Local Constraints Behind Fourier Analysis of Neural Distinguishers for SPECK32/64

Yunjae Hwang[1], Sunyeop Kim[1,2], Hanbeom Shin[1], Deukjo Hong[3], Seokhie Hong[4], Dongjae Lee[5], Jaechul Sung[6] and Byoungjin Seok[7]

[1] Korea University, Seoul, South Korea
[2] Nanyang Technological University, Jurong, Singapore
[3] Jeonbuk National University, Jeonju, South Korea
[4] Smartm2m, Busan, South Korea
[5] Kangwon National University, Chuncheon, South Korea
[6] University of Seoul, Seoul, South Korea
[7] Hansung University, Seoul, South Korea

Abstract

Neural distinguishers for ARX ciphers can exploit information beyond classical difference distributions and several interpretability frameworks have been proposed. In this paper, we study two frameworks for `SPECK32/64` by connecting their viewpoints: local constraints of modular addition and Fourier analysis of trained neural distinguishers. We show that the dominant Fourier parities of a raw-pair differential neural distinguisher can be rewritten in the local variables associated with the last modular addition. This representation separates value-dependent variant differential-linear terms from difference-dependent traditional terms, and explains their biases through specific local constraints and branch effects. We further extend the analysis to a boomerang right-quartet setting. We construct a neural distinguisher whose input is only the original ciphertext pair, while positive and negative samples are matched with respect to the observed ciphertext difference. Fourier analysis of this distinguisher reveals dominant value-dependent parities. We trace these terms to borrow synchronization in the first inverse step of the lower boomerang characteristic, yielding specific local conditions. Our results indicate that the dominant Fourier features learned in these settings are observable projections of concrete carry or borrow constraints of the ARX operation.

Keywords: Neural distinguisher, Fourier analysis, `SPECK32/64`, Boomerang distinguisher

1 Introduction

Differential cryptanalysis and its variants are among the central tools for evaluating block ciphers. For ARX designs, the nonlinear operation is modular addition and differential, linear, and boomerang properties are often governed by carry or borrow propagation across bit positions. In particular, the lightweight block cipher `SPECK` [BTCS+15] has been extensively studied by classical differential techniques and automatic ARX trail search methods [BV14, Din14].

Emails: hyj019@korea.ac.kr (Yunjae Hwang), kin3548@gmail.com (Sunyeop Kim), newonetiger@korea.ac.kr (Hanbeom Shin), deukjo.hong@jbnu.ac.kr (Deukjo Hong), shhong@smartm2m.co.kr (Seokhie Hong), dongjae.lee@kangwon.ac.kr (Dongjae Lee), jcsung@uos.ac.kr (Jaechul Sung), bjseok@hansung.ac.kr (Byoungjin Seok)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2026-06-02.

A new line of research was initiated by Gohr, showing that deep residual networks can be trained as differential distinguishers for round-reduced `SPECK32/64` and can be integrated into competitive key-recovery attacks [Goh19]. In the standard differential-neural setting, the classifier is given a ciphertext pair and decides whether it originates from plaintexts with a fixed input difference or from random data. A key observation of Gohr's work is that neural distinguishers can outperform pure differential baselines that are based on the differential distribution table (DDT).

Several works have sought to explain this phenomenon. Benamira *et al.* analyzed Gohr's neural distinguisher and related its behavior to the biases of truncated differences in the last rounds, providing a more interpretable cryptanalysis-based approximation of the neural decision rule [BGPT21]. Bao *et al.* gave a more explicit local explanation for the advantage of differential-neural distinguishers on `SPECK32/64` [BLYZ23]. Their analysis exploits the fact that from a raw ciphertext pair (C, C'), one can compute not only the ciphertext difference but also the partially known values associated with the last modular addition. These values can reveal whether certain carry-dependent local constraints of the last modular addition are likely to be fulfilled. Thus, these local constraints provide a value-conditioned explanation for features that are invisible to a pure DDT-based distinguisher.

A complementary global interpretation was proposed by Yuan *et al.* through Fourier analysis of neural distinguishers [YWZW26]. In this approach, a trained neural distinguisher is regarded as a Boolean function on the ciphertext-pair bits, and large Fourier coefficients identify parities strongly correlated with the neural decision. For a ciphertext pair (C, C'), such a parity has the form

$$T_{\Gamma,\Gamma'}(C, C') = \langle \Gamma, C \rangle \oplus \langle \Gamma', C' \rangle.$$

If $\Gamma = \Gamma'$, the parity depends only on $C \oplus C'$ and corresponds to a traditional differential-linear (DL) feature, in the sense of differential-linear cryptanalysis [LH94, BLN17, BODKW19]. If $\Gamma \neq \Gamma'$, the parity cannot in general be reduced to a function of the ciphertext difference alone; Yuan *et al.* interpret such terms as variant differential-linear features. This terminology is also compatible with connectivity-table based approaches to differential-linear dependencies, such as EDLCT [CY21].

The first goal of this paper is to connect the local-constraint interpretation with the Fourier representation in the same coordinate system. These two viewpoints operate at different levels: Bao's analysis explains local constraints of the last modular addition in terms of the known local variables, whereas Fourier analysis extracts global parities of the raw ciphertext-pair bits. We show that the dominant Fourier terms of a raw-pair neural distinguisher for `SPECK32/64` can be rewritten in the same local variables.

