

On the Mismatch between Neural-Discovered Differential-Linear Features and Long-Round Distinguisher Construction

Thomas Peyrin¹, Zilong Wang², Liu Zhang¹ and Chenlu Zheng²

¹ Nanyang Technological University, Singapore, Singapore

² Xidian University, Xi'an, China

Abstract

To the best of our knowledge, existing differential-neural cryptanalysis have not yet shown a clear round advantage over the strongest comparable classical analyses. Recent Fourier-based interpretability results show that, under a difference-only representation, features extracted from differential-neural distinguishers can be interpreted as classical differential-linear masks. This suggests a possible route toward longer-round classical cryptanalysis and motivates our question: can such neural-discovered masks serve as useful candidates in the search for long-round differential-linear distinguishers of ARX ciphers?

As a prerequisite to the long-round study, we first characterize the short-round differential-linear candidates exposed by difference-only differential-neural distinguishers. We introduce CONV1DFULLY to facilitate mask-level analysis by removing the residual tower and reorganizing the first convolution along the ciphertext-difference bit dimension. On **Speck32/64**, the dominant differential-linear feature remains preserved after these modifications. On **SipHash**, we compare Fourier masks extracted from trained distinguishers with an exhaustive evaluation of a low-Hamming-weight output-mask space. The neural-extracted masks are concentrated among high-correlation differential-linear approximations, including several of the strongest candidates examined. These experiments provide a controlled basis for treating neural-extracted masks as candidates in the subsequent long-round analysis.

We then examine their utility in the known 18-round **Speck128/128** distinguisher with a $5 + 8 + 5$ decomposition. Under the same middle input difference, an 8-round difference-only differential-neural distinguisher recurrently exposes several masks with substantially stronger local middle correlations than the classically selected mask. However, after 5-round single XOR-linear extensions, these masks yield considerably weaker overall 18-round correlations. We further impose sparsity guidance on the first convolutional layer to promote low-Hamming-weight candidates. Under this guidance, the intermediate mask used in the classical 18-round distinguisher is recovered in the first-layer candidate set in 9 of 30 independent runs, showing that the neural model can reproduce a long-round-useful classical candidate. Nevertheless, this recovery is not stable, and the final neural decision rule still favors locally stronger features rather than the classically selected mask. These results indicate that differential-neural distinguishers can assist long-round candidate generation, while reliable recovery and long-round-aware prioritization remain unresolved.

Keywords: differential-neural cryptanalysis, differential-linear cryptanalysis, fourier analysis, mask-oriented convolution, sparsity-guided learning

Emails: thomas.peyrin@ntu.edu.sg (Thomas Peyrin), zlwang@xidian.edu.cn (Zilong Wang), liu.zhang@ntu.edu.sg (Liu Zhang), clzheng@stu.xidian.edu.cn (Chenlu Zheng)

Date of this document: 2026-09-02.

1 Introduction

Since the work of Gohr at CRYPTO 2019, differential-neural cryptanalysis has attracted considerable attention as a data-driven approach to reduced-round cryptanalysis. In Gohr’s framework, a neural network is trained to distinguish ciphertext pairs generated from a fixed input difference from random pairs, and the resulting differential-neural distinguisher is further used as a scoring component in a key-recovery attack on reduced-round Speck32/64 [Goh19]. Subsequent work has extended this paradigm to improved distinguishing and key-recovery frameworks and has also sought to understand the source of the advantage exhibited by differential-neural distinguishers. Benamira et al. [BGPT21] analyzed the internal processing of Gohr’s distinguisher and related its behavior to differential and differential-linear information. At ASIACRYPT 2023, Bao et al. [BLYZ23] further investigated why differential-neural distinguishers can outperform conventional DDT-based distinguishers in reduced-round settings, showing that the neural model can exploit additional relations involving ciphertext differences and intermediate-state differences. More recently, Yuan et al. [YWZW26] introduced a Fourier-based framework for analyzing the Boolean decision rules induced by trained differential-neural distinguishers. For a difference-only input representation, the extracted parity terms admit a direct interpretation as classical equal-mask differential-linear features, and their experiments further showed that trained distinguishers can expose multiple strong classical differential-linear masks, including multi-bit masks. These developments suggest that a differential-neural distinguisher may be viewed not only as a classifier, but also as a data-driven source of classical cryptanalytic features.

A recent success in integral-neural cryptanalysis further supports this perspective. Zhang et al. [ZYS⁺26] used neural networks to identify integral-related features and fed the resulting observations back into classical integral cryptanalysis, leading to improved integral distinguishers and attacks with extended round bounds. Their work shows that neural feature discovery can contribute beyond the neural classifier itself when the learned information can be translated into a form suitable for automated classical search. This motivation is particularly relevant to differential-linear cryptanalysis, where selecting useful intermediate masks remains a non-trivial structured search problem. Chen et al. [CBY23] proposed a semi-unconstrained framework based on strong unbalanced bits, iterative combination, and meet-in-the-middle procedures, while Gong et al. [GWH⁺25] later exploited hourglass-like structural regularities to guide more efficient search. These works illustrate that practical intermediate-mask search relies on carefully designed heuristics, filtering strategies, or structural restrictions to navigate a large candidate space.

To the best of our knowledge, existing differential-neural attacks have not yet demonstrated a clear round advantage over the strongest comparable classical analyses. This leaves open whether the differential-linear features exposed by difference-only differential-neural distinguishers can be translated into useful candidates for long-round search. We therefore ask whether neural-discovered masks can serve as effective intermediate-mask candidates in the search for long-round differential-linear distinguishers. This question is non-trivial because the objective of a fixed-round differential-neural distinguisher is not necessarily aligned with that of long-round differential-linear search. Differential-neural training directly optimizes binary classification under a fixed data distribution and a fixed number of rounds, whereas long-round differential-linear analysis must additionally account for the behavior of the selected intermediate mask over the following rounds. We therefore investigate whether locally strong differential-linear features discovered by a fixed-round difference-only differential-neural distinguisher remain useful after classical linear extension.

To study this question, we analyze differential-neural features from two complementary perspectives: Fourier reconstruction of the final decision rule and direct inspection of first-layer filters through CONV1DFULLY. We then evaluate the resulting mask candidates

under classical long-round linear extension and use sparsity guidance to examine whether structural bias can improve the discovery of low-Hamming-weight candidates. Our main contributions are summarized as follows³.

- **Mask-level analysis of neural representations.** We introduce CONV1DFULLY to make the learned representation of a difference-only differential-neural distinguisher directly accessible at the bit level. The architecture removes the residual tower and reorganizes the first convolution so that each learned filter is aligned with the ciphertext-difference bits, allowing dominant filter coordinates to be projected onto sparse classical mask candidates. In a controlled Speck32/64 experiment, we further show that the dominant differential-linear Fourier feature identified in the Gohr-style architecture remains unchanged after these modifications, supporting the use of CONV1DFULLY for subsequent mask-level analysis.
- **Short-round neural differential-linear candidate discovery.** We provide empirical evidence that difference-only differential-neural distinguishers can serve as meaningful sources of short-round classical differential-linear candidates. On SipHash, we compare Fourier masks extracted from trained differential-neural distinguishers with an exhaustive classical evaluation of all output masks of Hamming weight at most 4 under a fixed input difference. The neural-extracted masks are concentrated among high-correlation differential-linear approximations, including several of the strongest candidates in the examined space. These results support viewing short-round differential-neural distinguishers as data-driven generators of classical differential-linear candidates without claiming global optimality over the complete mask space.
- **Local neural preference versus long-round differential-linear utility.** We evaluate neural-discovered intermediate masks within the known 18 (5+8+5)-round Speck128/128 differential-linear setting. Although several neural-discovered candidates exhibit substantially stronger 8-round middle correlations than the classically selected mask, extending them through the following 5 rounds with single linear trails results in substantially weaker 18-round correlations. We further introduce sparsity guidance on the first convolutional layer to promote low-Hamming-weight candidates. This markedly increases the recovery frequency of the intermediate mask used in the classical 18-round distinguisher, showing that long-round useful masks can enter the neural-generated candidate set. However, the recovery remains unstable, and the final neural decision rule continues to favor locally stronger features. These results show that differential-neural distinguishers can assist long-round candidate generation, while reliable recovery and long-round-aware prioritization remain unresolved.

The rest of the paper is organized as follows. Section 2 introduces the necessary background. Section 3 studies the extraction and mask-level analysis of neural-discovered differential-linear features. Section 4 evaluates these features in the long-round Speck128/128 setting and examines the effect of sparsity guidance on candidate discovery. Section 5 concludes the paper.

2 Preliminaries

2.1 Brief Description of Speck and SipHash

Speck [BSS⁺13] is a family of ARX block ciphers with round function

$$x_{i+1} = ((x_i \ggg \alpha) \boxplus y_i) \oplus k_i, \quad y_{i+1} = (y_i \lll \beta) \oplus x_{i+1},$$

³An anonymized repository for the code and experiments of this work is available at https://anonymous.4open.science/r/revisiting_differential_neural_cryptanalysis-F573/README.md

where x_i and y_i are the two n -bit state words and k_i is the round key. Here, $\ggg$, $\lll$, $\boxplus$, and $\oplus$ denote right rotation, left rotation, modular addition, and bitwise XOR, respectively. The rotation constants are $(\alpha, \beta) = (7, 2)$ for **Speck32/64** and $(8, 3)$ for **Speck128/128**. Throughout this work, bit positions are numbered from the least significant bit (LSB) within each word, with the LSB at the rightmost position.

SipHash [AB12] is a family of ARX-based pseudorandom functions using a 128-bit key, a 256-bit internal state consisting of four 64-bit words, and a 64-bit output. In this work, we consider the standard instance **SipHash-2-4**. Our analysis focuses on the finalization phase, in which four rounds are applied before the four 64-bit state words are XORed to produce the final 64-bit output. The round function is shown in Fig. 1.

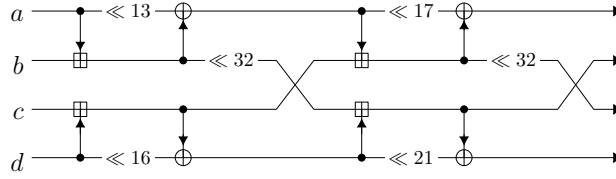


Figure 1: Round function of SipHash.

