MOD	48	10	$(0x0, 0x100000)$	81.40
	MOD	48	11	$(0x1000, 0x4400)$	61.43
SIMON64/128	SCP	—	11	$(0x0, 0x10)$	59.72
	MOD	4	11	$(0x0, 0x10)$	63.53
	MOD	64	11	$(0x0, 0x10)$	73.79
	MOD	64	12	$(0x0, 0x10)$	69.57

SCP: single ciphertext pair; MOD: multiple output differences.

Table 7: Comparison of SPECK using different data format.

Cipher	Data format	Rounds	Input difference	Accuracy (%)	Source
SPECK32/64	SCP	7	$(0x40, 0x0)$	61.6	[7]
	MCP	7	$(0x40, 0x0)$	70.74[1]	[12]
	MOD[2]	7	$(0x40, 0x0)$	88.87	Section 4
	SCP	8	$(0x40, 0x0)$	51.4	[7]
	MOD[2]	8	$(0x2800, 0x10)$	56.49	Section 4
SIMON48/96	MCP	5	—[3]	—	[15]
	MOD[4]	7	$(0x20082, 0x120200)$	63.43	Section 4
SIMON64/128	MOD[5]	8	$(0x0, 0x10)$	63.20	Section 4

SCP: single ciphertext pair; MCP: multiple ciphertext pairs; MOD: multiple output differences. [1]The highest accuracy of 7-round SPECK32/64 in [12]. [2]$k = 32$. [3]Chen et al. used 5-round neural distinguisher to attack SPECK48/x, but the accuracy was not presented in [15]. [4]$k = 48$. [5]$k = 64$.

difference, we obtain a new higher-accuracy neural distinguisher of 7-round SPECK32/64. Not only that, with the help of [18, 20], we obtain a good input difference $(0x2800, 0x10)$ and an effective 8-round neural distinguisher. As far as we know, this is the first effective 8-round neural distinguisher of SPECK32/64 with accuracy more than 55\%. Besides, we also obtain neural distinguishers of 7-round SPECK48/96 and 8-round SPECK64/128. Summary of the existing results is shown in Table 7. In Table 7, "SCP" refers to the data format of Gohr's neural distinguisher, "MCP" refers to the data format using multiple ciphertext pairs, and "MOD" refers to the data format shown in Figure 2. Other hyperparameters are posted in Table 2.

Utilizing the new model, we improve neural distinguishers in terms of length and accuracy. We can achieve better results in distinguishing attack utilizing the new neural distinguishers. Moreover, we give a further illustration of our model. Since we use more data in the new model than using ciphertext pairs, this makes our improved results seem to be related to increase of data. We perform supplementary experiments to show that the improvement of the accuracy of distinguishers is not due to the increase of the number of plaintexts but because of learning more features from the relationship between the output differences.

4.3. A Supplementary Explanation to Our New Model. Although the accuracy is higher using the new data format, the performance may be likely improved by training on more training samples. So we use the same number of ciphertext pairs to train neural distinguishers shown in Table 8. Other hyperparameters are posted in Table 2.

In Table 8, "SCP" refers to the data format of Gohr's neural distinguisher, and "MOD" refers to the data format shown in Figure 2. As shown in Table 8, the accuracy using multiple output differences is higher, even if the same amount of data is used. In addition, it takes up less memory using output differences, which can reduce training time in the training process.

To further illustrate the effectiveness of the new distinguishers, we conduct additional experiments. As shown in Figure 3, we use k same output differences as a sample; And we call the data format DF_{same}. As shown in Figures 2 and 3, $DF_{different}$ uses k different output difference in a sample, while DF_{same} uses k same output difference in a sample. Based on the data format DF_{same}, 10^6 positive and negative ciphertext pairs are randomly generated; and each output difference is reused k times and filled in a matrix as a sample. Then new neural distinguishers are performed on 10^6 samples. We calculate the accuracy of the new neural distinguishers for these special data. Table 9 shows the corresponding test results.

In Table 9, the "Accuracy using DF_{same}" refers to the accuracy of neural distinguishers trained by DF_{same}. The "Accuracy using $DF_{different}$" refers to the accuracy of neural distinguishers trained by $DF_{different}$. As shown in Table 9, the accuracy using DF_{same} is lower than that using $DF_{different}$.