The second goal is to ask whether an analogous value-dependent phenomenon arises in other settings. Boomerang attacks, introduced by Wagner [Wag99], combine two short differential characteristics rather than a single characteristic across the whole cipher. Subsequent work showed that the probability of a boomerang distinguisher depends critically on the compatibility of the upper and lower characteristics near their boundary, leading to the sandwich framework and the Boomerang Connectivity Table (BCT) [DKS10, CHP+18]. For ARX ciphers, BCT-type analysis is more delicate because modular addition acts on large words. A recent SAT-aided method makes it possible to compute and search boomerang properties for ARX ciphers, including 10-round boomerang distinguishers for `SPECK32/64` [WWS23].

Classical boomerang distinguishers are not ciphertext-pair classifiers in the same sense as differential-neural distinguishers: the relevant event is a right-quartet condition verified through an encryption–decryption experiment. This motivates the question whether conditioning on the right-quartet event leaves residual statistical information in the original ciphertext pair (C, C'). To isolate such information, we consider a boomerang neural

distinguisher whose input is only the original ciphertext pair, while the label is determined by the right-quartet condition. The positive and negative datasets are additionally matched with respect to the observed output difference $C \oplus C'$. Therefore, the task isolates information in the raw pair beyond the marginal output-difference distribution.

Our contributions are as follows:

- We give an algebraic bridge between Fourier terms of raw-pair neural distinguishers and the local variables used in the local-constraint interpretations of `SPECK32/64`.
- We refine the interpretation of the Fourier spectrum of Gohr's 7-round `SPECK32/64` neural distinguisher by explaining the biases of the extracted Fourier terms in terms of local variables.
- We introduce a boomerang right-quartet neural distinguisher experiment for `SPECK32/64` under matched output-difference distributions, thereby isolating value-dependent information induced by the boomerang event.
- We explain the dominant Fourier terms of the boomerang right-quartet neural distinguisher through a local borrow-chain analysis of the lower inverse boomerang transition, identifying the borrow-chain mechanism behind the observed value-dependent Fourier parities.

Organization. Section 2 recalls the notation and other necessary preliminaries. Section 3 reviews the two different frameworks for analyzing differential neural distinguishers: the local-constraint interpretation and Fourier analysis. Section 4 connects the local-constraint and Fourier viewpoints for differential neural distinguishers both algebraically and experimentally. Section 5 introduces and analyzes the boomerang right-quartet neural distinguisher through our framework. Finally, Section 6 concludes the paper.

The core source code of our experiment is available at an anonymous link.

2 Preliminaries

2.1 Notation

We denote by $\mathbb{F}_2$ the binary field and by $\mathbb{F}_2^n$ the set of n-bit words. For an n-bit word x, we write $x[i]$ for its i-th bit where $x[0]$ denotes the LSB. Left and right rotations by s bits are denoted by $\lll s$ and $\ggg s$, respectively. Modular addition and subtraction of two m-bit words are denoted by $\boxplus$ and $\boxminus$, where the operation is performed modulo 2^m. For $a, x \in \mathbb{F}_2^n$, the binary inner product is defined as $\langle a, x \rangle = \bigoplus_{i=0}^{n-1} a[i]x[i]$. We denote by $E_K^{(r)}$ the r-round encryption under key K. An n-bit plaintext and ciphertext are written as $P = (P_L, P_R), C = (C_L, C_R) = E_K^{(r)}(P)$, where each component is a $n/2$-bit word.

2.2 SPECK32/64

`SPECK` is a family of lightweight ARX block ciphers [BTCS+15]. In this work, we focus on its specific variant `SPECK32/64` which has a 32-bit block, a 64-bit master key, and 22 rounds. The internal state consists of two 16-bit words. For `SPECK32/64`, one round with subkey $k^r \in \mathbb{F}_2^{16}$ is defined by

$$C_L^{r+1} = ((C_L^r \ggg 7) \boxplus C_R^r) \oplus k_r, \quad C_R^{r+1} = (C_R^r \lll 2) \oplus C_L^{r+1},$$

where $\boxplus$ is performed modulo 2^{16}.

The master key of `SPECK32/64` consists of four 16-bit words

$$K = (\ell^2, \ell^1, \ell^0, k^0).$$

The remaining round subkeys are generated by

$$\ell^{r+3} = ((\ell^r \ggg 7) \boxplus k^r) \oplus r, \quad k^{r+1} = (k^r \lll 2) \oplus \ell^{r+3},$$

for $0 \le r \le 20$.

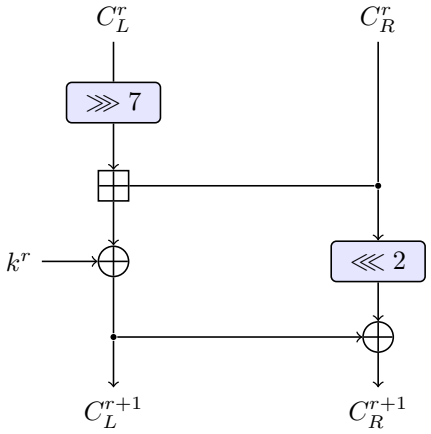


Figure 1: The round function of `SPECK32/64`

2.3 Differential Neural Distinguishers

A differential neural distinguisher is a supervised classifier trained to distinguish ciphertext pairs generated from a prescribed input difference from random ciphertext pairs. This paradigm was introduced by Gohr in the context of `SPECK32/64` and became the standard setting for differential neural cryptanalysis [Goh19].