2.2 Differential-Linear Distinguishers

We consider differential-linear (DL) distinguishers in the three-part framework of Bar-On et al. [BDKW19]; see Fig. 2. The cipher is decomposed as

$$E = E_2 \circ E_m \circ E_1,$$

where E_1 contains a differential transition $\Delta_{\text{in}} \xrightarrow{p} \Delta_m$, E_2 contains a linear approximation $\Gamma_m \xrightarrow{q} \Gamma_{\text{out}}$, and the contribution of E_m is evaluated experimentally as

$$r = \mathbb{E}_{x \in S} \left[(-1)^{\langle \Gamma_m, E_m(x) \rangle \oplus \langle \Gamma_m, E_m(x \oplus \Delta_m) \rangle} \right],$$

where S denotes the sample set used for estimation.

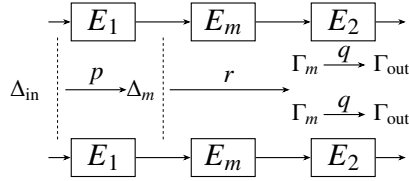


Figure 2: Differential-linear distinguisher.

The resulting DL correlation is given by prq^2 , yielding a data complexity of approximately $\varepsilon p^{-2} r^{-2} q^{-4}$ chosen-plaintext pairs, where ε is a non-negative constant. For a fixed differential prefix, the final distinguisher quality depends on both the middle correlation r and the following linear correlation q . In particular, a large $|r|$ alone does not necessarily imply a large overall contribution $|rq^2|$.

2.3 Differential-Neural Distinguishers

Let E be a reduced-round cryptographic primitive, Δ a fixed input difference, and ϕ a feature mapping. We denote by $\mathcal{D}_1$ and $\mathcal{D}_0$ the distributions of

$$\phi(E_K(x), E_K(x \oplus \Delta)) \quad \text{and} \quad \phi(E_K(x), E_K(x')),$$

respectively, where x and x' are sampled independently and uniformly at random, and K denotes a random key.

Depending on the design of the differential-neural distinguisher, ϕ may retain the ciphertext pair or only a derived representation. In particular, one may consider

$$\phi_{\text{full}}(C_0, C_1) = (C_0, C_1), \quad \phi_{\text{diff}}(C_0, C_1) = C_0 \oplus C_1.$$

The full-pair representation retains more information, since the difference can be deterministically derived from the pair whereas the converse does not hold in general. In this work, however, we focus on the difference-only representation. This choice is deliberate: the parity features extracted in this setting admit direct interpretations as classical differential-linear masks, providing a clean interface between neural feature discovery and classical differential-linear analysis. Full-pair differential-neural distinguishers may additionally exploit unequal-mask variant-DL structures, whose generalized long-round extension is outside the scope of this work.

A trained differential-neural distinguisher induces a binary decision function

$$f : \mathcal{X} \rightarrow \{0, 1\},$$

where $\mathcal{X}$ denotes the output space of ϕ . Here, $f(z) = 1$ and $f(z) = 0$ classify z as originating from $\mathcal{D}_1$ and $\mathcal{D}_0$, respectively. Thus, the differential-neural distinguisher exploits a detectable statistical difference between these two feature distributions.

2.4 Fourier-Based Feature Extraction

Yuan et al. [YWZW26] introduced a Fourier-based framework for analyzing the Boolean decision function of a trained differential-neural distinguisher. Given

$$f : \{0, 1\}^d \rightarrow \{0, 1\},$$

where d denotes the dimension of the neural input representation, they consider its sign representation

$$g(x) = (-1)^{f(x)}$$

and Fourier expansion

$$g(x) = \sum_{S \subseteq [d]} \hat{g}(S) \chi_S(x), \quad \hat{g}(S) = \mathbb{E}_{x \sim U(\mathbb{F}_2^d)} [g(x) \chi_S(x)],$$

where $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$. A Fourier coefficient with large magnitude indicates a strong correlation between the corresponding parity feature and the sign representation of the neural decision rule.

To recover the dominant parity features without exhaustively enumerating the Fourier spectrum, Yuan et al. employ the Goldreich–Levin algorithm to identify heavy Fourier coefficients above a prescribed threshold τ , thereby obtaining a sparse Fourier approximation of the sign representation; details are given in Appendix A. They further show that, for the difference-only representation, a parity feature indexed by a mask Γ satisfies

$$\langle \Gamma, C_0 \oplus C_1 \rangle = \langle \Gamma, C_0 \rangle \oplus \langle \Gamma, C_1 \rangle,$$

so the corresponding Fourier term admits a direct interpretation as a classical equal-mask DL feature.

In this work, we adopt this framework as a feature-extraction tool for trained difference-only differential-neural distinguishers. The Fourier interpretation itself is inherited from prior work; our focus is on whether the resulting classical equal-mask DL features can serve as useful candidates in the search for longer-round classical DL distinguishers.

3 From Neural Features to Classical Differential-Linear Candidates

Yuan et al. [YWZW26] showed that, under a difference-only representation, Fourier features extracted from differential-neural distinguishers admit direct interpretations as classical differential-linear approximations. Building on this connection, this section studies whether such neural features can be used as data-driven candidates for classical differential-linear cryptanalysis. We first introduce CONV1DFULLY, whose first-layer filters can be projected onto sparse mask candidates, and verify on **Speck32/64** that the dominant differential-linear feature considered here is preserved after the architectural reorganization. We then use **SipHash** to examine whether the masks extracted from trained differential-neural distinguishers are concentrated among locally strong differential-linear approximations in an exhaustively evaluated low-Hamming-weight space. Together, these experiments provide evidence that, in the short-round settings considered here, difference-only differential-neural distinguishers can serve as data-driven generators of classical differential-linear candidates.

3.1 From Fourier Features to Classical DL Masks

As reviewed in Section 2.4, Yuan et al. [YWZW26] established that, under a difference-only representation, Fourier terms extracted from a trained differential-neural distinguisher admit direct interpretations as classical equal-mask differential-linear approximations. In this setting, each retained Fourier support can therefore be associated with an explicit classical output mask. Their experiments further showed that the extracted masks can include strong multi-bit differential-linear approximations.

A concrete example was reported by Yuan et al. [YXZW25]. They trained a 9-round differential-neural distinguisher for **Speck32/64** with input difference

$$\Delta = (0\text{x}a840, 0\text{x}0010),$$

which is also the input difference of the known 9-round differential-linear distinguisher [BGG⁺23]

$$(0\text{x}a840, 0\text{x}0010) \xrightarrow{4} (0\text{x}1000, 0\text{x}5000) \xrightarrow{2} (0\text{x}1000, 0\text{x}0020) \xrightarrow{3} (0\text{x}0205, 0\text{x}0204). \quad (1)$$

The practical correlation of the distinguisher in Eq. (1) is $2^{-7.3}$. Using a training set of size 2^{30} under the same input difference, Yuan et al. [YXZW25] obtained a differential-neural distinguisher with an accuracy of 0.5016. Its leading Fourier term was

$$\{25, 18, 16, 9, 2\},$$

which corresponds exactly to the output mask

$$(0\text{x}0205, 0\text{x}0204)$$

in Eq. (1). Thus, the dominant Fourier feature of the trained distinguisher recovers a multi-bit mask already known to define a useful classical differential-linear approximation.

This correspondence provides the interface needed for the remainder of this section: Fourier features extracted from a difference-only differential-neural distinguisher can be converted into explicit classical mask candidates. To complement this decision-level view with a direct analysis of the learned internal representation, we next introduce an architecture whose first-layer parameters can be projected onto sparse classical masks.

3.2 Architectural Reorganization for DL Feature Analysis

Gohr [Goh19] introduced a representative network architecture for **Speck32/64**; see Fig. 3a. The architecture consists of an initial bit-sliced convolution layer, a residual tower, and a

fully connected prediction head. Since several subsequent interpretability studies are based on this architecture, we use it as the baseline for our modification. A detailed description of the network is deferred to Appendix B.

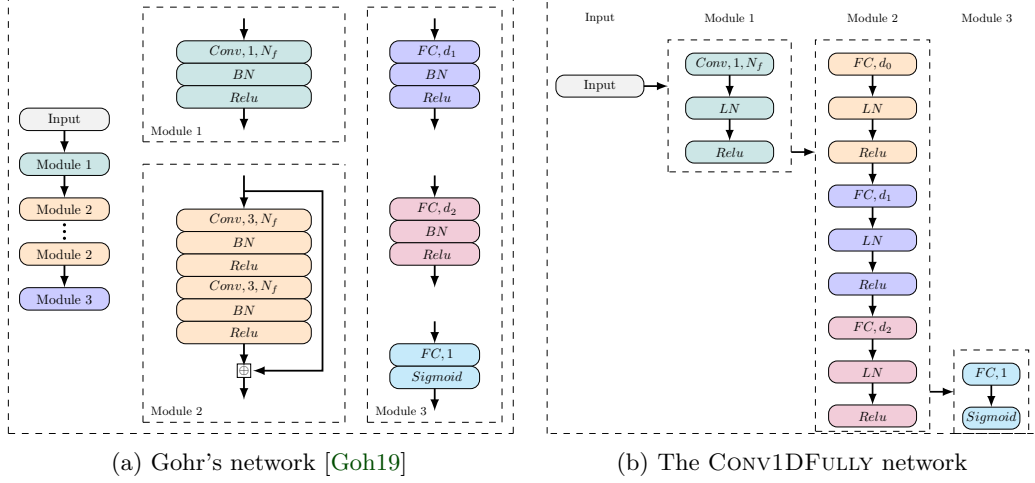


Figure 3: Representative network architectures for differential-neural distinguishers

Block-Level Interpretation of Gohr's Distinguisher Benamira et al. [BGPT21] analyzed the main components of Gohr's differential-neural distinguisher and provided different levels of interpretation for them. For the initial convolution block, they showed that many learned filters capture simple combinations of ciphertext differences and related derived quantities, giving a relatively explicit description of the low-level information extracted from the input. They also showed that the final prediction stage can be replaced by simpler and more interpretable classifiers when applied to suitable intermediate representations.