Table 8: Comparison with using ciphertext pairs and output differences.

Cipher	Data format	k	Rounds	Size of training data	Accuracy (%)
SIMON32/64	SCP	—	10	32×10^7	50.28
	MOD	32	10	10^7	61.09
SIMON48/96	SCP	—	11	48×10^7	50.43
	MOD	48	11	10^7	61.43
SIMON64/128	SCP	—	12	64×10^7	50.37
	MOD	64	12	10^7	69.57

SCP: single ciphertext pair; MOD: multiple output differences.

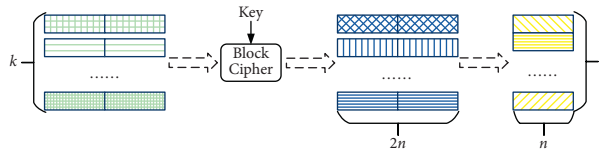

Figure 3: A data format using k same output differences.

Table 9: Accuracy comparison with DF_{same} and $DF_{different}$.

Ciphers	k	Rounds	Accuracy using DF_{same}(%)	Accuracy using $DF_{different}$(%)
SIMON32/64	32	10	51.69	61.09
SIMON48/96	48	11	54.81	61.43
SIMON64/128	64	12	49.78	69.57
SPECK32/64	32	8	51.18	56.49
SPECK48/96	48	7	50.06	63.43
SPECK64/128	64	8	50.84	63.20

Table 10: SIMON parameters.

Block size $2n$	Key size mn	Word size n	Rounds T
32	64	16	32
48	72	24	36
	96	24	36
64	96	32	42
	128	32	44
96	96	48	52
	144	48	54
128	128	64	68
	192	64	69
	256	64	72

Table 11: SPECK parameters.

Block size $2n$	Key size mn	Word size n	Rot α	Rot β	Rounds T
32	64	16	7	2	22
48	72	24	8	3	22
	96	24	8	3	23
64	96	32	8	3	26
	128	32	8	3	27
96	96	48	8	3	28
	144	48	8	3	29
128	128	64	8	3	32
	192	64	8	3	33
	256	64	8	3	34

This illustrates that the new distinguishers learn more unknown features especially in the connection of different output differences.

5. Conclusion and Future Work

In this paper, we proposed a new algorithm and model to further improve neural distinguishers. On the one hand, by carefully selecting the input differences with utilizing SAT/SMT algorithm, we managed to search for exact nr-round differential characteristics with high probability and trained nr-round neural distinguishers. On the other hand, by adopting the new data format, we spliced multiple output differences into a matrix as a sample to capture more derived features; thus we can improve the number of rounds and accuracy of neural distinguishers. Application to SIMON and SPECK has proved the superiorities of our algorithm and models.

With our results, we obtain new effective neural distinguishers, which can be used to distinguish reduced-round SIMON or SPECK from pseudorandom permutation better. Since there are numerous network architectures now with the development of deep learning, it is meaningful to explore other appropriate network models to improve neural distinguishers.

Appendix

A. Brief Description of SIMON and SPECK

SIMON

SIMON [8] is a lightweight block cipher proposed by the NSA (National Security Agency). The aim of SIMON is to fill the need for secure, flexible, and analyzable lightweight block ciphers. It is a family of lightweight block ciphers with block sizes of 32, 48, 64, 96, and 128 bits. The constructions are Feistel ciphers using a word size n of 16, 24, 32, 48, or 64 bits, respectively. Table 10 makes explicit all parameter choices for all versions of SIMON.

For SIMON $2n/mn$, the key-dependent SIMON $2n/mn$ round function is the map $R_{k_i}\mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \longrightarrow \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$ defined by

$$R_{k_i}(x_i, y_i) = (y_i \oplus f(x_i) \oplus k_i, x_i), \tag{A.1}$$

where $f(x_i) = (S^1 x_i \odot S^8 x_i) \oplus S^2 x_i k_i (k_i \in GF(2)^n)$ is the round subkey.

SPECK

Similar to SIMON, there are some different variants of SPECK, and these parameters about SPECK are shown in Table 11.