Let $\Delta_{\text{in}} \in \mathbb{F}_2^n$ be a fixed input difference of an n-bit plaintext pair. We consider two distributions over ciphertext pairs. The real distribution is defined by

$$\mathcal{D}_{\Delta_{\text{in}}}^{(r)} = \left\{ \left(E_K^{(r)}(P), E_K^{(r)}(P \oplus \Delta_{\text{in}}) \right) : K \leftarrow \mathcal{K},\ P \leftarrow \mathbb{F}_2^n \right\}.$$

The random distribution is

$$\mathcal{U}^{(r)} = \left\{ \left(E_K^{(r)}(P), E_K^{(r)}(P') \right) : K \leftarrow \mathcal{K},\ P, P' \leftarrow \mathbb{F}_2^n \right\},$$

where P and P' are sampled independently.

A differential neural distinguisher is a parameterized function

$$\mathcal{N}_\theta : \mathbb{F}_2^{2n} \to [0, 1],$$

where the input is a ciphertext pair (C, C') and the output is interpreted as the score that the pair comes from the real distribution. Training samples are generated with binary labels

$$b = \begin{cases} 1, & \text{if } (C, C') \leftarrow \mathcal{D}_{\Delta_{\text{in}}}^{(r)}, \\ 0, & \text{if } (C, C') \leftarrow \mathcal{U}^{(r)}. \end{cases}$$

For `SPECK32/64`, Gohr's original distinguisher uses the input difference

$$\Delta_{\text{in}} = (\texttt{0x0040}, \texttt{0x0000}),$$

and feeds the raw ciphertext pair

$$(C, C') = ((C_L, C_R), (C'_L, C'_R))$$

to a residual neural network. The raw-pair representation preserves more information than the XOR difference $C \oplus C'$ alone. A difference-only representation restricts the classifier to features depending only on the output difference, whereas a raw-pair representation may also exploit value-dependent relations between the two ciphertexts.

2.4 Boomerang Attack

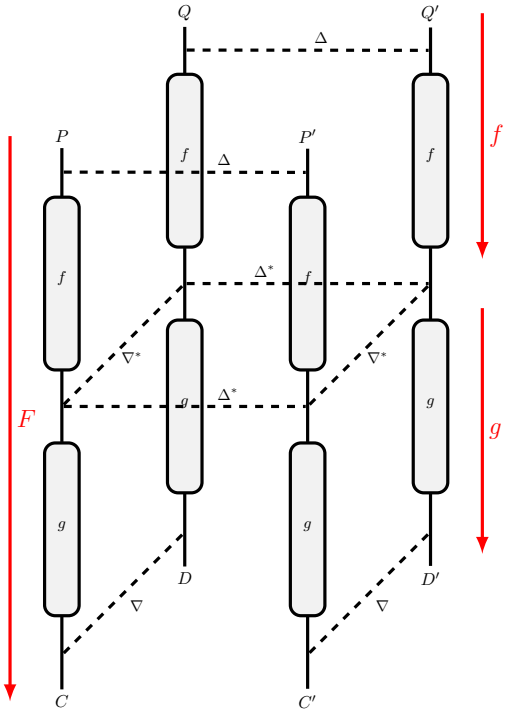


Figure 2: The standard boomerang attack

The boomerang attack is a differential-style distinguisher that combines two short differential characteristics instead of relying on a single characteristic of the whole cipher [Wag99]. Let

$$E = E_1 \circ E_0$$

and suppose that E_0 admits a differential characteristic $\Delta \xrightarrow{p} \Delta^\star$, while E_1 admits a characteristic $\nabla^\star \xrightarrow{q} \nabla$. Given a plaintext pair $P' = P \oplus \Delta$, let

$$C = E(P), \qquad C' = E(P').$$

The boomerang experiment applies the ciphertext-side difference ∇, decrypts

$$Q = E^{-1}(C \oplus \nabla), \qquad Q' = E^{-1}(C' \oplus \nabla),$$

and tests whether

$$Q \oplus Q' = \Delta.$$

A quartet satisfying this condition is a right boomerang quartet. Under the usual independence assumption for the four uses of the two characteristics, the right-quartet probability is estimated as

$$p^2 q^2,$$

whereas a random n-bit permutation satisfies the final difference condition with probability 2^{-n}.

The estimate p^2q^2 can be inaccurate when the two characteristics interact at their boundary. This is formalized by the sandwich framework

$$E = E_1 \circ E_m \circ E_0,$$

where the middle part E_m captures the compatibility of the upper and lower characteristics [DKS10]. If the middle boomerang switch has probability r, the boomerang probability is estimated as

$$p^2rq^2.$$

Thus the middle part may either invalidate the concatenation when $r = 0$, or increase the probability through switching effects.

When E_m is an invertible m-bit S-box S, the middle switch is measured by BCT [CHP+18]:

$$\mathcal{T}(a,b) = \#\left\{x \in \mathbb{F}_2^m : S^{-1}(S(x) \oplus b) \oplus S^{-1}(S(x \oplus a) \oplus b) = a\right\}.$$

$\mathcal{T}(a,b)/2^m$ is the local boomerang-switch probability for the input side difference a and the output side difference b. For ARX ciphers, the analogous analysis is more involved because modular addition acts on large words. Recent ARX boomerang analyses therefore compute or encode BCT-type quantities for modular addition, often using dynamic programming or SAT-aided search models [WWS23].