The residual tower, by contrast, was interpreted mainly at a functional level. Benamira et al. related its behavior to the processing of differential-distribution information and masked views of the output distribution, and their experiments further indicated that the resulting representation is compatible with truncated-differential and differential-linear behavior. However, this interpretation does not expose the internal features of the residual tower as explicit bit-wise cryptanalytic objects or individual classical masks. This motivates us to remove the residual tower and seek a first-layer representation that can be analyzed more directly at the mask level.

First-Layer Mask Projection We introduce CONV1DFULLY, shown in Fig. 3b, with two deliberate modifications to Gohr's architecture. First, the residual tower is removed, yielding a more direct computational path from the input representation to the prediction head. Second, the first convolution is reorganized to operate directly along the ciphertext-difference bit dimension rather than using the original bit-sliced organization.

For a d -bit ciphertext-difference input

$$x \in \{0, 1\}^d,$$

the first layer produces N_f channels and is parameterized by

$$W \in \mathbb{R}^{N_f \times d}.$$

Each row

$$W_j \in \mathbb{R}^d$$

assigns a learned coefficient to each ciphertext-difference bit. We use the relative magnitudes of these coefficients as a support signal for constructing sparse classical mask candidates. For a prescribed Hamming weight k , the k coordinates with largest absolute value are retained:

$$W_j \longrightarrow \text{TopK}(|W_j|) \longrightarrow \Gamma_j,$$

where Γ_j is a Hamming-weight- k binary mask. The resulting first-layer features are subsequently combined and reweighted by three fully connected hidden layers before the final output layer.

The purpose of CONV1DFULLY is not to improve distinguishing accuracy or parameter efficiency, but to make the learned first-layer representation directly accessible for mask-level analysis. The first-layer parameters indicate which ciphertext-difference coordinates are emphasized internally, while Fourier analysis of the final decision rule provides complementary evidence about which parity features remain influential after the subsequent layers. Before using this architecture for candidate extraction, we therefore verify in the next subsection that the dominant Fourier feature associated with the known differential-linear approximation is preserved after the architectural reorganization.

3.3 Preservation of the Dominant Fourier Feature

We next examine whether CONV1DFULLY preserves the dominant Fourier feature associated with the known differential-linear approximation. The natural reference is the 9-round Speck32/64 construction in Eq. 1. However, Yuan et al. [YXZW25] required 2^{30} training samples for the corresponding differential-neural distinguisher, which achieved an accuracy of only 0.5016. This weak signal is inconvenient for repeated architecture-level comparison. We therefore derive a stronger controlled task from the same differential-linear construction and compare the Gohr-style ResNet with CONV1DFULLY at both the classification and Fourier levels.

Controlled Setting The 9-round distinguisher in Eq. 1 consists of a 4-round differential prefix, followed by a 2-round middle component E_m and a 3-round linear approximation E_2 . Removing the 4-round differential prefix yields the following 5-round differential-linear approximation:

$$(0x1000, 0x5000) \xrightarrow{2} (0x1000, 0x0020) \xrightarrow{3} (0x0205, 0x0204). \quad (2)$$

Accordingly, we use

$$\Delta = (0x1000, 0x5000)$$

as the input difference and train differential-neural distinguishers on 5-round Speck32/64. This 5-round approximation preserves the same 2-round middle component and output mask as the final 5 rounds of the 9-round distinguisher, and its differential-linear correlation is 2^{-1} . It therefore provides a stronger and more stable setting for isolating the effect of the architectural reorganization.

Classification-Level Comparison Within this controlled setting, the Gohr-style ResNet and CONV1DFULLY are trained using the same data-generation procedure, input difference, ciphertext-difference representation, and positive and negative sample distributions. The reproduced Gohr-style ResNet serves as the baseline architecture, while CONV1DFULLY follows the design introduced in Section 3.2. Detailed training settings are provided in Appendix C.

Table 1 summarizes the classification performance and parameter counts of the two models. Both architectures are trained over 10 independent random seeds, and for each run we retain the checkpoint with the lowest validation loss. The two architectures achieve

comparable distinguishing accuracy in this controlled setting, although CONV1DFULLY uses substantially more parameters. Classification performance is not the primary quantity of interest here; rather, this comparison verifies that the reorganized architecture retains sufficient distinguishing capability for the subsequent feature analysis.

Table 1: Validation accuracy and parameter counts of the differential-neural distinguishers on Speck32/64. Accuracy is reported as mean $\pm$ standard deviation over independent training runs.

Network	Parameters	Accuracy
Gohr-style ResNet	69217	0.6268 ± 0.0003
CONV1DFULLY	1367041	0.6255 ± 0.0004

Fourier-Level Feature Preservation We next compare the decision rules induced by the two architectures using the Fourier-reconstruction framework of Yuan et al. [YWZW26]. We set the reconstruction threshold to $\tau = 0.1$ and use the retained heavy Fourier terms to construct a sparse approximation of each decision rule.

To evaluate the reconstruction quality, we sample 10^5 fresh inputs independently from the Boolean input space and compare the classifications produced by the reconstructed rule with those of the original differential-neural distinguisher. The *similarity* reported in Table 2 is the fraction of inputs on which the two rules agree. Thus, the Fourier coefficient quantifies the strength of an individual retained term, whereas the similarity measures how closely the reconstructed rule reproduces the behavior of the original differential-neural distinguisher.

Table 2: Fourier reconstruction of the Gohr-style ResNet and CONV1DFULLY on the 5-round Speck32/64 task. Similarity and Fourier coefficients are reported as mean $\pm$ standard deviation over independent training runs.

Network	Similarity	Fourier term S	Coefficient
Gohr-style ResNet	$96.54\% \pm 0.18\%$	$\{25, 18, 16, 9, 2\}$	0.9308 ± 0.0034
CONV1DFULLY	$98.45\% \pm 0.69\%$	$\{25, 18, 16, 9, 2\}$	0.9666 ± 0.0138

For both architectures, the dominant retained Fourier term is

$$S = \{25, 18, 16, 9, 2\},$$

which corresponds exactly to the output mask

$$(0x0205, 0x0204)$$

in Eq. 2. Moreover, across all 10 independently trained CONV1DFULLY models, this term remains dominant. The reconstruction similarity ranges from 97.658% to 99.633%, while the corresponding Fourier coefficient ranges from 0.95434 to 0.99298. In this controlled setting, the dominant Fourier feature is therefore more pronounced in CONV1DFULLY: the same term is dominant for both architectures, while CONV1DFULLY exhibits both a higher reconstruction similarity and a larger Fourier coefficient for this term.

These results do not imply that the learned representations are architecture independent. Rather, in this controlled setting, the dominant Fourier feature associated with the known differential-linear approximation remains preserved after removing the residual tower and reorganizing the first convolution along the ciphertext-difference bit dimension. This controlled experiment supports the use of CONV1DFULLY for the subsequent mask-level candidate analysis.

3.4 Local Strength of Neural-Discovered Differential-Linear Approximations

The preceding experiment shows that the dominant differential-linear approximation considered here remains visible after the architectural reorganization. We now ask a different question: when multiple classical differential-linear approximations exist under the same input difference, are the features exposed by a differential-neural distinguisher concentrated among those with stronger local correlations? To study this question, we use `SipHash`, for which the low-Hamming-weight output-mask space is sufficiently large to provide a meaningful comparison while remaining small enough for exhaustive evaluation in the selected short-round setting.

Classical Setting We use the input difference appearing in the 4-round differential-linear distinguisher of [NSLL22]:

$$\Delta = (0x40000, 0x80040000, 0x0, 0x0).$$

Under this setting, Niu et al. [NSLL22] derived a differential-linear distinguisher with output mask

$$\gamma_{\text{out}} = 0x2000000020000000,$$

whose active bit positions are $\{61, 29\}$ and whose experimental correlation is $2^{-6.03}$.

We use this result only to define a common classical reference setting; the mask $\{61, 29\}$ is not treated as a target that the differential-neural distinguisher is expected to reproduce. Once the input difference Δ is fixed, the neural model may exploit any output relation that is statistically useful for the corresponding 4-round distinguishing task.

Neural Candidate Extraction Using `CONV1DFULLY` under the same input difference, we train a 4-round differential-neural distinguisher on 10^7 samples, corresponding to approximately $2^{23.25}$ training instances. We repeat the training over 10 independent random seeds and select, for each run, the checkpoint with the lowest validation loss. The selected models achieve a mean validation accuracy of 0.5191 ± 0.0004 , where the variation denotes the standard deviation across the 10 independent runs.

Table 3: Recurrent Fourier terms across 10 runs of the 4-round `SipHash` distinguisher

Fourier term	Corr.	Freq.	Fourier term	Corr.	Freq.
$\{25\}$	$2^{-4.86}$	10	$\{9\}$	$2^{-5.44}$	10
$\{25, 24\}$	$2^{-4.92}$	10	$\{46, 29, 14\}$	$2^{-5.43}$	9
$\{41\}$	$2^{-5.06}$	10	$\{45, 44, 13, 12\}$	$2^{-5.16}$	8
$\{41, 40\}$	$2^{-5.08}$	10	$\{41, 9\}$	$2^{-5.55}$	8
$\{57, 56, 25, 24\}$	$2^{-5.12}$	10	$\{41, 40, 9, 8\}$	$2^{-5.56}$	6
$\{57, 25\}$	$2^{-5.15}$	10	$\{62, 45, 30\}$	$2^{-5.66}$	2
$\{45, 13\}$	$2^{-5.17}$	10	$\{46, 29, 28, 14\}$	$2^{-5.74}$	1
$\{9, 8\}$	$2^{-5.38}$	10	$\{61, 29\}$	$2^{-6.03}$	1

Terms are ordered first by decreasing frequency and then by decreasing measured correlation. Corr. denotes the experimentally estimated correlation of the corresponding differential-linear approximation under the same input difference, using 2^{18} random pairs. Freq. denotes the number of runs, out of 10, in which the Fourier term is retained by Algorithm 1 with $\tau = 0.1$.

We then apply Algorithm 1 with reconstruction threshold $\tau = 0.1$. Table 3 summarizes the Fourier terms that recur across the 10 independently trained models, together with the experimentally estimated differential-linear correlation of each corresponding output mask.

