Fixed-Budget Allocation in Neural Differential Distinguishers

Alireza Gholizadeh Shahrbejari* and Reza Ebrahimi Atani†

Department of Computer Engineering, University of Guilan, Rasht, Iran

Abstract

Neural differential distinguishers are usually compared at a fixed number of labeled samples. However, different input representations may require different numbers of ciphertexts per sample, making fixed-sample comparisons potentially misleading from a cryptanalytic data-complexity perspective. In this paper, we study neural differential distinguishers under a fixed ciphertext budget. We ask whether the available encryption queries should be spent on more independent plaintext bases, or on richer samples containing more ciphertext-difference rows.

We introduce a shared-base multi-difference representation in which several input differences are applied around the same plaintext base, and compare it with the standard single-difference baseline and an independent-pair control representation. Experiments on GIFT-64, PRESENT-64, RECTANGLE-64, and SPECK-64/128 show that the single-difference baseline is rarely the best fixed-budget allocation. Adding more difference rows often improves the distinguisher even though it reduces the number of independent training samples. At the same time, the best tested number of rows is not universal: logistic regression often benefits from larger representations, while a multilayer perceptron frequently prefers intermediate values due to sample-starvation and overfitting.

We further test several non-adaptive difference sets and observe that the main trend is not tied to a single hand-picked set. The results suggest that the number of differences per sample should be treated as an explicit design parameter in neural differential cryptanalysis, and that fixed-budget evaluation is necessary for comparing richer neural distinguisher inputs fairly.

1 Introduction

Differential cryptanalysis is one of the central techniques for evaluating the security of block ciphers. Since the seminal work of Biham and Shamir [7], many attacks have relied on carefully selected input differences and on the statistical behavior of the corresponding output differences. In classical differential cryptanalysis, the analyst typically searches for good trails, differentials, or distinguishers using mathematical or algorithmic tools, and then uses the resulting statistical bias to distinguish a reduced-round cipher from a random permutation.

In recent years, neural networks have provided a new way to build differential distinguishers. Instead of manually designing the full statistic used by the distinguisher, one can generate labeled ciphertext data from structured chosen-plaintext queries and train a classifier to separate real cipher outputs from random-reference outputs. Gohr's work on round-reduced SPECK showed that deep learning can significantly improve attacks in this setting [13]. This result initiated a large body of work on neural differential cryptanalysis, including studies of what neural distinguishers learn, how to improve their training, how to select good input differences, and how to construct richer input representations.

*Email: gholizadeh.a2000@gmail.com

†Email: rebrahimi@guilan.ac.ir

Most neural differential distinguishers are based on some form of chosen input difference. In the simplest setting, one fixes a difference Δ, generates plaintext pairs of the form $(P, P \oplus \Delta)$, encrypts them through a reduced-round cipher, and feeds a ciphertext-derived representation to a neural classifier. This design creates many independent samples when the number of queries is fixed. However, it exposes only one differential view of the cipher per sample. A natural alternative is to include several ciphertext-difference rows inside each sample. Such a representation may give the model a richer view of the cipher's differential behavior, but it also consumes more encryption queries per sample.

This leads to the main question of this paper. If the total number of ciphertexts is fixed, should the data-generation budget be spent on more independent plaintext bases, or on more differences around each base plaintext? Equivalently, is it better to train a neural distinguisher from many simple single-difference samples, or from fewer but richer multi-difference samples? This question is important because comparing two representations at the same number of labeled samples does not control the cryptanalytic cost. A richer representation may appear stronger simply because it uses more ciphertexts. In a cryptanalytic setting, the number of encryption queries is a natural resource, and therefore the comparison should be made under a fixed ciphertext budget.

1.1 Related work

Gohr introduced a neural differential distinguisher for round-reduced SPECK and showed that deep learning can improve practical attacks in a differential cryptanalysis setting [13]. Benamira et al. later provided a deeper analysis of machine-learning-based cryptanalysis and studied how the neural distinguisher relates to classical differential features [6]. Gohr, Leander, and Neumann further assessed several differential-neural distinguishers and studied how they can be optimized and understood through distributional differences [14]. Other works improved neural distinguishers through training strategies, architectural changes, and cryptanalytic refinements [2, 26, 20]. More recently, Ren et al. used neural architecture search with an information-entropy objective to automate neural-distinguisher design across several ciphertext datasets [22], while Liu et al. proposed an SPN-oriented framework that jointly redesigns the training-data format and network structure [18]. A recent survey by Gérault et al. systematizes six years of neural differential cryptanalysis and highlights comparability across neural distinguishers as an important methodological challenge [11].

A closely related line of work studies how to construct the input data given to the neural distinguisher. Chen et al. proposed a neural distinguisher that considers features derived from multiple ciphertext pairs simultaneously [9]. Their motivation is that several ciphertext pairs may expose features that are not available from a single pair. Hou et al. also studied multiple ciphertext-pair inputs for SPECK and SIMON and reported improved accuracy compared with previous distinguishers [16]. Yue and Wu proposed an improved neural differential distinguisher for SPECK with a new data format and network modifications, and also studied the effect of using multiple ciphertext pairs in a sample [30]. Liu, Ren, and Chen considered single- and multiple-ciphertext-pair input formats in a lightweight and more general improvement framework [19]. Zhang et al. studied differential-neural cryptanalysis of SIMECK and evaluated multi-pair distinguishers in comparison with DDT-based distinguishers [31]. Hu et al. introduced MCPLD, which combines multiple ciphertext pairs with leaked difference bits to provide richer features to the network [17]. Bellini et al. studied generic partial decryption as a feature-engineering technique and evaluated its contribution separately from the number of pairs per sample [4]. Mirzaali et al. further explored richer differential structure through polytopic differential neural distinguishers for SIMON, SIMECK, and SPECK [21]. Yuan and Wang extended multi-differential input construction to the related-key setting and jointly searched input- and key-difference combinations [29]. At a finer input granularity, Gauthier-Umaña et al. used entropy-based selection of informative ciphertext-difference bits to construct a compact

PRESENT distinguisher and a neural-assisted key-recovery procedure [10]. Very recently, Guo, Duan, and Zhu combined multi-sample-pair inputs with attention-based modelling for neural distinguishers targeting SHA-3-256 and Ascon-Hash256 [15].

These works show that richer or more carefully selected inputs can improve neural distinguishers. However, they mainly target distinguishing accuracy, attack depth, feature selection, or network design. The question considered here is different: when a richer representation consumes more ciphertexts per sample, does it remain a better use of a *fixed total ciphertext budget*? A fixed-sample comparison cannot answer this question because it implicitly grants a larger encryption budget to representations that use more ciphertexts per sample.

Dataset construction and difference selection have also received increasing attention. Bellini et al. proposed a cipher-agnostic neural training pipeline with automated search for useful input differences [5]. Seok and Lee studied dataset construction for differential-neural cryptanalysis and emphasized that data quality strongly affects the resulting distinguisher [23]. Xu, Zhang, and Yuan proposed an evaluation algorithm for identifying high-quality differential datasets before neural-distinguisher training [27]. Related work has also explored generic machine-learning extensions of differential cryptanalysis [28], neural distinguishers for lightweight ciphers such as GIFT and TinyJAMBU [25], and neural differential distinguishers for GIFT-128 and ASCON [24].

Gholizadeh Shahrbejari and Ebrahimi Atani [12] compared single-difference and multi-difference representations across the same four lightweight ciphers while controlling the model architecture. That study focused on separating gains due to representation from gains due to model capacity, and included an equal-ciphertext-budget comparison as a supplementary analysis. Building on that observation, the present study makes ciphertext-budget allocation the primary research question. It systematically varies the number of retained difference rows, evaluates two total ciphertext budgets, introduces an independent-pair control for matched-cost comparisons, examines train–test behavior as the representation becomes sample-starved, and tests robustness across several non-adaptive difference sets.