3 Related Work

3.1 Local-Constraint Interpretation on Differential Neural Distinguishers

Bao *et al.* investigated how differential neural distinguishers for `SPECK32/64` can outperform distinguishers based on the full DDT [BLYZ23]. Their main observation is that a neural distinguisher fed with raw ciphertext pairs is not restricted to the output difference $C \oplus C'$ but it can also take advantage of value-dependent information that remains available in the ciphertext pair. In `SPECK`, this additional information is closely related to the last modular addition.

For a pure DDT-based distinguisher, only Inter-XOR information $(C_L \oplus C'_L, C_R \oplus C'_R)$ is directly used since it fully depends on the ciphertext difference. On the other hand, Intra-XOR $(C_L \oplus C_R, C'_L \oplus C'_R)$ exploits the last round equation of `SPECK32/64`:

$$(C_L^r \oplus C_R^r) \ggg 2 = C_R^{r-1}.$$

This implies that the Intra-XOR terms reveal the right input word of the last round without information of the last-round subkey. For convenience, they defined

$$x := C_L^{r-1} \ggg 7, \qquad y := C_R^{r-1}, \qquad z := x \boxplus y.$$

For a pair of executions, they also defined

$$\alpha := \Delta x, \qquad \beta := \Delta y, \qquad \gamma := \Delta z.$$

In `SPECK32/64`, for a ciphertext pair $((C_L, C_R), (C'_L, C'_R))$, the quantities

$$y = (C_L \oplus C_R) \ggg 2, \qquad y' = (C'_L \oplus C'_R) \ggg 2$$

are directly computable from the ciphertext pair. Hence

$$\beta = y \oplus y' = ((C_L \oplus C_R) \oplus (C'_L \oplus C'_R)) \ggg 2$$

is known. Moreover, since the last subkey is XORed after the modular addition and cancels in the pair difference,

$$\gamma = C_L \oplus C'_L$$

is also known. In contrast, α is not directly observable from the ciphertext pair as it depends on the other input word of the last modular addition. Nevertheless, for positive samples generated from a fixed input difference, α may be biased and can therefore influence the distribution seen by the distinguisher.

A conventional DDT-based distinguisher averages over all input values compatible with a given difference transition. Bao *et al.* observed that this averaging discards useful information since the validity of the modular-addition transition can depend on one-bit constraints involving the concrete values of $(\alpha, \beta, \gamma, y)$. Consequently, the tuple (β, γ, y) contains value-dependent information beyond the output difference alone. This gives a cryptanalytic explanation for why a neural distinguisher receiving raw ciphertext pairs may outperform a distinguisher based only on the DDT. The detailed one-bit constraint that we use for our analysis is recalled in Appendix A.

3.2 Fourier Analysis on Differential Neural Distinguishers

Yuan *et al.* proposed to analyze differential neural distinguishers through their Fourier representation [YWZW26]. The central idea is to abstract away from the internal architecture of the neural network and regard a trained neural distinguisher as a Boolean oracle on ciphertext-pair inputs. This makes it possible to ask which parity combinations of the input bits are most correlated with the decision of the distinguisher. Let $\mathcal{N}$ be a trained distinguisher on inputs $X = C\|C'$. We obtain a Boolean function

$$f_{\mathcal{N}} : \mathbb{F}_2^{2n} \to \{0, 1\}.$$

Using the sign representation $g_{\mathcal{N}}(X) = (-1)^{f_{\mathcal{N}}(X)}$, one has

$$g_{\mathcal{N}}(X) = \sum_{a \in \mathbb{F}_2^{2n}} \hat{g}_{\mathcal{N}}(a)(-1)^{\langle a, X\rangle}.$$

A large coefficient $|\hat{g}_{\mathcal{N}}(a)|$ means that the parity $\langle a, X\rangle$ is strongly correlated with the neural decision. Since enumerating all masks is infeasible, Yuan *et al.* used the Parseval theorem and the Goldreich–Levin algorithm [O'D21, GL89] to find masks with large Fourier coefficients and to reconstruct the distinguisher from a sparse set of parity terms.

For a ciphertext pair input X, write the Fourier mask as

$$a = (\Gamma, \Gamma'), \qquad \Gamma, \Gamma' \in \mathbb{F}_2^n.$$

The corresponding parity is

$$T_{\Gamma,\Gamma'}(C, C') = \langle \Gamma, C\rangle \oplus \langle \Gamma', C'\rangle.$$

If $\Gamma = \Gamma'$, then

$$T_{\Gamma,\Gamma}(C, C') = \langle \Gamma, C \oplus C'\rangle,$$

so the term depends only on the ciphertext difference. This is a *traditional DL distinguisher* and for ARX ciphers, this differential-linear behavior of modular addition has been studied directly [NSLL22]. If $\Gamma \neq \Gamma'$, the two ciphertexts are tested with different masks, and the term cannot in general be reduced to a mask on $C \oplus C'$. They interpret such terms as *variant DL distinguishers*.