Exhaustive Low-Weight Comparison To place the neural-extracted masks in a classical differential-linear context, we exhaustively enumerate all output masks of Hamming weight at most 4 under the same fixed input difference. The search space contains

$$\binom{64}{1} + \binom{64}{2} + \binom{64}{3} + \binom{64}{4} = 679120$$

candidate masks. For each candidate, we experimentally estimate the corresponding differential-linear correlation. Table 4 lists all masks in this search space whose measured correlation exceeds 2^{-6} and indicates whether each mask appears among the Fourier terms extracted from the differential-neural distinguishers.

Table 4: Differential-linear approximations for 4-round SipHash with correlation larger than 2^{-6}

Output mask	Corr.	Appears	Output mask	Corr.	Appears
{25}	$2^{-4.86}$	✓	{46, 29, 14}	$2^{-5.43}$	✓
{25, 24}	$2^{-4.92}$	✓	{9}	$2^{-5.44}$	✓
{41}	$2^{-5.06}$	✓	{41, 9}	$2^{-5.55}$	✓
{41, 40}	$2^{-5.08}$	✓	{41, 40, 9, 8}	$2^{-5.56}$	✓
{57, 56, 25, 24}	$2^{-5.12}$	✓	{62, 45, 30}	$2^{-5.66}$	✓
{57, 25}	$2^{-5.15}$	✓	{46, 29, 28, 14}	$2^{-5.74}$	✓
{45, 44, 13, 12}	$2^{-5.16}$	✓	{62, 45, 44, 30}	$2^{-5.75}$	×
{45, 13}	$2^{-5.17}$	✓	{57}	$2^{-5.89}$	×
{9, 8}	$2^{-5.38}$	✓	{62, 30, 13}	$2^{-5.97}$	×

Appears indicates whether the corresponding mask is listed among the Fourier terms in Table 3. The candidates are arranged in two columns for compact presentation; within each column, masks are listed in decreasing order of measured correlation.

Correlation Strength and Recovery Frequency The exhaustive comparison shows that the neural-extracted masks are concentrated among strong low-weight differential-linear approximations under the fixed input difference. Among the 18 masks of Hamming weight at most 4 whose measured correlations exceed 2^{-6} , 15 appear among the reconstructed Fourier terms. In particular, the four highest-correlation masks in Table 4, namely {25}, {25, 24}, {41}, and {41, 40}, are all recovered and recur across all 10 independent runs.

The recurrence frequencies in Table 3 exhibit a broad tendency to decrease as the measured differential-linear correlation becomes weaker. Many of the stronger approximations appear in all 10 runs, whereas several weaker ones are recovered less frequently. This relation is not strictly monotonic, and the recurrence frequency should therefore not be interpreted as being determined solely by the differential-linear correlation. Nevertheless, the observed pattern indicates an association between local statistical strength and the stability with which an approximation is exposed by the differential-neural distinguisher.

The three masks above the 2^{-6} threshold that are absent from the reconstructed Fourier terms have measured correlations $2^{-5.75}$, $2^{-5.89}$, and $2^{-5.97}$, respectively, making them the three weakest candidates listed in Table 4. This pattern suggests a possible association between local correlation strength and recovery stability. However, since the present

experiment does not vary the training-set size or control for other factors affecting neural feature recovery, we do not interpret this observation as establishing a causal relation.

The exhaustive evaluation is restricted to output masks of Hamming weight at most 4 and therefore does not establish optimality over the full 2^{64} output-mask space. Within the examined low-weight space, however, the differential-neural distinguisher recovers 15 of the 18 masks above the selected correlation threshold, including the strongest approximations identified by the exhaustive evaluation. These results provide evidence that, in the short-round setting considered here, difference-only differential-neural distinguishers can serve as data-driven generators of locally strong classical differential-linear candidates. Complementary experiments on **ChaCha** are provided in Appendix D and offer additional qualitative evidence in a larger-state setting.

Taken together, the **Speck32/64** and **SipHash** experiments establish the premise for the subsequent long-round study. The former shows that the dominant differential-linear feature considered here is preserved after the architectural reorganization, while the latter shows that masks extracted from a short-round differential-neural distinguisher are concentrated among strong classical approximations in an exhaustively evaluated low-Hamming-weight space. These observations motivate treating neural-extracted masks as candidates for subsequent classical long-round analysis.

4 From Neural-Discovered DL Candidates to Long-Round DL Distinguishers

The previous section shows that, in the short-round difference-only settings considered here, differential-neural distinguishers can expose locally strong classical differential-linear candidates. We now ask whether such candidates remain useful as intermediate masks in long-round differential-linear analysis. To study this question, we use a known 18-round distinguisher for **Speck128/128** as a controlled setting: we train a differential-neural distinguisher over its 8-round middle component, extract alternative intermediate masks, and evaluate their 5-round single XOR-linear trail extensions. Since the fixed-round neural training objective does not directly account for the quality of the following linear approximation, we further introduce sparsity guidance as a structural bias to encourage the emergence of low-Hamming-weight candidates and examine whether this changes the masks exposed and prioritized by the neural distinguisher.

4.1 A Classical 18-Round Differential-Linear Distinguisher for Speck128/128

We first revisit the 18-round differential-linear distinguisher of Chen et al. [CBY23], which defines the long-round setting used throughout this section.

Classical Search and the 18-Round Distinguisher Chen et al. [CBY23] proposed a semi-unconstrained search framework for differential-linear approximations. Their method starts from strong unbalanced bits and combines them through iterative and meet-in-the-middle procedures, thereby relaxing the common restriction that the intermediate linear mask is limited to Hamming weight one or two. The search also incorporates constraints from the following linear approximation when constructing candidate masks.

Using this methodology, Chen et al. [CBY23] derived the following 18-round differential-linear distinguisher for **Speck128/128**:

$$(0x40002403c012, 0x10020040000400c2) \xrightarrow{5} \{117\} \xrightarrow{8} \{77, 5\} \xrightarrow{5} (0xa49000000020343, 0x208000000020303).$$

Under the standard three-part differential-linear decomposition, the first 5 rounds form the differential component E_1 , the following 8 rounds form the middle component E_m , and the final 5 rounds form the linear approximation E_2 . The corresponding quantities are

$$p = 2^{-30}, \quad r = 2^{-5.81}, \quad q = 2^{-10},$$

which give the estimated overall correlation

$$prq^2 = 2^{-30} \times 2^{-5.81} \times 2^{-20} = 2^{-55.81}.$$

Choice of the 5 + 8 + 5 Decomposition Gong et al. [GWH⁺25] later re-decomposed the same 18-round distinguisher into a 5 + 9 + 4 structure. In their decomposition, the two-bit intermediate mask $\{77, 5\}$ propagates for one additional round to the single-bit mask $\{5\}$, yielding an hourglass-like boundary. We nevertheless retain Chen et al.’s original 5 + 8 + 5 decomposition because our goal is to study whether a difference-only differential-neural distinguisher can discover useful multi-bit intermediate masks. Under the 5+8+5 boundary, the output of the 8-round middle component remains the multi-bit mask $\{77, 5\}$, whereas the 5 + 9 + 4 decomposition would make the corresponding neural boundary a known singleton mask. The two decompositions therefore serve different analytical purposes, and the 5 + 8 + 5 setting is more suitable for the candidate-discovery question considered here.

Controlled Setting for Intermediate-Mask Comparison Although the semi-unconstrained framework permits more general multi-bit intermediate masks, the mask selected in this particular Speck128/128 distinguisher is the sparse Hamming-weight-2 mask $\{77, 5\}$. This setting is suitable for our study because it fixes the middle input difference $\{117\}$ and provides an explicit intermediate mask together with a concrete following 5-round linear approximation.

In the following analysis, we therefore fix

$$\Delta = \{117\}$$

and train a difference-only differential-neural distinguisher over the same 8-round middle component E_m . The masks exposed by the neural model are treated as alternative intermediate-mask candidates and are subsequently evaluated through the same following 5 rounds.

4.2 Neural Candidate Extraction for the Speck128/128 Middle Component

We next examine which intermediate masks are exposed by an 8-round difference-only differential-neural distinguisher trained on the same middle component E_m . Using CONV1DFULLY with input difference $\{117\}$, we train the distinguisher on a dataset of size 10^7 for 30 epochs. Across 30 independent random seeds, the selected models achieve a mean validation accuracy of 0.8217 ± 0.0012 , with a range of $[0.8191, 0.8238]$, where $\pm$ denotes the standard deviation across runs. We analyze the resulting candidates from two complementary perspectives: Fourier reconstruction identifies parity relations that recur in the final neural decision rules and provides the decision-level candidates for subsequent long-round evaluation, while the first-layer parameter matrix offers a direct parameter-level view of support patterns emphasized by the learned representation and is used to characterize feature visibility within CONV1DFULLY.

Decision-Level Candidates We apply Algorithm 1 with reconstruction threshold $\tau = 0.1$ to the trained models. The reconstruction similarity has mean $82.17\% \pm 0.79\%$ and range $[80.33\%, 83.95\%]$. Across the 30 runs, five nontrivial Fourier terms recur consistently, whereas one additional nontrivial term appears in only a single run. These recurrent terms are associated with strong local 8-round differential-linear correlations under the fixed input difference $\{117\}$. By contrast, the classically selected mask $\{77, 5\}$ is not retained under $\tau = 0.1$ in any of the 30 runs. This does not imply that it is absent from the learned representation, since Fourier reconstruction characterizes the final neural decision rule after the intermediate features have been processed by the subsequent layers. Table 5 summarizes the retained terms and their statistics.

Table 5: Nontrivial Fourier terms retained across 30 runs of the 8-round Speck128/128 distinguisher

Fourier term S	Frequency	Range of $ \widehat{g}(S) $	Middle corr. $ r $
$\{75, 67, 64, 11, 3\}$	30/30	$[0.41524, 0.48224]$	$2^{-0.01}$
$\{75, 72, 67, 64, 11, 8, 3\}$	29/30	$[0.11442, 0.14846]$	$2^{-0.97}$
$\{72, 8\}$	28/30	$[0.11656, 0.15140]$	$2^{-0.97}$
$\{75, 72, 71, 67, 64, 11, 8, 7, 3\}$	26/30	$[0.10874, 0.13048]$	$2^{-0.99}$
$\{72, 71, 8, 7\}$	25/30	$[0.10510, 0.13300]$	$2^{-0.99}$
$\{127, 124, 123, 75, 71, 67, 64, 63, 11, 7, 3\}$	1/30	$[0.11030, 0.11030]$	$2^{-1.99}$

Frequency denotes the number of runs in which the term is retained. Range gives the observed range of $|\widehat{g}(S)|$ over those runs. Middle corr. $|r|$ is the experimentally estimated absolute 8-round differential-linear correlation under input difference $\{117\}$ using 2^{30} random pairs.