Accordingly, the present paper is closest in spirit to the multi-pair, multi-feature, and dataset-design literature, but it separates two competing uses of the same cryptanalytic resource. The available ciphertexts can be spent on more independent plaintext bases or on more ciphertext-difference rows per sample. We formulate this choice as a fixed-budget allocation problem and study how the best observed allocation changes with the cipher, model, round, budget, and difference set.

1.2 Our approach and contributions

We study neural differential distinguishers under a fixed ciphertext budget. We compare three related input representations. The first is the standard single-difference baseline. The second is a shared-base multi-difference representation, where several input differences are applied around the same plaintext base and the resulting ciphertext differences form a multi-row input matrix. The third is an independent-pair control representation, where the rows are produced from independent plaintext pairs. This control allows us to separate the benefit of having multiple rows from the benefit of sharing a common plaintext base.

Our experiments cover four lightweight 64-bit block ciphers: GIFT-64 [1], PRESENT-64 [8], RECTANGLE-64 [32], and SPECK-64/128 [3]. The first three are SPN-type ciphers, while SPECK is an ARX cipher. We use two simple classifier families: logistic regression and a compact multilayer perceptron. This allows us to study how the budget allocation interacts with model capacity without turning the work into an architecture search.

The main contributions of this paper are as follows.

- We formulate neural differential distinguisher construction as a fixed-ciphertext-budget allocation problem. The key trade-off is between the number of independent plaintext

bases and the number of ciphertext differences included in each sample.

- We define a shared-base multi-difference representation and compare it with both the single-difference baseline and an independent-pair control. Matched-cost comparisons separate the effect of multiple rows from the effect of organizing those rows around a shared plaintext base.
- We evaluate the allocation trade-off across two ciphertext budgets, multiple round settings, four lightweight ciphers, and two classifier families. Under these tested settings, the single-difference baseline is rarely the best allocation.
- We show that the best tested number of retained difference rows is not universal. Logistic regression often benefits from larger representations, while the MLP frequently prefers intermediate values because larger inputs can become sample-starved and overfit.
- We test robustness across several non-adaptive difference sets and show that the main trend is not tied to a single hand-picked set.

Overall, the results suggest that the number of differences per sample should be treated as a first-class design parameter in neural differential cryptanalysis. Richer representations can provide useful information, but only up to the point where the loss of independent plaintext bases begins to hurt generalization.

The implementation and training scripts are available at Github.

1.3 Paper organization

The rest of the paper is organized as follows. Section 2 introduces the neural distinguishing setting and formalizes the fixed-budget research question. Section 3 defines the input representations and their encryption costs. Section 4 describes the target ciphers, budgets, models, and training protocol. Section 5 presents the experimental results, including the main fixed-budget experiment and the robustness experiment over multiple difference sets. Section 6 discusses the observed trade-offs, their cipher-dependent behavior, and the limitations of the study. Finally, Section 7 concludes the paper.

2 Background and Problem Formulation

Neural differential distinguishers are trained to separate ciphertext samples generated from structured chosen-plaintext queries from samples in which the corresponding plaintext relation has been removed. In this work, we study this setting from a budget-allocation perspective. The total number of ciphertexts, or equivalently the total number of encryption queries, is kept fixed. The main question is then how this fixed resource should be spent: should we generate more independent plaintext bases for training, or should we spend more ciphertexts around each base plaintext in order to expose more input differences?

2.1 Neural distinguishing setting

Let $E_{K,r}$ denote an r-round reduced block cipher under key K. All target ciphers in this work have 64-bit blocks. A labeled training example is denoted by (X, y), where X is a ciphertext-derived binary representation and $y \in \{0, 1\}$ is the class label. We use $y = 1$ for real samples and $y = 0$ for random-reference samples.

In the real class, plaintexts are queried under prescribed XOR relations. In the simplest case, a plaintext pair has the form

$$(P, P \oplus \Delta),$$

and the model receives a representation derived from the corresponding ciphertexts. In the random-reference class, the same representation format is used, but the relation between plaintexts is broken by replacing the related plaintexts with independent random plaintexts. Thus, the classifier cannot exploit the input shape alone; it must learn statistical differences induced by the reduced-round cipher.

The classifier is trained as a binary predictor

$$f_\theta(X) \approx \Pr[y = 1 \mid X],$$

where f_θ is either a linear model or a small neural network in our experiments.

2.2 Plaintext bases versus input differences

The key trade-off in this paper is between the number of independent plaintext bases and the number of differences attached to each base. If we spend only a small number of ciphertexts per sample, we can generate many independent training examples. If we spend more ciphertexts per sample, each example can contain more differential information, but the number of independent examples decreases.

We refer to these two quantities as follows:

- *plaintext-base count*: the number of independent plaintext bases used to construct the dataset;
- *differential richness*: the number of ciphertext-difference rows included in each sample.

This distinction is important because a fixed-sample comparison does not control the cryptanalytic cost. A representation with more rows may appear stronger simply because it uses more ciphertexts per labeled example. We therefore use a fixed-budget comparison.

Let Q be the total encryption budget and let m be the number of encryptions used per labeled sample. The number of available labeled samples is then

$$N_s = \left\lfloor \frac{Q}{m} \right\rfloor.$$

Thus, increasing the number of ciphertexts per sample directly reduces the number of independent plaintext bases available for training.

2.3 Research question

The central research question is:

> Under a fixed ciphertext budget, is it better to train on more independent plaintext bases, or on fewer samples that contain more ciphertext differences?

Equivalently, we study whether the performance gain of richer multi-difference representations comes from useful information density, or whether it is offset by the loss of independent training examples.

2.4 Evaluation metrics

The primary metric is test accuracy on a held-out test set. Random guessing corresponds to an accuracy of 50%. We also report the empirical distinguishing advantage

$$\mathrm{Adv} = 2\,\mathrm{Acc} - 1,$$

so that random guessing has advantage zero.

Because increasing the number of differences per sample can reduce the number of independent training examples, we also track the train–test accuracy gap:

$$\mathrm{Gap} = \mathrm{Acc}_{\mathrm{train}} - \mathrm{Acc}_{\mathrm{test}} .$$

This quantity helps identify regimes where a richer representation becomes sample-starved and starts to overfit.

3 Budget-Aware Input Representations

This section defines the representations used in the experiments. All inputs are ciphertext-only. The model receives only bits derived from ciphertexts or ciphertext differences. Plaintexts are used only to generate the chosen-plaintext queries.

We use k to denote the number of effective ciphertext-difference rows given to the model. The case $k = 1$ corresponds to the usual single-difference baseline.

3.1 Single-difference baseline

The standard single-difference representation is built from one plaintext pair. For the real class, the pair is $(P, P \oplus \Delta)$ and the input is

$$X = \mathrm{bits}\left(E_{K,r}(P) \oplus E_{K,r}(P \oplus \Delta)\right) \in \{0,1\}^{1\times 64}.$$

For the random-reference class, the two plaintexts are independent, but the same ciphertext-difference encoding is used.

This baseline uses two encryptions per labeled sample. In our implementation, it is represented as the shared-base construction with $k = 1$.