4 Connecting the Local-Constraint Interpretation and Fourier Analysis

The local-constraint interpretation and Fourier analysis provide two different views of differential neural distinguishers. The former explains value-dependent effects of the last modular addition, while the latter identifies parity features correlated with the neural decision. In this section, we show that the dominant Fourier terms of Gohr's 7-round raw-pair neural distinguisher, denoted by $\mathcal{ND}_7^{\text{Gohr}}$, can be rewritten in the local variables used in the local-constraint interpretation. We train the neural distinguisher for 7-round `SPECK32/64` following the Gohr's training process using 10^7 training samples and 10^6 test samples. We then apply Fourier analysis to the trained network with $\tau = 0.01$. Among 83 terms of the Fourier representation under $\tau = 0.01$, we consider top 20 Fourier terms with their absolute value of the Fourier coefficient for our investigation. Tables 1 and 2 show the overall results of the distinguisher $\mathcal{ND}_7^{\text{Gohr}}$. Note that Yuan *et al.* further introduced a different type of differential neural distinguisher that is trained by differences of ciphertexts $C \oplus C'$ as input data. We also reconstruct such a distinguisher $\mathcal{ND}_7^{\text{Yuan}}$ and the corresponding results are provided in Appendix B.

Table 1: Accuracy of raw $\mathcal{ND}_7^{\text{Gohr}}$ and the corresponding result of Fourier analysis.

τ	#Terms	ACC
0.01	83	0.598
0	-	0.611

Table 2: Top 20 Fourier terms of $\mathcal{ND}_7^{\text{Gohr}}$ sorted by the absolute value of the Fourier coefficient.

ID	Γ	Γ'	Coefficient
1	0x58204020	0x58204020	−0.1744
2	0x2c102010	0x2c102010	−0.1730
3	0x70206020	0x70206020	−0.1729
4	0x38103010	0x38103010	−0.1707
5	0x50304030	0x50304030	−0.1354
6	0x78306030	0x78306030	−0.1351
7	0x28102010	0x28102010	−0.1346
8	0x3c103010	0x3c103010	−0.1268
9	0x16001000	0x16001000	−0.1172
10	0x14001000	0x14001000	−0.1154
11	0x1e001800	0x1e001800	−0.1145
12	0x1c001800	0x1c001800	−0.1016
13	0x04000000	0x04000000	0.0950
14	0x102c1020	0x102c1020	−0.0913
15	0x10381030	0x10381030	−0.0897
16	0x70206020	0x40205020	−0.0890
17	0x58204020	0x68207020	0.0887
18	0x68207020	0x58204020	0.0868
19	0x40205020	0x70206020	−0.0866
20	0x2c102010	0x34103810	0.0804

4.1 Framework for Analysis on Neural Distinguishers

The top 15 terms are all associated with traditional DL distinguishers which apply the same linear mask $\Gamma = \Gamma'$ on C and C'

$$\langle \Gamma, C \rangle \oplus \langle \Gamma', C' \rangle = \langle \Gamma, C \oplus C' \rangle,$$

whereas the case of $\Gamma \neq \Gamma'$ is called *variant DL distinguishers*

$$\langle \Gamma, C \rangle \oplus \langle \Gamma', C' \rangle \neq \langle \Gamma, C \oplus C' \rangle.$$

The parity $T_{\Gamma,\Gamma'}$ can be written in a more general way if we split Γ into two words $\Gamma = (u, v)$:

$$\begin{aligned} T_{\Gamma,\Gamma'} =& \langle (u, v), (C_L, C_R) \rangle \oplus \langle (u', v'), (C'_L, C'_R) \rangle \\ =& \langle u, C_L \rangle \oplus \langle u', C'_L \rangle \oplus \langle v, C_R \rangle \oplus \langle v', C'_R \rangle. \end{aligned}$$

In the notation of the local-constraint interpretation, we have

$$y = (C_L \oplus C_R) \ggg 2, \quad y' = (C'_L \oplus C'_R) \ggg 2.$$

Applying the inverse rotation $\lll 2$ yields

$$C_R = C_L \oplus (y \lll 2), \quad C'_R = C'_L \oplus (y' \lll 2).$$

Substituting these expressions into the Fourier terms gives

$$T_{\Gamma,\Gamma'} = \langle u, C_L \rangle \oplus \langle u', C'_L \rangle \oplus \langle v, C_L \oplus (y \lll 2) \rangle \oplus \langle v', C'_L \oplus (y' \lll 2) \rangle.$$

By the linearity of the inner product over $\mathbb{F}_2$,

$$T_{\Gamma,\Gamma'} = \langle u \oplus v, C_L \rangle \oplus \langle u' \oplus v', C'_L \rangle \oplus \langle v, y \lll 2 \rangle \oplus \langle v', y' \lll 2 \rangle.$$

Using the duality of the rotation, we can write

$$T_{\Gamma,\Gamma'} = \langle u \oplus v, C_L \rangle \oplus \langle u' \oplus v', C'_L \rangle \oplus \langle v \ggg 2, y \rangle \oplus \langle v' \ggg 2, y' \rangle.$$

It should be noted that all 20 terms in Table 2 satisfy the condition

$$u \oplus v = u' \oplus v'.$$

We therefore define three useful terms

$$\begin{aligned} \mu &:= u \oplus v = u' \oplus v', \\ \nu &:= v' \ggg 2, \\ \eta &:= (v \oplus v') \ggg 2. \end{aligned}$$

Finally, we can get the equation of the parity compatible with the local-constraint interpretation

$$\begin{aligned} T_{\Gamma,\Gamma'} &= \langle \mu, \gamma \rangle \oplus \langle v \ggg 2, y \rangle \oplus \langle v' \ggg 2, y \oplus \beta \rangle \\ &= \langle \mu, \gamma \rangle \oplus \langle v' \ggg 2, \beta \rangle \oplus \langle (v \oplus v') \ggg 2, y \rangle \\ &= \langle \mu, \gamma \rangle \oplus \langle \nu, \beta \rangle \oplus \langle \eta, y \rangle. \end{aligned} \tag{1}$$

Table 3: Top variant DL distinguishers rewritten in local-constraint variables.