The recurrent decision-level features are therefore concentrated on locally strong differential-linear relations, while the classically selected mask is not emphasized by the final neural decision rule.

Representation-Level Visibility We next examine the parameter matrix of the first convolutional layer to determine whether additional sparse masks are visible in the learned representation even when they are not retained by the final Fourier reconstruction. For the 128-bit ciphertext-difference input, the layer contains 1024 channels and has parameter matrix

$$W \in \mathbb{R}^{1024 \times 128}.$$

Following the support-oriented projection introduced in Section 3.2, we project each row onto a sparse binary mask by retaining the positions of its largest absolute weights.

We choose $k = 2$ to examine Hamming-weight-2 candidates, matching the weight of the classically selected mask $\{77, 5\}$. Since different channels may project to the same binary mask, a candidate may occur multiple times within a single model. Table 6 reports, for one representative model, the projected masks whose experimentally estimated correlations exceed 2^{-8} .

Among the

$$\binom{128}{2} = 8128$$

possible Hamming-weight-2 masks, 77 have experimentally estimated correlations larger than 2^{-8} . Across the 30 independently trained models, the first-layer projection exposes 34.40 ± 1.96 such high-correlation masks per model on average, with the number ranging from 32 to 39 across random seeds.

Table 6: Top-2 projected masks from the first convolutional layer of the 8-round Speck128/128 distinguisher

Projected mask	Corr.	Channels	Projected mask	Corr.	Channels
{72, 8}	$2^{-0.97}$	70	{5, 3}	$2^{-5.20}$	6
{71, 7}	$2^{-1.15}$	61	{4, 3}	$2^{-5.01}$	4
{70, 6}	$2^{-1.64}$	51	{67, 4}	$2^{-3.93}$	4
{67, 3}	$2^{-5.03}$	34	{64, 5}	$2^{-7.65}$	4
{69, 5}	$2^{-2.19}$	20	{67, 5}	$2^{-3.53}$	4
{75, 11}	$2^{-7.78}$	20	{68, 5}	$2^{-3.11}$	4
{64, 0}	$2^{-6.18}$	14	{69, 68}	$2^{-2.57}$	4
{64, 3}	$2^{-6.12}$	14	{69, 3}	$2^{-5.27}$	3
{124, 123}	$2^{-6.88}$	14	{5, 4}	$2^{-3.48}$	3
{126, 62}	$2^{-7.34}$	13	{68, 64}	$2^{-7.14}$	2
{67, 64}	$2^{-6.77}$	12	{69, 64}	$2^{-7.64}$	2
{68, 4}	$2^{-2.95}$	11	{13, 3}	$2^{-7.87}$	1
{127, 63}	$2^{-5.31}$	11	{66, 3}	$2^{-5.57}$	1
{79, 15}	$2^{-3.08}$	10	{68, 3}	$2^{-5.27}$	1
{68, 67}	$2^{-3.21}$	9	{69, 4}	$2^{-3.65}$	1
{80, 16}	$2^{-2.35}$	8	{67, 13}	$2^{-7.29}$	1
{124, 60}	$2^{-6.11}$	8	{123, 59}	$2^{-7.95}$	1
{64, 4}	$2^{-7.18}$	7	{67, 66}	$2^{-4.16}$	1
{69, 67}	$2^{-3.34}$	6	{77, 69}	$2^{-5.57}$	1

Corr. denotes the experimentally estimated correlation of the corresponding differential-linear approximation under input difference {117}, using 2^{30} random pairs. Channels denotes the number of first-layer channels in this representative model whose Top-2 projection yields the corresponding mask.

In the representative model, multiple first-layer channels project to several masks associated with strong local correlations, including {72, 8}, {71, 7}, {70, 6}, {67, 3}, and {69, 5}. By contrast, the classically selected mask {77, 5} is absent from the representative model shown in Table 6 and appears in the Top-2 projection of only one of the 30 independently trained models.

The first-layer analysis is used here to characterize representation-level visibility rather than to define the candidate set for long-round evaluation; the latter is determined by the recurrent Fourier terms identified at the decision level. Taken together, the two analyses show that the trained differential-neural distinguishers expose several classical differential-linear relations with strong local correlations under the fixed 8-round middle-component setting. The next subsection evaluates whether these locally strong decision-level candidates remain useful as intermediate masks after the following 5-round linear approximation is taken into account.

4.3 Long-Round Evaluation of Neural-Discovered Masks

We now evaluate the neural-discovered intermediate masks within the classical long-round differential-linear framework. Among the six nontrivial Fourier terms in Table 5, five recur

across independent runs, whereas one appears in only a single run. We therefore focus on the five recurrent terms and include the classically selected mask $\{77, 5\}$ separately as a reference. The goal is to compare the local strength of these masks over the 8-round middle component with their utility after the following 5-round linear approximation is taken into account.

Classical Linear Extension For each candidate intermediate mask, we fix the mask as the input mask of the following 5 rounds and use a SAT encoding solved with CRYPTOMINISAT to search for a minimum-weight single XOR-linear trail. The resulting trail correlation is used as the linear correlation q in the differential-linear evaluation, where q refers to a single-trail correlation rather than a linear-hull correlation. Since the preceding 5-round differential component is fixed for all candidates with probability $p = 2^{-30}$, comparing the candidates only requires the 8-round middle correlation r and the following 5-round linear correlation q . We therefore report the candidate-dependent quantity $|rq^2|$ over the last 13 rounds; the complete 18-round correlation is obtained by multiplying by the common factor p . Table 7 reports $|r|$, $|q|$, and $|rq^2|$ for the five recurrent neural candidates and the classical reference mask, together with the corresponding output masks in left-word/right-word order.

Table 7: Middle correlations and 5-round single XOR-linear extensions for Speck128/128

Intermediate mask	$ r $	q	Output mask of the 5-round trail	$ rq^2 $
$\{75, 67, 64, 11, 3\}$	$2^{-0.01}$	2^{-25}	0x0900000004024498 0x0a00000004028418	$2^{-50.01}$
$\{75, 72, 67, 64, 11, 8, 3\}$	$2^{-0.97}$	2^{-30}	0x101300000482dc80 0x1010000004820402	$2^{-60.97}$
$\{72, 8\}$	$2^{-0.97}$	2^{-19}	0x00829a6000809800 0x0080820000808002	$2^{-38.97}$
$\{75, 72, 71, 67, 64, 11, 8, 7, 3\}$	$2^{-0.99}$	2^{-28}	0xc300000004c04481 0xd000000004c0dc18	$2^{-56.99}$
$\{72, 71, 8, 7\}$	$2^{-0.99}$	2^{-20}	0x0332030c00c08001 0x0102034000c09001	$2^{-40.99}$
$\{77, 5\}$	$2^{-5.81}$	2^{-10}	0x2b4d000000020262 0x230c000000020202	$2^{-25.81}$

Middle corr. $|r|$ denotes the experimentally estimated absolute correlation over the 8-round middle component. Linear corr. $|q|$ denotes the correlation of the minimum-weight single XOR-linear trail found by the search over the following 5 rounds. The last column reports the candidate-dependent correlation $|rq^2|$ over these 13 rounds; the complete 18-round correlation is obtained by multiplying by the common differential probability $p = 2^{-30}$.

Local Strength versus Long-Round Utility Table 7 shows a clear mismatch between local middle-component strength and utility after the following linear approximation is incorporated. The contrast is particularly clear for $\{72, 8\}$ and the classically selected mask $\{77, 5\}$. Their 8-round middle correlations are $2^{-0.97}$ and $2^{-5.81}$, respectively, whereas the correlations of the minimum-weight 5-round single XOR-linear trails found by the search are 2^{-19} and 2^{-10} . Consequently, the corresponding 13-round quantities $|rq^2|$ are $2^{-38.97}$ and $2^{-25.81}$.

The same pattern is observed across the five recurrent neural candidates. Although all five have strong local middle correlations, their following 5-round single-trail correlations range from 2^{-19} to 2^{-30} , whereas $\{77, 5\}$ admits a substantially stronger 2^{-10} trail.

Thus, ranking the candidates by local middle correlation leads to a markedly different ordering from ranking them by the corresponding $|rq^2|$ values. Since the preceding 5-round differential component is identical for all candidates, this ordering is preserved for the complete 18-round distinguishers.

This phenomenon is fully consistent with the classical differential-linear framework and is not itself a new theoretical observation. The empirical point relevant to our study is that the masks repeatedly emphasized by the fixed-round differential-neural distinguisher correspond to strong relations over the isolated 8-round middle component, whereas the mask selected in the classical 18-round distinguisher has a weaker middle correlation but a substantially stronger following linear approximation. Hence, the preference induced by the fixed-round neural task is not necessarily aligned with the ranking relevant to long-round differential-linear analysis.

In the current workflow, the quality of the following linear approximation becomes available only after an explicit intermediate mask has been extracted and evaluated by classical linear search, and is therefore not directly incorporated into the neural training objective. We therefore next investigate whether a simpler structural bias toward sparse first-layer representations can improve the generation of potentially useful low-Hamming-weight candidates.

4.4 Sparsity Guidance as a Structural Bias for Candidate Discovery

The unconstrained models expose only part of the strong Hamming-weight-2 mask space. Among the 77 masks whose experimentally estimated correlations exceed 2^{-8} , the 30-epoch unconstrained models expose 34.40 ± 1.96 masks on average across 30 independent runs, while the classically selected mask $\{77, 5\}$ appears in the Top-2 projection of only one run. Since intermediate masks used in classical differential-linear constructions are often sparse, we investigate whether a sparsity bias can improve the generation and visibility of low-Hamming-weight candidates.