3.2 Shared-base multi-difference representation

The shared-base representation spends more ciphertexts around a single plaintext base. Given an ordered difference set

$$D = \{\delta_0, \delta_1, \ldots, \delta_k\}, \qquad \delta_0 = 0,$$

we generate the related plaintexts

$$P \oplus \delta_0,\ P \oplus \delta_1,\ \ldots,\ P \oplus \delta_k.$$

The ciphertext corresponding to $\delta_0 = 0$ is used as the reference. The input matrix consists of the k row-wise ciphertext differences:

$$X_{\mathrm{shared}} = \begin{bmatrix} \mathrm{bits}(C_1 \oplus C_0) \\ \mathrm{bits}(C_2 \oplus C_0) \\ \vdots \\ \mathrm{bits}(C_k \oplus C_0) \end{bmatrix} \in \{0,1\}^{k\times 64},$$

where $C_i = E_{K,r}(P \oplus \delta_i)$.

For the random-reference class, the same matrix format is preserved, but $C_0, \ldots, C_k$ are generated from independent random plaintexts. Therefore, the plaintext relation is removed while the representation shape remains unchanged.

The shared-base representation uses $k+1$ encryptions per sample. Hence, larger k increases the number of differences observed around each plaintext base, but decreases the number of bases that can be sampled under a fixed budget.

Table 1: Input shape and encryption cost per labeled sample.

Representation	Input shape	Encryptions per sample
Single-difference baseline	1×64	2
Shared-base multi-difference	$k \times 64$	$k + 1$
Independent-pair control	$k \times 64$	$2k$

3.3 Independent-pair representation

The independent-pair representation is used as a control. Instead of building all rows around one plaintext base, it uses k independent plaintext pairs. For the real class, each row is generated from

$$(P_i, P_i \oplus \Delta), \qquad 1 \le i \le k.$$

The input matrix is

$$X_{\text{ind}} = \begin{bmatrix} \text{bits}(C'_1 \oplus C_1) \\ \text{bits}(C'_2 \oplus C_2) \\ \vdots \\ \text{bits}(C'_k \oplus C_k) \end{bmatrix} \in \{0,1\}^{k \times 64},$$

where $C_i = E_{K,r}(P_i)$ and $C'_i = E_{K,r}(P_i \oplus \Delta)$.

In the random-reference class, each row is generated from two independent random plaintexts. This representation therefore measures the effect of having multiple independent differential rows, without the shared plaintext base used in the previous construction.

The independent-pair representation uses $2k$ encryptions per sample. It is therefore more expensive per sample than the shared-base representation at the same value of k.

3.4 Cost comparison

Table 1 summarizes the input shape and encryption cost of the representations.

This table highlights why fixed-budget evaluation is necessary. At fixed sample count, larger k automatically consumes more ciphertexts. At fixed budget, however, increasing k forces the dataset to contain fewer labeled samples.

Some comparisons are made at the same value of k, while others are made at matched encryption cost. For example, shared-base with $k = 7$ and independent-pair with $k = 4$ both use eight encryptions per sample. Such matched-cost comparisons help isolate the role of the shared plaintext base.

3.5 Difference set for the main experiments

The main experiments use the following fixed ordered difference set:

$$D_A = \{\texttt{0x0}, \texttt{0x1}, \texttt{0x2}, \texttt{0x10}, \texttt{0x100000001}, \texttt{0x10008}, \texttt{0x33}, \texttt{0xF0F}\}.$$

For a given value of k, the first $k + 1$ elements of this set are used. Thus, $k = 1$ gives the single-difference baseline based on `0x1`, while $k = 7$ uses the full set.

We do not claim that this set is optimal for any of the target ciphers. It is used as a fixed non-adaptive set in the main experiments to avoid cipher-specific tuning. Robustness to alternative difference sets is considered separately.

4 Experimental Methodology

This section describes the target ciphers, budgets, models, and training protocol used in the main experiments.

4.1 Target ciphers and round settings

We evaluate four lightweight 64-bit block ciphers: GIFT-64, PRESENT-64, RECTANGLE-64, and SPECK-64/128. GIFT, PRESENT, and RECTANGLE are SPN-type ciphers, while SPECK is an ARX cipher. This selection allows us to compare the budget-allocation behavior across different design styles.

The main experiments use the following reduced-round settings:

- GIFT-64: 5 and 6 rounds;
- PRESENT-64: 5 and 6 rounds;
- RECTANGLE-64: 5 and 6 rounds;
- SPECK-64/128: 4 and 5 rounds.

These rounds were selected to avoid fully saturated easy cases while keeping the classification problem informative under the fixed training protocol.

4.2 Budgets and representation sizes

The main experiments use two ciphertext budgets:

$$Q \in \{2^{17}, 2^{19}\}.$$

For each budget, we evaluate

$$k \in \{1, 2, 3, 4, 7\}.$$

The value $k = 1$ is the single-difference baseline. Larger values allocate more ciphertexts to each plaintext base and therefore reduce the number of plaintext bases available for training.

The independent-pair representation is used as a control on RECTANGLE-64 and SPECK-64/128 at 5 rounds. This keeps the control experiment focused while still allowing a comparison between shared-base aggregation and independent-pair aggregation.

4.3 Models

We use two classifiers. The first is logistic regression on the flattened binary input. This model tests how much distinguishing information is already exposed linearly by the representation.

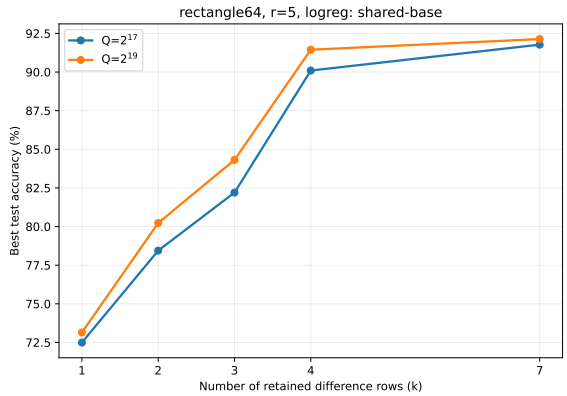
The second model is a multilayer perceptron with two hidden layers. It is a lightweight nonlinear classifier and can exploit interactions between bits and between rows. We intentionally keep the model set small because the purpose of the paper is not architecture search, but the study of how a fixed ciphertext budget should be allocated.

4.4 Training protocol

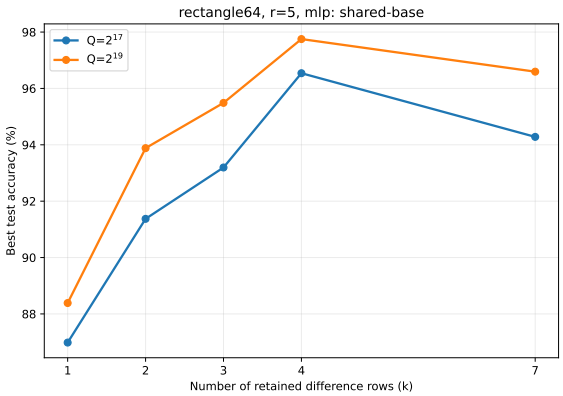
For each cipher, round, budget, representation, value of k, and model, we generate balanced datasets with equal numbers of real and random-reference samples. The number of samples is determined by the ciphertext budget and the cost of the representation.

Each experiment is repeated over five independent runs. In each run, a fresh master key and dataset seed are used. The dataset is split into training and test subsets using an 80/20 split. All models are trained for 20 epochs using Adam with learning rate 10^{-3} and batch size 128.

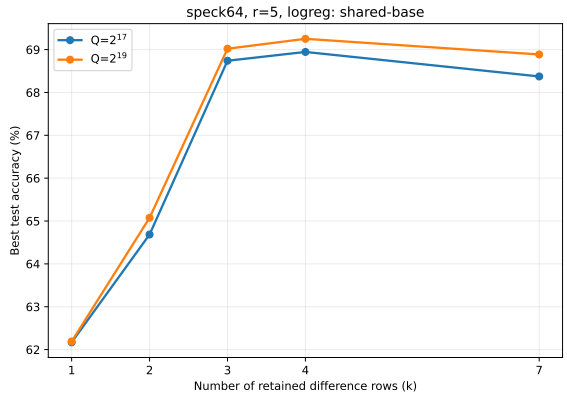
The main reported metric is mean test accuracy over the five runs. We also track the best value of k, the gain over $k = 1$, and the train–test gap. The latter is used to identify cases where richer representations become sample-starved under the fixed budget.