ID	Γ	Γ'	(μ, ν, η)	(β, γ, y)
16	0x70206020	0x40205020	(0x1000, 0x1408, 0x0c00)	$\beta_{12} \oplus \beta_{10} \oplus \beta_3 \oplus \gamma_{12} \oplus y_{11} \oplus y_{10}$
17	0x58204020	0x68207020	(0x1800, 0x1c08, 0x0c00)	$\beta_{12} \oplus \beta_{11} \oplus \beta_{10} \oplus \gamma_{12} \oplus \gamma_{11} \oplus \beta_3 \oplus y_{11} \oplus y_{10}$
18	0x68207020	0x58204020	(0x1800, 0x1008, 0x0c00)	$\beta_{12} \oplus \beta_3 \oplus \gamma_{12} \oplus \gamma_{11} \oplus y_{11} \oplus y_{10}$
19	0x40205020	0x70206020	(0x1000, 0x1808, 0x0c00)	$\beta_{12} \oplus \beta_{11} \oplus \beta_3 \oplus \gamma_{12} \oplus y_{11} \oplus y_{10}$
20	0x2c102010	0x34103810	(0x0c00, 0x0e04, 0x0600)	$\beta_{11} \oplus \beta_{10} \oplus \beta_9 \oplus \beta_2 \oplus \gamma_{11} \oplus \gamma_{10} \oplus y_{10} \oplus y_9$

4.2 Variant DL Distinguishers and C3 Triggers

We first revisit the variant DL distinguishers in Table 2 through local-constraint variables. A variant DL term has $\Gamma \neq \Gamma'$ and in our top-20 list, this always produces a nonzero mask η on the known value y. Table 3 gives the resulting decomposition for the five variant terms ID16–ID20.

Table 3 provides the local-variable form of the variant Fourier terms and is the basis for comparing them with the one-bit local constraints. The terms ID16–ID19 all contain the combination

$$\gamma_{12} \oplus \beta_{12} \oplus y_{11} \oplus y_{10},$$

whereas ID20 contains

$$\gamma_{11} \oplus \beta_{11} \oplus y_{10} \oplus y_9.$$

These combinations match the observable (β, γ, y)-components that arise in the C3 case of the local-constraint interpretation. The structural similarity motivated the following experiment: Bao *et al.* showed that for positive samples, the biased distribution of the hidden difference α can interact with one-bit modular-addition constraints so that certain observable combinations of (β, γ, y) become biased on the corresponding local branches represented by Equation 5. We therefore test whether biases of the variant DL distinguishers are concentrated on the event $\mathsf{C3}_{11}$ and $\mathsf{C3}_{10}$ for ID16–ID19 and ID20 respectively. For each variant term T_j, we define the C3 event to be tested as

$$E_j = \begin{cases} \mathsf{C3}_{11}, & j \in \{16, 17, 18, 19\}, \\ \mathsf{C3}_{10}, & j = 20. \end{cases}$$

We also define the favored value ε_j of each parity according to the sign of the corresponding Fourier coefficient; with our sign convention, a negative coefficient favors $T_j = 0$, while a positive coefficient favors $T_j = 1$. Hence,

$$\varepsilon_{16} = 0, \quad \varepsilon_{17} = 1, \quad \varepsilon_{18} = 1, \quad \varepsilon_{19} = 0, \quad \varepsilon_{20} = 1.$$

For $b \in \{0, 1\}$, where $b = 1$ denotes real pairs and $b = 0$ denotes random pairs, we first measure the unconditional probability

$$q_j^{(b)} := \Pr[T_j = \varepsilon_j \mid \text{label} = b].$$

We then compare it with the two conditional probabilities

$$q_{j,E}^{(b)} := \Pr[T_j = \varepsilon_j \mid \text{label} = b,\ E_j], \qquad q_{j,\neg E}^{(b)} := \Pr[T_j = \varepsilon_j \mid \text{label} = b,\ \neg E_j].$$

Thus, $q_{j,E}^{(b)}$ measures the bias of T_j inside the C3 branch while $q_{j,\neg E}^{(b)}$ measures the remaining bias outside that branch.

To quantify how much of the positive sample bias is explained by the C3 branch, we also define the contribution ratio

$$F_j := \frac{\Pr[E_j \mid \text{label} = 1] \cdot \left(q_{j,E}^{(1)} - \frac{1}{2}\right)}{q_j^{(1)} - \frac{1}{2}}.$$

If F_j is close to one, then most of the global bias of T_j comes from the C3-triggered subset.

Table 4: C3-triggered concentration of the variant DL biases. All probabilities are computed with $5 \cdot 10^6$ real pairs and $5 \cdot 10^6$ random pairs.