For SPECK $2n/mn$, the key-dependent SPECK $2n/mn$ round function is the map $R_{k_i} : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \longrightarrow \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$ defined by

$$R_{k_i}(x_i, y_i) = \left((S^{-\alpha} x_i + y_i) \oplus k_i, (S^{-\alpha} x_i + y_i) \oplus k_i \oplus S^{\beta} y_i\right), \tag{A.2}$$

where $k_i (k_i \in \mathrm{GF}(2)^n)$ is the round subkey.

As it is out of scope for our purpose, we refer to [8] for the description of the key-scheduling.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

A preprint has previously been published in [22].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Lawrence, C. L. Giles, A. D. Tsoi, and A. C. Tsoi, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[2] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[3] I. Goodfellow, J. A. Pouget, and M. Mirza, "Generative adversarial nets," in *Proceedings of the International Conference On Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.

[4] R. L. Rivest, "Cryptography and machine learning," in *Proceedings of the Advances In Cryptology — ASIACRYPT '91*, pp. 427–439, Fujiyosida, Japan, November 1991.

[5] L. Masure, C. Dumas, and E. Prouff, "Gradient visualization for general characterization in profiling attacks," in *Proceedings of the Constructive Side-Channel Analysis And Secure Design—COSADE 2019*, pp. 145–167, Darmstadt, Germany, April 2019.

[6] L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 348–375, 2020.

[7] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," in *Proceedings of the Advances in Cryptology - CRYPTO 2019*, pp. 150–179, Santa Barbara, CA, USA, August 2019.

[8] R. Beaulieu, D. Shors, and J. Smith, "The SIMON and SPECK Families of Lightweight Block Ciphers," *Eprint*, vol. 2013, no. 404, 2013.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV USA, December 2016.

[10] A. Benamira, D. Gerault, T. Peyrin, and Q. Tan, "A deeper look at machine learning-based cryptanalysis, lecture notes in computer science," in *Proceedings of the Advances In Cryptology – EUROCRYPT 2021*, pp. 805–835, Zagreb, Croatia, October 2021.

[11] Y. Chen and H. Yu, "Bridging machine learning and cryptanalysis via EDLCT," *IACR eprint*, vol. 2021, no. 705, 2021.

[12] Y. Chen and H. Yu, "A new neural distinguisher considering features derived from multiple ciphertext pairs," *Eprint*, vol. 2021, no. 310, 2021.

[13] H. Su, X. Zhu, and D. Ming, "Polytopic attack on round-reduced simon32/64 using deep learning," in *Proceedings of the Information Security And Cryptology – Inscrypt 2020*, pp. 3–20, Guangzhou, China, December 2020.

[14] Y. Chen and H. Yu, "Neural aided statistical attack for cryptanalysis," *IACR eprint*, vol. 2020, no. 1620, 2020.

[15] Y. Chen and H. Yu, "Improved neural aided statistical attack for cryptanalysis," *IACR eprint*, vol. 2021, no. 311, 2021.

[16] Z. Bao, J. Guo, and M. Liu, "Conditional differential-neural cryptanalysis," *IACR eprint*, vol. 2021, no. 719, 2021.

[17] F. Abed, E. List, and S. Lucks, "Differential cryptanalysis of round-reduced simon and speck," in *Proceedings of the Fast Software Encryption. – FSE 2014*, pp. 525–545, London, UK, March 2014.

[18] N. Mouha and B. Preneel, "Towards finding optimal differential characteristics for arx: application to salsa20," *IACR eprint*, vol. 2013, no. 328, 2013.

[19] S. Kölbl, G. Leander, and T. Tiessen, "Observations on the SIMON block cipher family," in *Proceedings of the Advances In Cryptology – CRYPTO 2015*, pp. 161–185, Santa Barbara, CA, USA, August 2015.

[20] Y. Liu, Q. Wang, and V. Rijmen, *Automatic Search of Linear Trails in ARX with Applications to SPECK and Chaskey*, pp. 485–499, Applied Cryptography And Network Security – ACNS 2016, Guildford, UK, 2015.

[21] L. d. Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *Proceedings of the Tools And Algorithms For the Construction And Analysis Of Systems – TACAS 2008*, pp. 337–340, Budapest, Hungary, April 2008.

[22] Z. Hou, J. Ren, and S. Chen, "Improve Neural Distinguisher for Cryptanalysis," *IACR eprint*, vol. 2021, no. 1017, 2021.