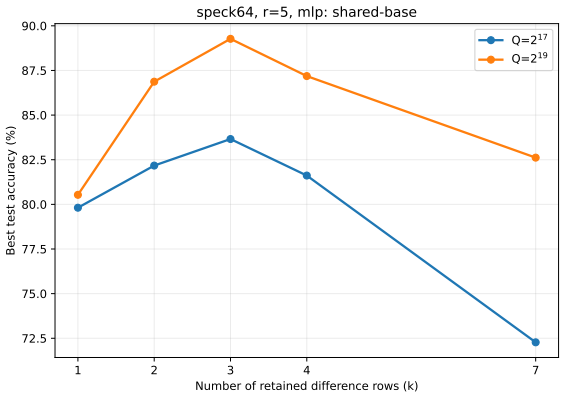
(a) RECTANGLE-64, LogReg



(b) RECTANGLE-64, MLP



(c) SPECK-64/128, LogReg



(d) SPECK-64/128, MLP

Figure 1: Shared-base accuracy as a function of the number of retained difference rows k at 5 rounds. RECTANGLE-64 gains strongly from increasing k, while SPECK-64/128 benefits from richer inputs but often prefers an intermediate value of k, especially with the MLP.

5 Experimental Results

This section presents the experimental results. The goal is to understand how a fixed ciphertext budget should be allocated between two competing choices: using more independent plaintext bases, or using more ciphertext differences per sample. We first present the main fixed-budget experiment using the primary difference set D_A. We then test whether the observed trend is robust to the choice of difference set.

5.1 Main fixed-budget experiment

The main experiment evaluates the shared-base representation on four ciphers and uses the independent-pair representation as a focused control on RECTANGLE-64 and SPECK-64/128. Unless otherwise stated, all reported accuracies are means over five independent runs.

5.1.1 Shared-base budget allocation

We first study the shared-base representation, where several ciphertext differences are generated around the same plaintext base. Figure 1 shows the effect of increasing the number of retained difference rows k for RECTANGLE-64 and SPECK-64/128 at 5 rounds.

The first observation is that $k = 1$ is rarely the best use of the fixed ciphertext budget. For RECTANGLE-64, increasing k produces a large and stable improvement. With logistic

Table 2: Shared-base representation: representative best results over $k \in \{1, 2, 3, 4, 7\}$. Accuracy values are means over five runs. Gains are absolute percentage-point improvements over $k = 1$.

Cipher	Model	Round	Budget	Acc. at $k = 1$	Best k	Best acc. / gain
GIFT-64	LogReg	5	2^{17}	66.7	7	74.8 / +8.2
GIFT-64	MLP	5	2^{17}	76.7	3	82.2 / +5.5
GIFT-64	LogReg	5	2^{19}	66.8	7	75.8 / +9.0
GIFT-64	MLP	5	2^{19}	78.7	3	86.9 / +8.2
GIFT-64	LogReg	6	2^{19}	55.6	3	59.0 / +3.4
GIFT-64	MLP	6	2^{19}	61.4	3	64.8 / +3.4
PRESENT-64	LogReg	5	2^{17}	62.4	7	70.2 / +7.8
PRESENT-64	MLP	5	2^{17}	73.0	7	75.8 / +2.7
PRESENT-64	LogReg	5	2^{19}	63.1	7	71.4 / +8.4
PRESENT-64	MLP	5	2^{19}	75.7	3	82.8 / +7.1
PRESENT-64	LogReg	6	2^{19}	53.2	7	55.3 / +2.1
PRESENT-64	MLP	6	2^{19}	56.5	2	56.6 / +0.0
RECTANGLE-64	LogReg	5	2^{17}	72.5	7	91.8 / +19.3
RECTANGLE-64	MLP	5	2^{17}	87.0	4	96.5 / +9.6
RECTANGLE-64	LogReg	5	2^{19}	73.1	7	92.1 / +19.0
RECTANGLE-64	MLP	5	2^{19}	88.4	4	97.8 / +9.4
RECTANGLE-64	LogReg	6	2^{19}	56.0	7	68.9 / +12.8
RECTANGLE-64	MLP	6	2^{19}	65.4	4	80.5 / +15.1
SPECK-64/128	LogReg	4	2^{17}	82.9	4	92.9 / +10.0
SPECK-64/128	MLP	4	2^{17}	96.5	3	98.5 / +2.1
SPECK-64/128	LogReg	5	2^{17}	62.2	4	68.9 / +6.8
SPECK-64/128	MLP	5	2^{17}	79.8	3	83.7 / +3.8
SPECK-64/128	LogReg	5	2^{19}	62.2	4	69.3 / +7.1
SPECK-64/128	MLP	5	2^{19}	80.5	3	89.3 / +8.7

regression, the best result is usually obtained at the largest tested value, $k = 7$. With the MLP, the best result is typically obtained at an intermediate value, especially $k = 4$. This shows that additional difference rows expose useful information, but also that the largest representation is not always best among the tested settings once the number of plaintext bases is reduced.

SPECK-64/128 behaves differently. The shared-base representation still improves over the single-difference case, but the best tested value is more often moderate, such as $k = 3$ or $k = 4$. This suggests that for SPECK, spending too much of the budget inside each sample can reduce the number of independent plaintext bases too aggressively.

Table 2 summarizes representative shared-base results. For each setting, it reports the accuracy at $k = 1$, the best value of k, the best accuracy, and the gain over $k = 1$.

The table confirms the trend seen in Figure 1. The single-difference baseline is usually not best among the tested settings. At the same time, the best value of k is not fixed. Logistic regression often benefits from larger values of k, while the MLP frequently performs best at an intermediate value. This indicates that the best observed allocation depends not only on the cipher but also on the capacity of the model.

Table 3: Shared-base versus independent-pair control at 5 rounds. Accuracy values are means over five runs.

Cipher	Model	Budget	Best shared-base	Best independent-pair
RECTANGLE-64	LogReg	2^{17}	91.8	91.1
RECTANGLE-64	MLP	2^{17}	96.5	93.3
RECTANGLE-64	LogReg	2^{19}	92.1	91.4
RECTANGLE-64	MLP	2^{19}	97.8	96.4
SPECK-64/128	LogReg	2^{17}	68.9	72.9
SPECK-64/128	MLP	2^{17}	83.7	80.8
SPECK-64/128	LogReg	2^{19}	69.3	74.2
SPECK-64/128	MLP	2^{19}	89.3	86.8

5.1.2 Empirical variation across ciphers

The shared-base gains are not uniform across the target ciphers. RECTANGLE-64 shows the strongest improvement. At 5 rounds, the logistic regression distinguisher improves by about 19 percentage points over the $k = 1$ baseline, and the MLP also gains substantially. At 6 rounds, where the baseline is much weaker, the multi-difference representation still recovers a significant distinguishing signal.

GIFT-64 and PRESENT-64 also benefit from increasing k, although the gains are smaller in the harder 6-round settings. This indicates that richer multi-difference inputs are useful beyond a single cipher, but the size of the benefit depends on how much exploitable signal remains at the selected round depth.

SPECK-64/128 shows a different empirical pattern. Richer shared-base inputs improve over $k = 1$, but the best value is usually moderate rather than maximal. In particular, $k = 3$ or $k = 4$ often performs better than $k = 7$. We therefore treat SPECK as evidence that the best observed fixed-budget allocation can differ substantially from the SPN cases, rather than as evidence for a universal ARX trend.