ID	event E_j	$q_j^{(1)}$	$\Pr[E_j \mid 1]$	$q_{j,E}^{(1)}$	$q_{j,\neg E}^{(1)}$	F_j	$q_{j,E}^{(0)}$
16	$\mathsf{C3}_{11}$	0.5336	0.0906	0.7620	0.5109	0.7060	0.5003
17	$\mathsf{C3}_{11}$	0.5340	0.0905	0.7620	0.5113	0.6973	0.5003
18	$\mathsf{C3}_{11}$	0.5336	0.0906	0.7620	0.5108	0.7066	0.5003
19	$\mathsf{C3}_{11}$	0.5342	0.0906	0.7620	0.5115	0.6946	0.5003
20	$\mathsf{C3}_{10}$	0.5206	0.1016	0.7005	0.5003	0.9887	0.5000

Table 4 shows a clear concentration phenomenon. For ID16–ID19, the unconditional probability $q_j^{(1)}$ is only about 0.534. However, after conditioning on $\mathsf{C3}_{11}$, the probability increases to 0.762. Outside $\mathsf{C3}_{11}$, the same probabilities drop to approximately 0.511. Therefore, the bias of these four variant DL terms is not distributed uniformly over all real pairs. Instead, a large part of the bias is concentrated on a C3-triggered subset whose probability is only about 0.091 among real pairs.

The contribution ratios F_j support the same conclusion. For ID16–ID19, roughly 70% of the global positive sample bias of these terms can be attributed to the $\mathsf{C3}_{11}$ branch. ID20 is even more sharply localized; its unconditional probability is 0.5206, while conditioning on $\mathsf{C3}_{10}$ raises it to 0.7005. Outside $\mathsf{C3}_{10}$, the probability is 0.5003, which is essentially unbiased. The corresponding contribution ratio is 0.9887, indicating that almost all of the global bias of ID20 is explained by the $\mathsf{C3}_{10}$-triggered subset.

For random pairs, the same C3 events do not create any comparable alignment:

$$q_{j,E}^{(0)} \approx 0.500$$

for all five terms. Hence, the observed effect is not caused merely by the C3 event being frequent, nor by the parity T_j being intrinsically unbalanced as a Boolean function. The C3 event rather selects a structured subset of real pairs on which the corresponding variant DL parity is strongly aligned.

Finally, we check whether the same C3-triggered alignment is reflected in the neural score itself. We divide the samples into score ranges and repeat the same conditional measurement. The result is summarized in Table 5.

Table 5: C3-triggered alignment in selected score ranges. The first four variant terms use $\mathsf{C3}_{11}$, whereas ID20 uses $\mathsf{C3}_{10}$.

ND score	$q_{\mathsf{C3}_{11}}^{(1)}$	$q_{\mathsf{C3}_{11}}^{(0)}$	$q_{\mathsf{C3}_{10}}^{(1)}$	$q_{\mathsf{C3}_{10}}^{(0)}$
$[0.9, 1]$	1.0000	1.0000	1.0000	1.0000
$[0.8, 0.9)$	0.9971	0.9709	0.9976	0.9744
$[0.1, 0.2)$	0.0167	0.0142	0.9184	0.8846

Table 5 shows that the C3-triggered alignment is more directly tied to the neural score than to the true label alone. For high scores, both real pairs and random pairs show almost

perfect alignment with the C3-triggered variant-DL parities. In the interval $[0.9, 1]$, we have

$$q^{(1)}_{\mathsf{C3}_{11}} = q^{(0)}_{\mathsf{C3}_{11}} = 1.0000, \qquad q^{(1)}_{\mathsf{C3}_{10}} = q^{(0)}_{\mathsf{C3}_{10}} = 1.0000.$$

Even false-positive random pairs are aligned with the same local-constraint Fourier features as high-score real pairs. Thus, the high neural score selects random pairs that look positive with respect to these C3-triggered Fourier features.

The low-score interval gives the complementary behavior. For ID16–ID19, the $\mathsf{C3}_{11}$-conditioned probabilities are close to zero for both labels:

$$q^{(1)}_{\mathsf{C3}_{11}}([0.1, 0.2)) = 0.0167, \qquad q^{(0)}_{\mathsf{C3}_{11}}([0.1, 0.2)) = 0.0142.$$

This indicates strong anti-alignment of the favored parities in the low-score region, independently of whether the sample is real or random. Therefore the same C3-triggered variant-DL features that characterize high-score samples appear with the opposite orientation in low-score samples.

The behavior of ID20 is different. The $\mathsf{C3}_{10}$-conditioned probabilities remain high in the low-score interval:

$$q^{(1)}_{\mathsf{C3}_{10}}([0.1, 0.2)) = 0.9184, \qquad q^{(0)}_{\mathsf{C3}_{10}}([0.1, 0.2)) = 0.8846.$$

ID20 does not follow the same score-conditioned alignment pattern as ID16–ID19. This confirms that the neural score is not determined by a single C3-triggered variant-DL term. Instead, the score reflects the combined effect of several local-constraint and difference-dependent features, among which the $\mathsf{C3}_{11}$-triggered terms ID16–ID19 and the $\mathsf{C3}_{10}$-triggered term ID20 play different roles.

4.3 Traditional DL Distinguishers and Branch Coordinates

For every traditional DL distinguisher, we have $\Gamma = \Gamma'$ which implies that its parity does not contain $\langle \eta, y \rangle$. Traditional DL terms intrinsically describe the difference-dependent component of the distinguisher, expressed in the last round variables naturally used in the local-constraint interpretation. This suggests that their mechanism should not be forced into the same C3 explanation used for the variant DL distinguishers.