The sparsity constraint is used only as a structural bias and does not directly encode the quality of the following linear approximation or favor particular bit positions. Its purpose is to encourage a sparser organization of the first-layer representation and thereby increase the coverage of low-support candidate masks.

Sparsity Intervention For Speck128/128, the first-layer parameter matrix is

$$W \in \mathbb{R}^{1024 \times 128}.$$

We augment the standard classification objective with a sparsity regularizer,

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{sp}}(e) b_{\text{sp}}(e) \mathcal{L}_{\text{sp}}(W),$$

which applies a global sparsity bias to the first-layer parameter matrix while retaining the original binary classification objective. The global sparsity formulation and its target were selected through preliminary calibration experiments, and we use a target of 3/128 in the following analysis. The full formulation, schedules, and training hyperparameters are deferred to Appendix E.

Using CONV1DFULLY with the global sparsity constraint and input difference $\{117\}$, we train 30 independent 8-round differential-neural distinguishers for 30 epochs. The selected models achieve a mean validation accuracy of 0.81872 ± 0.02280 , with a range of $[0.75039, 0.82896]$, where $\pm$ denotes the standard deviation across runs. The substantially larger variation than in the unconstrained setting indicates that sparsity-guided training is less stable across random seeds. We therefore interpret the subsequent candidate-coverage results at the aggregate level rather than as evidence of uniformly improved models. To separate the effect of sparsity guidance from that of training duration, we use the same 30

unconstrained models trained for 30 epochs and analyzed in Section 4.2 as the matched control.

Effect on Low-Weight Candidate Coverage We first examine the first-layer parameter projections. Applying the same Top-2 projection as in the unconstrained setting, the sparsity-guided models expose 44.37 ± 3.16 of the 77 strong Hamming-weight-2 masks on average, compared with 34.40 ± 1.96 for the training-length-matched unconstrained models. Thus, sparsity guidance substantially increases the coverage of strong Hamming-weight-2 candidates in the first-layer representation.

The classically selected mask $\{77, 5\}$ also becomes more frequently visible. It appears in the Top-2 projection of 9 of the 30 sparsity-guided models, compared with only 1 of the 30 matched unconstrained models. Although this remains far from stable recovery, the increase shows that a mask known to be useful in the classical long-round distinguisher can enter the neural-generated first-layer candidate set substantially more often under sparsity guidance.

Effect on the Final Decision Rule We next examine whether the increased first-layer candidate coverage leads to a corresponding change in the relations emphasized by the final differential-neural decision rule. Using the same Fourier reconstruction setting as in the unconstrained analysis, the sparsity-guided models achieve a reconstruction similarity of $82.10\% \pm 1.76\%$, with a range of $[78.61\%, 85.23\%]$. Table 8 reports the recurrence of the five neural-discovered masks considered in the preceding long-round analysis.

Table 8: Fourier terms retained across 30 sparsity-guided runs of the 8-round Speck128/128 distinguisher

Fourier term S	Freq.	Fourier term S	Freq.
$\{75, 67, 64, 11, 3\}$	27	$\{72, 8\}$	23
$\{75, 72, 67, 64, 11, 8, 3\}$	21	$\{72, 71, 8, 7\}$	17
$\{75, 72, 71, 67, 64, 11, 8, 7, 3\}$	11		

Freq. denotes the number of runs, out of 30, in which the Fourier term is retained by Algorithm 1 with $\tau = 0.1$.

Sparsity guidance changes the recurrence profile of the decision-level Fourier features, but the recurrent terms remain drawn from the same locally strong relations identified in the unconstrained analysis. In particular, $\{75, 67, 64, 11, 3\}$, $\{72, 8\}$, and $\{75, 72, 67, 64, 11, 8, 3\}$ remain among the most recurrent nontrivial terms. By contrast, the classically selected mask $\{77, 5\}$ is not retained by the Fourier reconstruction in any of the 30 sparsity-guided runs, despite appearing in the first-layer Top-2 projection in 9/30 runs.

The sparsity intervention therefore affects candidate generation and neural prioritization differently. At the representation level, it broadens the coverage of strong low-Hamming-weight masks and substantially increases the frequency with which the long-round-useful mask $\{77, 5\}$ enters the first-layer candidate set. At the decision level, however, this increased visibility does not translate into stable recovery or prioritization of $\{77, 5\}$ by the final neural rule; the recurrent Fourier features remain associated with locally strong 8-round relations.

Together with the long-round evaluation above, these results show that sparsity guidance improves the generation of potentially useful sparse candidates but does not by itself resolve the mismatch between fixed-round neural preference and long-round differential-linear utility. In particular, long-round-useful masks can enter the neural-generated candidate set more frequently, while their stable recovery and long-round-aware prioritization remain unresolved.

5 Conclusion

In this work, we investigated whether classical differential-linear features discovered by fixed-round difference-only differential-neural distinguishers can serve as useful candidates for long-round differential-linear analysis. Our experiments show that such distinguishers can provide meaningful short-round candidates. On **Speck32/64**, the dominant differential-linear Fourier feature is preserved after the architectural reorganization introduced in **CONV1DFULLY**. On **SipHash**, the extracted Fourier masks are concentrated among high-correlation classical differential-linear approximations within the exhaustively evaluated low-Hamming-weight space.

The **Speck128/128** case study shows, however, that strong local middle correlations do not necessarily translate into strong long-round utility. Several recurrent neural-discovered masks have substantially stronger 8-round middle correlations than the mask used in the known 18-round classical distinguisher, but the minimum-weight single XOR-linear trails found by the search over the following 5 rounds have considerably weaker correlations. Sparsity guidance improves the generation of low-Hamming-weight candidates and increases the appearance frequency of the classical mask $\{77, 5\}$ in the first-layer Top-2 candidate set from 1/30 to 9/30. Nevertheless, this recovery remains unstable, and the final neural decision rule continues to favor locally stronger relations.

Overall, our results indicate that fixed-round difference-only differential-neural distinguishers can assist long-round candidate generation, while reliable recovery and long-round-aware prioritization remain unresolved. A natural direction for future work is to incorporate downstream cryptanalytic utility more directly into neural feature learning, or to combine neural candidate generation with classical linear extension and structural post-processing in long-round differential-linear search.

References

- [AB12] Jean-Philippe Aumasson and Daniel J. Bernstein. Siphash: A fast short-input PRF. In *INDOCRYPT*, Lecture Notes in Computer Science, pages 489–508. Springer, 2012.
- [BDKW19] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A new tool for differential-linear cryptanalysis. In *EUROCRYPT (1)*, volume 11476 of *Lecture Notes in Computer Science*, pages 313–342. Springer, 2019.
- [Ber08] Daniel J. Bernstein. ChaCha, a variant of Salsa20. In *Workshop Record of SASC 2008: The State of the Art of Stream Ciphers*, pages 273–278, 2008.
- [BGG⁺23] Emanuele Bellini, David Gérardt, Juan Grados, Rusydi H. Makarim, and Thomas Peyrin. Fully automated differential-linear attacks against ARX ciphers. In *CT-RSA*, Lecture Notes in Computer Science, pages 252–276. Springer, 2023.
- [BGPT21] Adrien Benamira, David Gérardt, Thomas Peyrin, and Quan Quan Tan. A deeper look at machine learning-based cryptanalysis. In *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 805–835. Springer, 2021.
- [BLT20] Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to ARX ciphers. In *CRYPTO (3)*, Lecture Notes in Computer Science, pages 329–358. Springer, 2020.

- [BLYZ23] Zhenzhen Bao, Jinyu Lu, Yiran Yao, and Liu Zhang. More insight on deep learning-aided cryptanalysis. In *ASIACRYPT (3)*, volume 14440 of *Lecture Notes in Computer Science*, pages 436–467. Springer, 2023.
- [BSS⁺13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. *IACR Cryptol. ePrint Arch.*, 2013:404, 2013.
- [CBY23] Yi Chen, Zhenzhen Bao, and Hongbo Yu. Differential-linear approximation semi-unconstrained searching and partition tree: Application to lea and speck. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 223–255. Springer, 2023.
- [Goh19] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11692 of *Lecture Notes in Computer Science*, pages 150–179. Springer, 2019.
- [GWH⁺25] Xinxin Gong, Qingju Wang, Yonglin Hao, Lin Jiao, and Xichao Hu. Persistence of hourglass(-like) structure: Improved differential-linear distinguishers for several ARX ciphers. In *ASIACRYPT (1)*, volume 16245 of *Lecture Notes in Computer Science*, pages 317–351. Springer, 2025.
- [NSLL22] Zhongfeng Niu, Siwei Sun, Yunwen Liu, and Chao Li. Rotational differential-linear distinguishers of ARX ciphers with arbitrary output linear masks. In *CRYPTO (1)*, *Lecture Notes in Computer Science*, pages 3–32. Springer, 2022.
- [YWZW26] Yufei Yuan, Wenling Wu, Lei Zhang, and Ruichen Wu. Fourier analysis of neural distinguishers. *IACR Transactions on Symmetric Cryptology*, 2026(1):441–467, Mar. 2026. URL: <https://tosc.iacr.org/index.php/ToSC/article/view/12792>, doi:10.46586/tosc.v2026.i1.441-467.
- [YXZW25] Yufei Yuan, Haiyi Xu, Lei Zhang, and Wenling Wu. Rethinking learning-based symmetric cryptanalysis: a theoretical perspective. *IACR Cryptol. ePrint Arch.*, 2025:1306, 2025.
- [ZYS⁺26] Liu Zhang, Yiran Yao, Danping Shi, Dongchen Chai, Jian Guo, and Zilong Wang. Neural-inspired advances in integral cryptanalysis. In *EUROCRYPT*, volume 16546 of *Lecture Notes in Computer Science*, pages 451–481. Springer, 2026.

A Fourier Reconstruction of Differential-Neural Distinguishers

For completeness, this appendix recalls the Fourier reconstruction procedure of Yuan et al. [YWZW26], which is based on the Goldreich–Levin algorithm. We include the main steps here to make the reconstruction used in our experiments self-contained.