5.1.3 Effect of increasing the ciphertext budget

Increasing the budget from 2^{17} to 2^{19} generally improves performance or makes richer representations more stable. This is most visible for the MLP. For example, on SPECK-64/128 at 5 rounds, the best shared-base MLP accuracy increases from about 83.7% at budget 2^{17} to about 89.3% at budget 2^{19}. Similarly, for RECTANGLE-64 at 6 rounds, the best MLP accuracy increases from about 75.5% to about 80.5%.

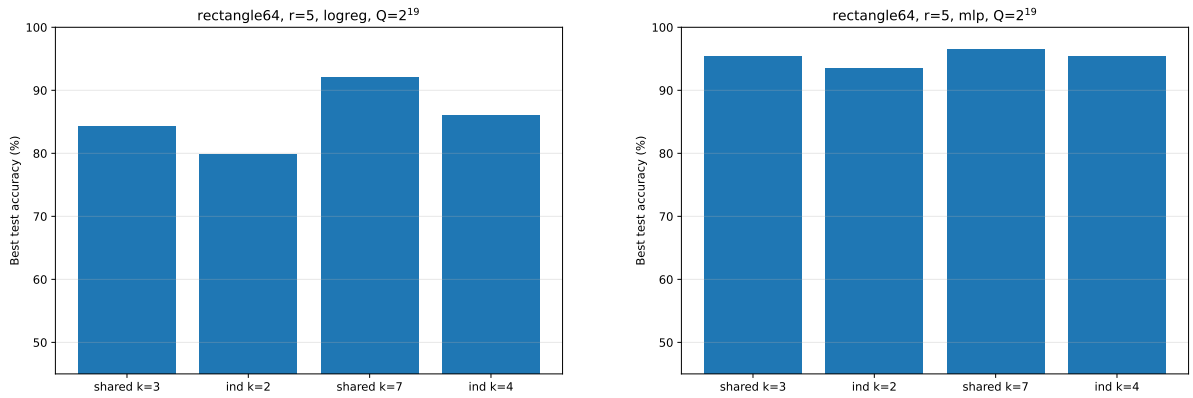
This is consistent with the fixed-budget trade-off. Larger k values produce richer samples, but fewer plaintext bases. When the total ciphertext budget is larger, the model can afford richer samples while still seeing enough independent plaintext bases to generalize.

5.1.4 Shared-base versus independent-pair control

The independent-pair control was evaluated at 5 rounds for RECTANGLE-64 and SPECK-64/128. This experiment asks whether the improvement comes only from having multiple ciphertext-difference rows, or whether the common plaintext base in the shared-base representation also provides useful structure.

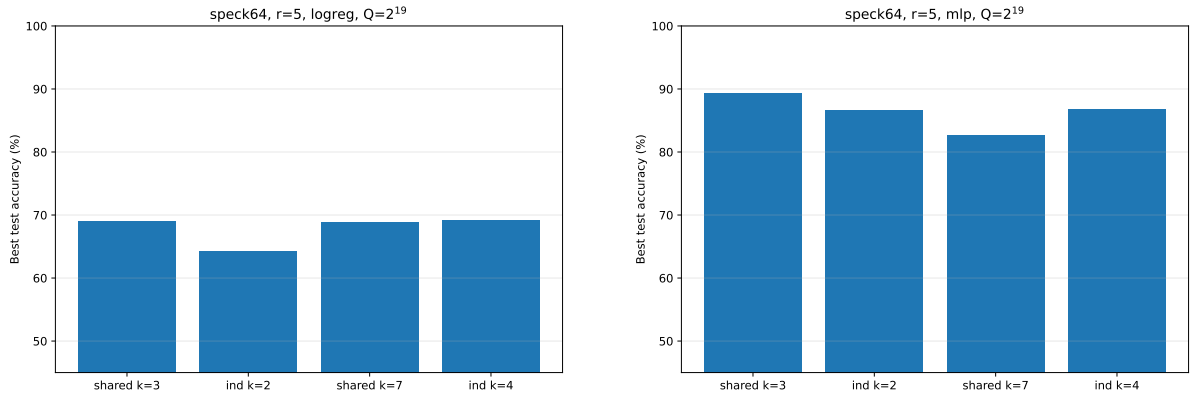
Table 3 compares the best shared-base result with the best independent-pair result for the same cipher, budget, and model.

For RECTANGLE-64, the shared-base representation is consistently competitive and usually better, especially for the MLP. This suggests that the shared plaintext base contributes useful



(a) LogReg (b) MLP

Figure 2: Matched-cost comparison between shared-base and independent-pair representations on RECTANGLE-64 at 5 rounds. The shared-base representation remains competitive or better, suggesting that the common plaintext base provides useful structure beyond simply adding more rows.



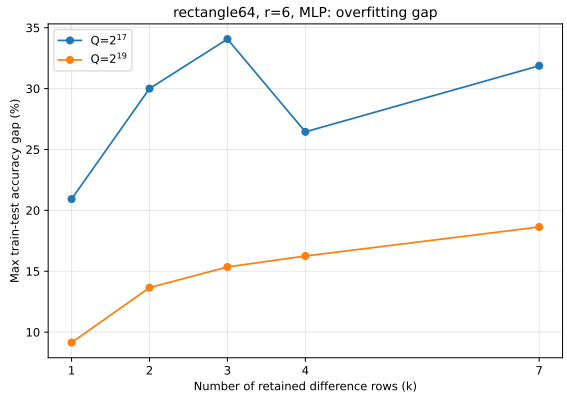
(a) LogReg (b) MLP

Figure 3: Matched-cost comparison between shared-base and independent-pair representations on SPECK-64/128 at 5 rounds. In contrast to RECTANGLE-64, the preferred representation depends more strongly on the model.

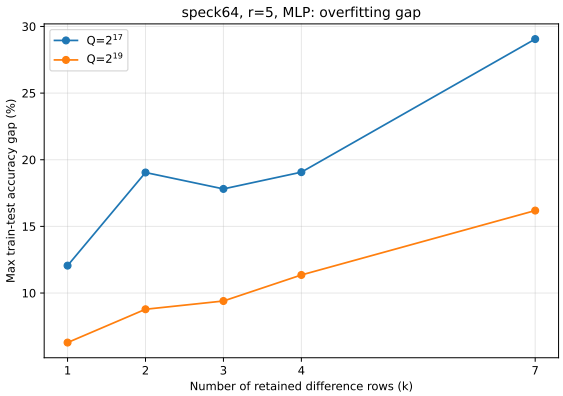
structure beyond simply increasing the number of rows. Figure 2 shows the matched-cost comparison for RECTANGLE-64 at the larger budget.

For SPECK-64/128, the picture is more mixed. Logistic regression performs better with independent pairs, while the MLP performs better with shared-base inputs. This reinforces the conclusion that the best observed allocation is not universal and depends on both the cipher and the model. Figure 3 shows the matched-cost comparison for SPECK-64/128.

The matched-cost comparisons are particularly important. Shared-base with $k = 7$ and independent-pair with $k = 4$ both use eight encryptions per sample. Thus, the comparison is not simply between a cheaper and a more expensive input. It compares two different ways of spending the same per-sample ciphertext cost: around one plaintext base, or across several independent plaintext bases.



(a) RECTANGLE-64, 6 rounds



(b) SPECK-64/128, 5 rounds

Figure 4: Train–test accuracy gap for the MLP as k increases. Larger values of k provide richer inputs but reduce the number of independent plaintext bases under a fixed ciphertext budget. This sample-starvation effect helps explain why the largest representation is not always best among the tested settings.

5.1.5 Overfitting and sample starvation

A consistent pattern in the results is that increasing k can improve the information content of each sample but can also increase the risk of overfitting. This is especially visible for the MLP in harder round settings.

Figure 4 shows the train–test accuracy gap for the MLP on RECTANGLE-64 at 6 rounds and SPECK-64/128 at 5 rounds. These are representative settings where the tension between differential richness and plaintext-base count is visible.

The figure supports the interpretation that larger representations can become sample-starved. A larger k gives the model more information inside each sample, but it also reduces the number of independent plaintext bases used for training. A more expressive model such as the MLP can exploit the richer representation, but it may also fit run-specific noise when the number of independent examples becomes too small. This explains why intermediate values such as $k = 3$ or $k = 4$ often outperform $k = 7$.

5.1.6 Summary of the main experiment

The main fixed-budget experiment leads to the following findings.

- Under a fixed ciphertext budget, the single-difference baseline ($k = 1$) is rarely the best allocation.
- Shared-base multi-difference inputs substantially improve performance for several ciphers, especially RECTANGLE-64.
- The best value of k is not universal. Logistic regression often benefits from larger k, while the MLP frequently prefers intermediate values.
- SPECK-64/128 also benefits from richer inputs, but its best tested value is more sensitive than in the SPN cases.
- Independent-pair controls show that the shared plaintext base can be useful, but its benefit is cipher- and model-dependent.
- Overfitting and sample starvation explain why the largest value of k is not always the best use of the fixed ciphertext budget.

Table 4: Robustness to difference-set choice. The gain range reports the absolute percentage-point improvement of the best $k \in \{2, 3, 4, 7\}$ over $k = 1$, across D_A, D_B, and D_C.

Cipher	Model	Gain range over $k = 1$	Observed best k values
GIFT-64	LogReg	+3.6 to +15.1	7
GIFT-64	MLP	+1.1 to +8.7	2, 3, 7
PRESENT-64	LogReg	+1.4 to +10.8	7
PRESENT-64	MLP	+2.0 to +5.0	2
RECTANGLE-64	LogReg	+13.6 to +36.4	7
RECTANGLE-64	MLP	+7.4 to +37.4	4, 7
SPECK-64/128	LogReg	+8.1 to +24.0	7
SPECK-64/128	MLP	+5.8 to +25.1	3, 4

5.2 Robustness to difference-set choice

The main experiment used a fixed ordered difference set D_A. To check whether the observed budget-allocation trend is tied to this particular choice, we repeat the shared-base experiment with two additional non-adaptive difference sets. The purpose of this experiment is not to optimize the differences for each cipher, but to test whether the main conclusion remains visible when the difference set changes.

The robustness experiment uses one representative round for each cipher, namely 5 rounds for GIFT-64, PRESENT-64, RECTANGLE-64, and SPECK-64/128. We use a medium ciphertext budget $Q = 2^{18}$, the same values $k \in \{1, 2, 3, 4, 7\}$, and three independent runs per setting. Only the shared-base representation is considered in this experiment, since the goal is to isolate the sensitivity to the choice of input differences.

In addition to the main set D_A, we use the following two ordered sets:

$$D_B = \{\texttt{0x0}, \texttt{0xF}, \texttt{0x80}, \texttt{0x8000}, \\ \texttt{0x80000000}, \texttt{0x1000000000000}, \\ \texttt{0x8080}, \texttt{0x0100000000000001}\},$$

and

$$D_C = \{\texttt{0x0}, \texttt{0x4}, \texttt{0x40}, \texttt{0x400}, \\ \texttt{0x4000}, \texttt{0x00000000F0000000}, \\ \texttt{0x0000F00000000000}, \texttt{0x8000000000000001}\}.$$

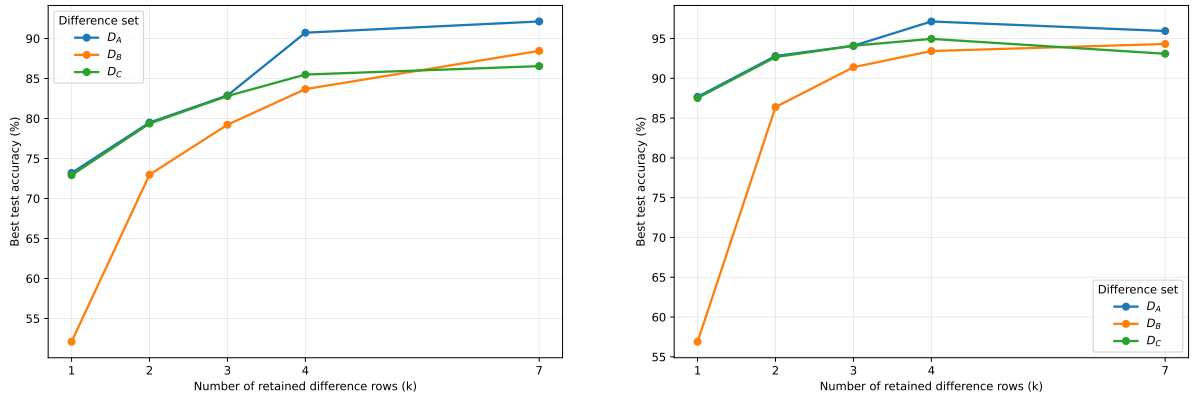
For each value of k, the first $k + 1$ elements of the corresponding ordered set are used.

Table 4 summarizes the gain of the best multi-difference setting over the $k = 1$ baseline. The table reports, for each cipher and model, the range of gains observed over the three difference sets and the set of best k values that occurred.

Figure 5 gives a more detailed view for RECTANGLE-64. It shows that, for both models, the $k = 1$ case is not the best allocation across the three tested difference sets.

The results strengthen the main conclusion. Across all 24 combinations of difference set, cipher, and model, the single-difference case $k = 1$ is never the best fixed-budget allocation. This suggests that the advantage of using several difference rows is not merely an artifact of the original set D_A.

At the same time, the experiment shows that the exact magnitude of the gain and the best value of k depend on the difference set. This is expected: different input differences expose different amounts of reduced-round signal. Some sets have a weak $k = 1$ baseline but recover a strong distinguisher when several rows are combined. This behavior is especially visible for RECTANGLE-64 and SPECK-64/128 under D_B.



(a) LogReg (b) MLP

Figure 5: Robustness to the choice of difference set on RECTANGLE-64 at 5 rounds using the shared-base representation. Across the three non-adaptive difference sets, the single-difference baseline $k = 1$ is not the best fixed-budget allocation.

These findings support a more precise interpretation of the paper's claim. We do not claim that a single difference set or a single value of k is optimal for all ciphers. Rather, the robust observation is that, under a fixed ciphertext budget, allocating all ciphertexts to independent single-difference samples is often not the best tested allocation. Multiple difference rows can provide useful information even when the best value of k varies with the cipher, the model, and the chosen difference set.

5.3 Overall summary

The experiments support the following overall conclusions. First, under a fixed ciphertext budget, using only single-difference samples is rarely the best allocation. Second, adding more ciphertext-difference rows can significantly improve neural distinguishers, but the best tested number of rows is not universal. Third, the trade-off depends on the cipher, the model, the budget, and the chosen difference set. Finally, richer representations can become sample-starved, especially for the MLP, which explains why intermediate values of k often outperform the largest tested representation.

6 Analysis and Discussion

The experiments show that the single-difference baseline is rarely the best fixed-budget allocation. However, they also show that increasing the number of difference rows is not always sufficient by itself. The best value of k depends on the cipher, the model, the budget, and the difference set. This section discusses the main reasons behind these observations.