Given a trained differential-neural distinguisher

$$f : \{0, 1\}^d \rightarrow \{0, 1\},$$

where d denotes the dimension of the Boolean input representation, Yuan et al. consider the sign representation

$$g(x) = (-1)^{f(x)} \in \{-1, 1\}.$$

Its Fourier expansion over $\mathbb{F}_2^d$ is

$$g(x) = \sum_{S \subseteq [d]} \widehat{g}(S) \chi_S(x), \quad \chi_S(x) = (-1)^{\sum_{i \in S} x_i},$$

with

$$\widehat{g}(S) = \mathbb{E}_{x \sim U(\mathbb{F}_2^d)} [g(x) \chi_S(x)].$$

The reconstruction retains Fourier coefficients with large absolute value and forms the sparse approximation

$$\bar{g}(x) = \sum_{S: |\widehat{g}(S)| > \tau} \widehat{g}(S) \chi_S(x).$$

The reconstructed sign rule is then defined as

$$\tilde{g}(x) = \text{sgn}(\bar{g}(x)).$$

Since only heavy coefficients are retained and the coefficients are estimated empirically, $\tilde{g}$ should be understood as an approximation of the original sign rule rather than an exact symbolic representation.

In the Goldreich–Levin search used by Yuan et al., a bucket is represented by (k, S) , where

$$S \subseteq \{1, \dots, k\}$$

specifies the support restricted to the first k coordinates. Following [YWZW26], the bucket weight is estimated by

$$w_g(k, S) = \mathbb{E}_{z \sim U(\mathbb{F}_2^{d-k})} \mathbb{E}_{y, y' \sim U(\mathbb{F}_2^k)} [g(y \| z) g(y' \| z) \chi_S(y) \chi_S(y')].$$

Intuitively, $w_g(k, S)$ estimates the total Fourier energy of the coefficients contained in the corresponding bucket. Buckets whose estimated weight exceeds $\tau^2/2$ are retained and recursively split. After the search terminates, the Fourier coefficients of the surviving candidates are estimated by sampling from $U(\mathbb{F}_2^d)$, and only terms satisfying

$$|\widehat{g}(S)| > \tau$$

are retained.

The reconstruction similarity is evaluated on fresh uniformly sampled Boolean inputs from $\mathbb{F}_2^d$, rather than on the neural distinguisher’s training data. The procedure used in our experiments is summarized in Algorithm 1.

Algorithm 1 Empirical Fourier reconstruction of a differential-neural distinguisher

```

1: Input: Boolean distinguisher oracle  $f : \mathbb{F}_2^d \rightarrow \{0, 1\}$ , threshold  $\tau$ 
2: Output: reconstructed sign rule  $\tilde{g}$  and retained Fourier coefficients
3: Define  $g(x) \leftarrow (-1)^{f(x)}$ 
4: Initialize  $B \leftarrow \{(0, \emptyset)\}$ 
5: for  $k = 0$  to  $d - 1$  do
6:    $B_{\text{next}} \leftarrow \emptyset$ 
7:   for all  $(k, S) \in B$  do
8:     for all  $S' \in \{S, S \cup \{k + 1\}\}$  do
9:       Estimate  $\hat{w}_g(k + 1, S')$ 
10:      if  $\hat{w}_g(k + 1, S') > \tau^2/2$  then
11:         $B_{\text{next}} \leftarrow B_{\text{next}} \cup \{(k + 1, S')\}$ 
12:      end if
13:    end for
14:  end for
15:   $B \leftarrow B_{\text{next}}$ 
16: end for
17: for all  $(d, S) \in B$  do
18:   Estimate  $\hat{g}(S) = \mathbb{E}_{x \sim U(\mathbb{F}_2^d)}[g(x)\chi_S(x)]$ 
19: end for
20:  $\mathcal{S}_\tau \leftarrow \{S : (d, S) \in B, |\hat{g}(S)| > \tau\}$ 
21:  $\bar{g}(x) \leftarrow \sum_{S \in \mathcal{S}_\tau} \hat{g}(S)\chi_S(x)$ 
22:  $\tilde{g}(x) \leftarrow \text{sgn}(\bar{g}(x))$ 
23: return  $\tilde{g}$  and  $\{(S, \hat{g}(S)) : S \in \mathcal{S}_\tau\}$ 
    
```

B Detailed Structure of Gohr’s Network

For completeness, we briefly describe the differential-neural network architecture proposed by Gohr [Goh19] for Speck32/64. In Gohr’s original setting, a ciphertext pair (C_0, C_1) is represented as four 16-bit words,

$$(c_{0l}, c_{0r}, c_{1l}, c_{1r}),$$

and arranged as a 4×16 binary matrix.

The network consists of three modules. Module 1 applies a bit-sliced one-dimensional convolution along the four-word dimension, transforming the input into $N_f = 32$ channels, followed by batch normalization and a ReLU activation.

Module 2 consists of a residual tower. Each residual block contains two convolutional layers with kernel size 3 and N_f filters. Each convolution is followed by batch normalization and a ReLU activation, and the block output is combined with its input through a skip connection.

Module 3 is the prediction head, consisting of two fully connected hidden layers with dimensions

$$d_1 = 64, \quad d_2 = 64,$$

followed by a single sigmoid output unit for binary classification.

C Experimental Details for Conv1DFully on Speck32/64

This appendix provides the implementation and training details of the CONV1DFULLY models used in the controlled Speck32/64 experiment of Section 3.3. The experiment considers the 5-round setting derived from the differential-linear approximation in Eq. (2).

Data Generation Training, validation, and test samples are generated using uniformly sampled random keys and plaintexts. For a positive sample, we sample a random key K and a plaintext P , form the pair

$$(P, P \oplus \Delta), \quad \Delta = (0x1000, 0x5000),$$

and encrypt both plaintexts under the same key K for 5 rounds. For a negative sample, we independently and uniformly sample two plaintexts P and P' and encrypt both under the same randomly sampled key K .

The network input is the ciphertext difference

$$C_0 \oplus C_1,$$

represented as a 32-bit binary vector. The training set contains 10^7 samples with equal proportions of positive and negative examples. Independently generated validation and test sets each contain 10^6 samples.

Network Configuration The 32-bit ciphertext-difference input is processed directly along the bit dimension. The first one-dimensional convolution produces $N_f = 1024$ output channels. The resulting features are processed by three fully connected hidden layers with dimensions

$$d_0 = 1024, \quad d_1 = 256, \quad d_2 = 64,$$

followed by a single output neuron for binary classification.

The convolutional layer and each hidden fully connected layer are followed by layer normalization and a ReLU activation. The final layer outputs logits directly, and no sigmoid activation is applied during training.

Training Procedure Each model is trained for 20 epochs using Adam with a fixed learning rate of 10^{-3} . Since the final layer outputs logits directly, we use `BCEWithLogitsLoss` as the training objective. Training is performed with `DataParallel` on 7 GPUs using a batch size of 4000 per GPU, corresponding to an effective batch size of 28000.

At the end of each epoch, the model is evaluated on the independently generated validation set, and the checkpoint with the lowest validation loss is retained. The selected checkpoint is subsequently evaluated on the independent test set.

To assess the stability of the learned features, we repeat the complete CONV1DFULLY training procedure over 10 independent random seeds. For each run, the checkpoint with the lowest validation loss is selected. The validation accuracies and Fourier-analysis results reported in Section 3.3 are aggregated over these selected checkpoints.

Implementation Platform All experiments are implemented in Python 3.9.21 and PyTorch 2.7.1 on Ubuntu 20.04. The hardware platform consists of two Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz processors, 512 GB of RAM, and 7 NVIDIA RTX3080Ti GPUs, each with 12 GB of memory.

D Additional Experiment on ChaCha

This appendix provides an additional qualitative experiment on **ChaCha** to examine whether differential-linear-related statistical structure remains visible in a substantially larger-state setting. Unlike the **Speck32/64** and **SipHash** experiments in the main text, the purpose here is not to establish a strong stability or optimality claim. Instead, we use **ChaCha** as complementary evidence that a difference-only differential-neural distinguisher can capture differential-linear-related information in a more complex ARX setting.

ChaCha [Ber08] is a widely used ARX primitive with a 512-bit internal state consisting of sixteen 32-bit words arranged as a 4×4 matrix. The state contains constant, key, counter, and nonce words. Following the setting of Beierle et al. [BLT20], we consider the non-related-key setting and introduce no differences in the constant or key words.

Let

$$(x_0^r, \dots, x_{15}^r)$$

denote the input state of round r . Each round consists of four parallel applications of the quarter-round transformation shown in Fig. 4. In odd rounds, the quarter round is applied to the four columns, whereas in even rounds it is applied to the four diagonals.

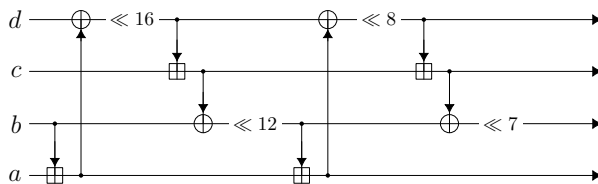


Figure 4: Quarter-round function of ChaCha.

D.1 Experimental Setting

We follow the differential-linear setting of Beierle et al. [BLT20]. Their construction considers a 1-round differential part followed by a 2.5-round middle part. To obtain a low-Hamming-weight difference entering the middle part, they use the following 1-round differential characteristic over one quarter-round function:

$$\begin{aligned} \Delta_{\text{in}} &= (([]), ([[]]), ([[]]), ([i])), \\ \Delta_m &= (([i + 28]), ([i + 31, i + 23, i + 11, i + 3]), \\ &\quad ([i + 24, i + 16, i + 4]), ([i + 24, i + 4])), \end{aligned}$$

where Δ_{in} propagates to Δ_m after one round and bit indices are taken modulo 32.

For $i = 6$, this gives

$$\Delta_m = (([2]), ([5, 29, 17, 9]), ([30, 22, 10]), ([30, 10])).$$

Using this intermediate difference, Beierle et al. [BLT20] experimentally evaluate the following 2.5-round middle part and obtain a differential-linear distinguisher with a single-bit output mask and correlation $2^{-8.3}$. Combined with the preceding 1-round differential characteristic, this yields their 3.5-round differential-linear distinguisher.