6.1 Information density versus plaintext-base count

The central trade-off in this work is between information density and plaintext-base count. Increasing k gives each labeled sample more ciphertext differences and therefore exposes a richer view of the cipher's reduced-round differential behavior. At the same time, under a fixed ciphertext budget, a larger k reduces the number of independent plaintext bases available for training.

This explains why the best allocation is usually not at $k = 1$, but also not always at the largest tested value. When $k = 1$, each sample contains only one ciphertext difference. This can be insufficient when the single selected input difference exposes only a weak signal. Adding

more rows can combine several weak pieces of evidence into a more useful representation. This effect is especially clear in the robustness experiment, where some difference sets have a weak single-difference baseline but recover strong distinguishers when multiple rows are combined.

On the other hand, very large k values can become sample-starved. In that regime, the model receives richer individual samples but sees fewer independent plaintext bases. This can hurt generalization, especially for models with enough capacity to fit accidental patterns in the training data. Thus, the observed performance is governed by two competing effects: richer samples improve the available signal, while fewer plaintext bases increase the risk of overfitting.

6.2 Why logistic regression and MLP prefer different allocations

A consistent trend in the experiments is that logistic regression often benefits from larger values of k, while the MLP frequently prefers intermediate values such as $k = 3$ or $k = 4$. This difference is important because it shows that the best budget allocation is model-dependent.

Logistic regression has limited capacity. It cannot easily learn complex nonlinear interactions between rows, but it can benefit from having more linearly useful features. Adding more ciphertext-difference rows increases the number of such features. Therefore, for logistic regression, the largest tested value of k is often beneficial, especially in the SPN cases.

The MLP has higher capacity and can exploit nonlinear interactions between bits and between rows. This makes it stronger than logistic regression in many settings, but also more sensitive to the reduction in independent training samples. When k becomes too large, the MLP may fit the training data well but generalize worse. This explains why the MLP often achieves its best test accuracy at an intermediate value of k rather than at $k = 7$.

This observation is useful in practice. It suggests that representation design and model selection should not be treated independently. A representation that performs best for a linear distinguisher may not perform best for a nonlinear one, because the two models use the fixed ciphertext budget differently.

6.3 Role of the shared plaintext base

The independent-pair control experiment helps separate two possible explanations for the gain of multi-row inputs. One possibility is that the model only benefits from seeing multiple ciphertext-difference rows. Another possibility is that the specific organization of those rows around a shared plaintext base carries additional useful structure.

The results suggest that both effects can matter. For RECTANGLE-64, the shared-base representation is consistently competitive and usually better than the independent-pair control, especially for the MLP. This indicates that the common plaintext base contributes useful information beyond simply increasing the number of rows. In other words, the model appears to benefit from observing several related differences around the same local plaintext neighborhood.

For SPECK-64/128, the behavior is more mixed. Logistic regression performs better with independent pairs, while the MLP performs better with shared-base inputs. This suggests that the usefulness of shared-base structure depends on both the cipher and the model. It is not a universal advantage, but it can be valuable when the model is able to exploit the dependencies between rows.

6.4 Interpreting cipher-dependent trends

The strongest gains are observed for RECTANGLE-64. This suggests that, for this cipher, several related input differences expose complementary information that is not captured well by a single difference. The effect remains visible even in the harder 6-round setting, where the single-difference baseline is much weaker.

GIFT-64 and PRESENT-64 also benefit from multi-difference inputs, although the gains are smaller in the harder round settings. This indicates that the shared-base representation is useful across several SPN-type ciphers, but the magnitude of the benefit depends on the remaining differential signal at the chosen number of rounds.

SPECK-64/128 behaves differently. It still benefits from richer inputs, but the best value of k is usually moderate rather than maximal. This is consistent with the idea that the signal available to the model and the way it spreads through the cipher are structure-dependent. Since SPECK is the only ARX cipher in our experiments, we do not claim that this behavior generalizes to all ARX designs. The safer conclusion is that the best observed budget allocation can differ substantially between cipher families, and that SPECK is more sensitive to the choice of representation and model than the SPN cases studied here.

6.5 Difference-set sensitivity

The robustness experiment confirms that the main conclusion is not tied to a single hand-picked difference set. Across the tested difference sets, the single-difference case $k = 1$ is never the best fixed-budget allocation. This supports the claim that using several difference rows can be a better use of the ciphertext budget than spending the entire budget on independent single-difference samples.

At the same time, the robustness experiment also shows that the exact magnitude of the gain and the best value of k depend on the chosen difference set. This is expected. Different input differences may expose different amounts of reduced-round signal. Some individual differences may be weak on their own, but several such differences can still provide useful combined evidence when placed in a multi-row representation.

Therefore, the conclusion should not be interpreted as saying that any multi-difference set is good, or that a particular value of k is universally best. Rather, the results show that the allocation between plaintext-base count and differential richness is itself an important experimental and cryptanalytic design choice.

6.6 Implications for neural differential cryptanalysis

The results have several implications for the design of neural distinguishers.

First, fixed-sample comparisons can be misleading. If two representations use different numbers of ciphertexts per sample, comparing them at the same number of samples gives a larger encryption budget to the richer representation. A fixed-budget comparison is therefore necessary when the goal is to evaluate cryptanalytic data efficiency.

Second, the single-difference baseline should not be assumed to be the natural or best input representation. In many settings, using more difference rows improves performance even though it reduces the number of training samples. This suggests that future neural distinguisher designs should treat the number of differences per sample as a tunable design parameter.

Third, richer representations must be evaluated together with generalization. A high-capacity model may exploit additional rows, but it may also overfit when the number of independent plaintext bases becomes too small. Reporting only test accuracy is not sufficient to understand this trade-off; train–test gap and budget-normalized comparisons are also important.

Finally, the best allocation is not universal. It depends on the cipher, the round, the model, and the difference set. This is not a weakness of the approach; rather, it is the main finding of the paper. Neural differential distinguishers should be designed under an explicit budget model, and the allocation of that budget should be treated as part of the cryptanalytic experiment.

6.7 Limitations

This study has several limitations. First, the experiments focus on four 64-bit-block ciphers, with SPECK-64/128 as the only ARX representative. Therefore, the results should not be interpreted as a general statement about all ARX ciphers. Adding further ARX designs, such as SIMON or other lightweight ARX ciphers, would provide a stronger basis for broader conclusions.

Second, the difference sets used in the experiments are fixed and non-adaptive. This is intentional: the goal is to study budget allocation without tuning the differences separately for each cipher. However, a cipher-specific search for difference sets may produce stronger distinguishers and may shift the best-performing value of k.

Third, the model set is deliberately small. Logistic regression and a compact MLP are sufficient to study the interaction between representation richness and model capacity, but larger architectures may change the quantitative results. The qualitative trade-off should still remain: increasing k enriches each sample while reducing the number of independent plaintext bases.

Fourth, k is evaluated only over the discrete set $\{1, 2, 3, 4, 7\}$. Consequently, the reported best values are best among the tested configurations and should not be interpreted as globally optimal budget allocations. Adaptive or continuous budget-allocation strategies are left for future work.

Finally, this work focuses on distinguishing accuracy rather than full key recovery. A natural next step is to study whether the same fixed-budget allocation principles carry over to neural-assisted key recovery attacks.

7 Conclusion

This paper studied neural differential distinguishers from a fixed-budget perspective. Instead of comparing input representations at a fixed number of labeled samples, we fixed the total number of ciphertexts, or equivalently the total number of encryption queries. This allowed us to study a practical allocation question: should the available budget be spent on more independent plaintext bases, or on more ciphertext-difference rows inside each sample?

We introduced a shared-base multi-difference representation and compared it with the usual single-difference baseline and with an independent-pair control representation. The experiments on GIFT-64, PRESENT-64, RECTANGLE-64, and SPECK-64/128 show that the single-difference baseline is rarely the best fixed-budget allocation. Adding more difference rows often improves the distinguisher, even though it reduces the number of independent training samples. This demonstrates that differential richness can be a better use of the ciphertext budget than simply generating more single-difference samples.

At the same time, the results show that the largest representation is not always the best among those tested. The best value of k depends on the cipher, the model, the budget, and the chosen difference set. Logistic regression often benefits from larger values of k, while the MLP frequently prefers intermediate values. This behavior is explained by the trade-off between information density and plaintext-base count: larger inputs provide more differential evidence per sample, but they can also become sample-starved and overfit when the number of independent plaintext bases becomes too small.

The independent-pair control experiments further show that the organization of the rows matters. For RECTANGLE-64, the shared-base representation is consistently competitive and often better than the independent-pair control, suggesting that several related differences around the same plaintext base can provide useful structure. For SPECK-64/128, the behavior is more model-dependent, which supports the broader conclusion that budget allocation should be measured rather than assumed.

We also tested the robustness of the main trend with several non-adaptive difference sets. Although the exact gain and the best value of k change with the difference set, the single-difference case was never the best allocation in the tested robustness settings. This supports the conclusion that the observed benefit of multi-difference representations is not merely an artifact of a single hand-picked difference set.

Overall, the results suggest that the number of difference rows per sample should be treated as an explicit design parameter in neural differential cryptanalysis. Fixed-sample comparisons can hide the true data cost of richer representations, while fixed-budget comparisons reveal the trade-off between differential richness and independent plaintext-base count. Future work can extend this study in several directions, including additional ARX ciphers, cipher-specific or automated difference-set selection, larger neural architectures, and neural-assisted key recovery attacks under the same fixed-budget viewpoint.

References

[1] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present: Towards reaching the limit of lightweight encryption. In *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.

[2] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Enhancing differential-neural cryptanalysis. In *Advances in Cryptology – ASIACRYPT 2022*, volume 13791 of *Lecture Notes in Computer Science*, pages 318–347. Springer, 2022.

[3] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference*, pages 175:1–175:6. ACM, 2015.

[4] Emanuele Bellini, Rocco Brunelli, David Gérault, Anna Hambitzer, and Marco Pedicini. Generic partial decryption as feature engineering for neural distinguishers. In *Progress in Cryptology – LATINCRYPT 2025*, volume 16129 of *Lecture Notes in Computer Science*, pages 373–398. Springer, 2026.

[5] Emanuele Bellini, David Gerault, Anna Hambitzer, and Matteo Rossi. A cipher-agnostic neural training pipeline with automated finding of good input differences. *IACR Transactions on Symmetric Cryptology*, 2023(3):184–212, 2023.

[6] Adrien Benamira, David Gerault, Thomas Peyrin, and Quan Quan Tan. A deeper look at machine learning-based cryptanalysis. In *Advances in Cryptology – EUROCRYPT 2021*, volume 12696 of *Lecture Notes in Computer Science*, pages 805–835. Springer, 2021.

[7] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[8] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[9] Yi Chen, Yantian Shen, Hongbo Yu, and Sitong Yuan. A new neural distinguisher considering features derived from multiple ciphertext pairs. *The Computer Journal*, 66(6):1419–1433, 2023.

[10] Valérie Gauthier-Umaña, Isabella Martínez, Germán Obando, and Juan F. Pérez. Key recovery attack on PRESENT using an entropy-based neural distinguisher. *Neural Computing and Applications*, 38:305, 2026.

[11] David Gérault, Anna Hambitzer, Moritz Huppert, and Stjepan Picek. Survey: Six years of neural differential cryptanalysis. *IACR Communications in Cryptology*, 3(2), 2026.

[12] Alireza Gholizadeh Shahrbejari and Reza Ebrahimi Atani. Representation vs. architecture in neural differential distinguishers. *IACR Communications in Cryptology*, 3(2), 2026.

[13] Aron Gohr. Improving attacks on round-reduced SPECK32/64 using deep learning. In *Advances in Cryptology – CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 150–179. Springer, 2019.

[14] Aron Gohr, Gregor Leander, and Patrick Neumann. An assessment of differential-neural distinguishers. *Cryptology ePrint Archive*, 2022. Report 2022/1521.

[15] Lulu Guo, Ming Duan, and Yuefei Zhu. Improved differential neural distinguishers for SHA-3-256 and Ascon-Hash256. *Electronics*, 15(18):4142, 2026.

[16] Yufei Hou, Jie Liu, Shouxu Han, Zhongjun Ma, Xi Ye, and Xuan Nie. Improving deep learning-based neural distinguisher with multiple ciphertext pairs for SPECK and SIMON. *Scientific Reports*, 15:13696, 2025.

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

[18] Jiashuo Liu, Manman Li, Jiongjiong Ren, and Shaozhen Chen. A highly efficient neural distinguisher framework for IoT-friendly lightweight SPN block ciphers. *IEICE Transactions on Information and Systems*, E109-D(2):238–248, 2026.

[19] Jiashuo Liu, Jiongjiong Ren, and Shaozhen Chen. A deep learning aided differential distinguisher improvement framework with more lightweight and universality. *Cybersecurity*, 6:47, 2023.

[20] 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.

[21] Iman Mirzaali, Sadegh Sadeghi, and Nasour Bagheri. Improved polytopic differential neural distinguishers for SIMON, SIMECK, and SPECK block ciphers. *Cybersecurity*, 9:70, 2026.

[22] Wenhuan Ren, Tianyuan Liu, Ting Cui, Linyuan Wang, Libin Hou, Jian Chen, and Bin Yan. A neural distinguisher based on neural architecture search for information entropy maximization. *Applied Soft Computing*, 202:115832, 2026.

[23] Byoungjin Seok and Changhoon Lee. A novel approach to construct a good dataset for differential-neural cryptanalysis. *IEEE Transactions on Dependable and Secure Computing*, 22(1):246–262, 2025.

[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.

[25] Tao Sun, Dongsu Shen, Saiqin Long, Qingyong Deng, and Shiguo Wang. Neural distinguishers on TinyJAMBU-128 and GIFT-64. In *Neural Information Processing – ICONIP 2022*, volume 1792 of *Communications in Computer and Information Science*, pages 419–431. Springer, 2022.

[26] Gao Wang, Gaoli Wang, and Siwei Sun. Investigating and enhancing the neural distinguisher for differential cryptanalysis. *IEICE Transactions on Information and Systems*, E107-D(8):1016–1028, 2024.

[27] Haiyi Xu, Lei Zhang, and Yufei Yuan. A novel evaluation algorithm for identifying optimal datasets in differential-based neural distinguisher. *IEEE Transactions on Information Forensics and Security*, 20:12328–12341, 2025.

[28] Tarun Yadav and Manoj Kumar. Differential-ML distinguisher: Machine learning based generic extension for differential cryptanalysis. In *Progress in Cryptology – LATINCRYPT 2021*, Lecture Notes in Computer Science, pages 191–212. Springer, 2021.

[29] Xue Yuan and Qichun Wang. A multi-differential approach to enhance related-key neural distinguishers. *The Computer Journal*, page bxag073, 2026.

[30] Xiaoteng Yue and Wanqing Wu. Improved neural differential distinguisher model for lightweight cipher SPECK. *Applied Sciences*, 13(12):6994, 2023.

[31] Liu Zhang, Jinyu Lu, Zilong Wang, and Chao Li. Improved differential-neural cryptanalysis for round-reduced SIMECK32/64. *Frontiers of Computer Science*, 17(6):176817, 2023.

[32] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, 2015.