In our experiment, we isolate the same 2.5-round middle part and use

$$\Delta = (\{2\}, \{5, 29, 17, 9\}, \{30, 22, 10\}, \{30, 10\})$$

as the fixed input difference of the differential-neural distinguisher. The corresponding classical reference mask is the singleton mask $\{0\}$.

Using CONV1DFULLY, we train a 2.5-round difference-only differential-neural distinguisher on a dataset of size 10^8 for 40 epochs. Repeating the experiment over 10 independent random seeds yields a mean validation accuracy of 0.7988 ± 0.0505 with range $[0.7030, 0.8732]$, where $\pm$ denotes the standard deviation across runs.

The substantially larger variation than in the **Speck32/64** and **SipHash** experiments indicates that this setting is less stable across independent training runs. We therefore treat the following results as qualitative supporting evidence rather than as a basis for strong seed-level conclusions.

D.2 Fourier Analysis

We apply Algorithm 1 directly to the trained CONV1DFULLY distinguishers. With reconstruction threshold $\tau = 0.1$, the 10 runs produce 21 nonconstant heavy Fourier-term occurrences corresponding to 8 distinct masks. Table 9 reports all distinct masks, including terms that appear in only one run.

Table 9: Fourier terms recovered across 10 runs of the 2.5-round ChaCha distinguisher

Fourier term S	Freq.	Corr.
$\{416, 408, 400, 320, 256, 172, 40, 32, 0\}$	6	$2^{-0.86}$
$\{416, 320, 295, 256, 172, 147, 0\}$	5	$2^{-0.14}$
$\{480, 472, 464, 320, 256, 236, 104, 96, 64\}$	3	$2^{-3.84 \times 10^{-5}}$
$\{408, 400, 295, 147, 40, 32\}$	3	$2^{-1.00}$
$\{440, 432, 327, 179, 72, 64\}$	1	$2^{-0.47}$
$\{440, 432, 416, 327, 320, 295, 256, 179, 172, 147, 72, 64, 0\}$	1	$2^{-0.56}$
$\{480, 472, 464, 416, 295, 236, 172, 147, 104, 96, 64, 0\}$	1	$2^{-0.14}$
$\{480, 472, 464, 416, 408, 400, 236, 172, 104, 96, 64, 40, 32, 0\}$	1	$2^{-0.86}$

Corr. denotes the experimentally estimated correlation under the same intermediate difference and the same 2.5-round middle part, using 2^{30} random pairs. Freq. denotes the number of runs, out of 10, in which the corresponding Fourier term is retained.

Several Fourier terms recur across independent runs and exhibit non-negligible empirical differential-linear correlations. In particular, the classical reference bit 0 repeatedly appears as part of larger Fourier supports, including

$$\{416, 408, 400, 320, 256, 172, 40, 32, 0\}$$

and

$$\{416, 320, 295, 256, 172, 147, 0\}.$$

However, $\{0\}$ does not appear as a singleton Fourier term. Thus, in this experiment, the statistical information associated with the classical reference bit is exposed only as part of larger parity features rather than as an isolated singleton mask.

Some terms that occur in only one run also exhibit large empirical correlations. For example,

$$\{480, 472, 464, 416, 295, 236, 172, 147, 104, 96, 64, 0\}$$

has correlation $2^{-0.14}$ but appears in only one run. This shows that a Fourier term may correspond to a strong local differential-linear approximation without being recovered consistently across independently trained differential-neural distinguishers.

D.3 Interpretation and Limitations

The ChaCha experiment provides qualitative supporting evidence that differential-linear-related parity structure remains visible in a larger-state ARX setting. The reconstructed decision rules contain recurrent Fourier features with substantial empirical differential-linear correlations, including several whose supports contain the classical reference bit. At the same time, these features appear in a more combined form than in the controlled Speck32/64 and SipHash experiments.

The training outcomes also exhibit substantial variation across random seeds. We therefore do not interpret the ChaCha experiment as evidence of stable recovery of a particular

classical mask, nor does it establish optimality of the extracted approximations. Rather, it serves as complementary qualitative evidence that differential-neural distinguishers can capture differential-linear-related statistical structure in a substantially larger-state setting.

E Sparsity-Constrained Training Details for Speck128/128

This appendix provides the details of the sparsity-constrained training procedure used in Section 4.4. The purpose of the sparsity intervention is to increase the visibility of low-Hamming-weight differential-linear candidates in the first convolutional layer, rather than to optimize the differential-neural distinguisher specifically for the classically selected mask $\{77, 5\}$.

Using CONV1DFULLY with input difference $\{117\}$, we train an 8-round differential-neural distinguisher for **Speck128/128**. The first convolutional layer contains 1024 channels and is parameterized by

$$W \in \mathbb{R}^{1024 \times 128}.$$

The standard binary classification objective is augmented with a sparsity regularizer acting only on this first-layer parameter matrix:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{sp}}(e) b_{\text{sp}}(e) \mathcal{L}_{\text{sp}}(W),$$

where $\mathcal{L}_{\text{cls}}$ denotes the ordinary classification loss, $\lambda_{\text{sp}}(e)$ controls the overall strength of sparsity regularization at epoch e , and $b_{\text{sp}}(e)$ adaptively amplifies the penalty when the current representation remains denser than the target.

Soft Activity and Sparsity Regularization To estimate the number of active positions in a differentiable manner, each entry of W is mapped to a baseline-corrected soft activity value:

$$A_{ij} = \text{clip}_{[0,1]} \left(\frac{\sigma(k(|W_{ij}| - \tau_w)) - \sigma(-k\tau_w)}{1 - \sigma(-k\tau_w) + \epsilon} \right),$$

where $\sigma(\cdot)$ is the sigmoid function, k controls the sharpness of the transition, and τ_w is the weight-magnitude threshold. The resulting matrix

$$A \in [0, 1]^{1024 \times 128}$$

provides a differentiable approximation to a binary activity pattern.

We define the soft active ratio as

$$\rho_{\text{act}} = \text{mean}(A),$$

which measures the average fraction of active positions over the first-layer parameter matrix.

The sparsity regularizer consists of an L_1 term and a target-ratio penalty:

$$\mathcal{L}_{\text{sp}} = w_{L1} \mathcal{L}_{L1} + \mathcal{L}_{\text{target}},$$

where

$$\mathcal{L}_{L1} = \text{mean}(|W|)$$

and

$$\mathcal{L}_{\text{target}} = \alpha \max(\rho_{\text{act}} - \rho_t(e), 0) + \beta \max(\rho_t(e) - \rho_{\text{act}}, 0).$$

The first term penalizes the overall magnitude of the first-layer weights, while the second penalizes deviations of the current soft active ratio from the target ratio $\rho_t(e)$. The

asymmetric coefficients α and β allow over-dense and over-sparse representations to be penalized differently.

The final target active ratio is parameterized by the desired layer-wide average number of active positions per filter:

$$\rho_{\text{final}} = \frac{\text{max\_hw}}{128}.$$

In our main target-3 experiments, we set

$$\text{max\_hw} = 3.$$

This choice controls only the aggregate target sparsity level of the first-layer representation and does not directly reward or penalize any particular bit positions or mask identities. The target-3 setting is a calibration condition; it does not encode the identity or the Hamming weight of the classical mask $\{77, 5\}$.

Dynamic Sparsity Scheduling Imposing the final sparsity level from the beginning can interfere with the initial learning of discriminative features. We therefore introduce the sparsity constraint gradually during training.

First, the target active ratio is annealed according to

$$\rho_t(e) = \rho_0 - (\rho_0 - \rho_{\text{final}}) \text{progress}(e),$$

where ρ_0 is the initial target active ratio and

$$\text{progress}(e) \in [0, 1]$$

is a monotone schedule variable. Thus, the desired representation becomes progressively sparser as training proceeds.

Second, the coefficient of the sparsity term is increased in stages:

$$\lambda_{\text{sp}}(e) = \begin{cases} \lambda_A, & 0 \leq e < e_A, \\ \lambda_A + \frac{e - e_A}{e_{A2B}} (\lambda_B - \lambda_A), & e_A \leq e < e_A + e_{A2B}, \\ \lambda_B, & e \geq e_A + e_{A2B}. \end{cases}$$

Stage A lasts for e_A epochs, followed by an A2B transition of e_{A2B} epochs. After the transition, Stage B lasts for e_B epochs with

$$\lambda_{\text{sp}}(e) = \lambda_B.$$

This schedule keeps the sparsity pressure relatively mild during the initial training stage and strengthens it after the classifier has learned useful distinguishing information.

Third, when the current soft active ratio remains above the target, the sparsity penalty is adaptively amplified using

$$b_{\text{sp}}(e) = \text{clip}\left(\frac{\rho_{\text{act}}}{\rho_t(e) + \varepsilon}, b_{\text{min}}, b_{\text{max}}\right),$$

where ε is a small constant used for numerical stability. Hence, if the first-layer representation remains substantially denser than the current target, the effective sparsity penalty is increased.

Taken together, these components allow the classifier to first learn useful discriminative information and then progressively bias the first-layer representation toward sparse structures.

Training Configuration The sparsity-constrained models are trained on 10^7 training samples under input difference $\{117\}$. The remaining network architecture and optimization settings follow those of the unconstrained CONV1DFULLY models unless stated otherwise. The hyperparameters specific to sparsity-constrained training are summarized in Table 10.

Table 10: Hyperparameters of the global sparsity-guided target-3 training configuration.

Symbol	Description	Value
e_A, e_{A2B}, e_B	Numbers of epochs in Stage A, the A2B transition, and Stage B	10, 10, 10
<code>num_filters</code>	Number of channels in the first Conv1D layer	1024
<code>max_hw</code>	Final global target, expressed as the average number of active positions per first-layer channel	3
$\rho_0, \rho_{\text{final}}$	Initial and final target active ratios	0.1, 3/128
k, τ_w	Sharpness parameter and weight-magnitude threshold of the soft activity function	100, 0.01
w_{L_1}	Base coefficient of the L_1 regularization term	0.01
α, β	Penalty coefficients for ratios above and below the target, respectively	1.5, 3.0
$b_{\min}, b_{\max}$	Lower and upper bounds of the sparsity boost factor	1, 5
λ_A, λ_B	Sparsity coefficients used in Stage A and Stage B	0.8, 1.5