Instead of analyzing the specific mechanism of the traditional DL distinguisher in terms of the local-constraint interpretation, we focus on an intriguing phenomenon that is observed by comparing $\mathcal{ND}_7^{\text{Gohr}}$ and $\mathcal{ND}_7^{\text{Yuan}}$. Table 2 and 10 show that the strongest traditional DL masks extracted from $\mathcal{ND}_7^{\text{Gohr}}$ and $\mathcal{ND}_7^{\text{Yuan}}$ have a large overlap. In particular, the four dominant mixed masks

$$\texttt{0x58204020}, \quad \texttt{0x2c102010}, \quad \texttt{0x70206020}, \quad \texttt{0x38103010}$$

appear among the highest-ranked terms for both distinguishers, although their relative order and coefficients differ. This indicates that both distinguishers capture a common difference-dependent DL feature. However, two results inspire us in two important ways: First, the pure-γ terms

$$\texttt{0x04000000}, \qquad \texttt{0x02000000}$$

appear as significant terms despite having almost no empirical bias as ordinary DL parities. Second, $\mathcal{ND}_7^{\text{Yuan}}$ gives a much larger coefficient to the pure-β term

$$\texttt{0x00080008}.$$

These observations motivate a more refined analysis of the traditional DL part.

Pure-γ terms as branches. Consider first the traditional DL mask

$$\Gamma = \Gamma' = \texttt{0x04000000}.$$

This corresponds to

$$\mu = \texttt{0x0400}, \qquad \nu = \texttt{0x0000}.$$

Therefore

$$T_{0400} = \langle \texttt{0x0400}, \gamma \rangle.$$

Similarly,

$$\texttt{0x02000000} \quad \longleftrightarrow \quad T_{0200} = \langle \texttt{0x0200}, \gamma \rangle.$$

If Fourier coefficients were simply reflecting an ordinary empirical DL probability, then such a term would be expected to have a visibly biased distribution over positive samples. However, the empirical measurements show the opposite when we test with 10^9 samples:

$$\Pr[T_{0400} = 0 \mid \text{label} = 1] = 0.5014, \qquad \Pr[T_{0200} = 0 \mid \text{label} = 1] = 0.5002.$$

Both parities are empirically unbiased over real pairs, implying that neither T_{0400} nor T_{0200} should be interpreted as a strong standalone classical DL distinguisher.

Let

$$G_g := \langle g, \gamma \rangle$$

be a single-bit pure-γ parity. For a mixed traditional DL term

$$A_{u,v} := \langle u \oplus v, \gamma \rangle \oplus \langle v \ggg 2, \beta \rangle = \langle \mu, \gamma \rangle \oplus \langle \nu, \beta \rangle,$$

the first question is whether the bit g actually appears in the γ-mask μ. We therefore first focus on the direct case

$$\mu \,\&\, g \neq 0.$$

where & denote the bitwise intersection. In this case, the mixed term $A_{u,v}$ explicitly contains the pure γ direction g. The relevant question is then whether conditioning on

$$G_g = 0 \quad \text{or} \quad G_g = 1$$

changes the bias of such mixed terms over positive samples, and whether the neural score reflects the same branch dependence.

In the selected top mixed traditional DL family considered here, we focus on the two strongest single-bit γ terms in our Fourier representation:

$$\texttt{0x0200}, \quad \texttt{0x0400}.$$

For the direct branch-bias analysis, we only use those mixed terms whose γ-mask actually contains the tested bit g. For each g, we measure

$$\rho_{A|G_g=v} = \mathbb{E}[(-1)^A \mid \text{label} = 1,\ G_g = v]$$

for the mixed terms $A = A_{u,v}$ satisfying $\mu \,\&\, g \neq 0$. For

$$G_{0200} = \langle \texttt{0x0200}, \gamma \rangle,$$

the direct mixed terms are

$$A_{1600,1000} = \langle \texttt{0x0600}, \gamma \rangle \oplus \langle 0x\texttt{0400}, \beta \rangle, \qquad A_{1e00,1800} = \langle \texttt{0x0600}, \gamma \rangle \oplus \langle \texttt{0x0600}, \beta \rangle.$$

Their branchwise biases are

mixed term	$\rho_{A\|G_{0200}=0}$	$\rho_{A\|G_{0200}=1}$	gap
$A_{1600,1000}$	$+0.1687$	$+0.0003$	0.1683
$A_{1e00,1800}$	$+0.1678$	-0.0003	0.1680

Thus, the mixed terms containing the bit `0x0200` are strongly biased in the branch $G_{0200} = 0$, but almost unbiased in the branch $G_{0200} = 1$. In this sense, conditioning on G_{0200} separates the positive samples into a branch where these traditional DL relations exhibit strong bias and a branch where the bias almost disappears. Similarly, for

$$G_{0400} = \langle \texttt{0x0400}, \gamma \rangle,$$

the branch effects are

mixed term	$\rho_{A\|G_{0400}=0}$	$\rho_{A\|G_{0400}=1}$	gap
$A_{3c10,3010}$	$+0.2142$	-0.0131	0.2273
$A_{2c10,2010}$	$+0.2144$	$+0.0131$	0.2013
$A_{1400,1000}$	$+0.1043$	$+0.0640$	0.0403
$A_{1c00,1800}$	$+0.0788$	$+0.0894$	0.0106
$A_{1600,1000}$	$+0.0848$	$+0.0843$	0.0006
$A_{1e00,1800}$	$+0.0835$	$+0.0841$	0.0006

The effect of G_{0400} is more selective than that of G_{0200}; it is very strong for the local family with $\mu =$ `0x0c00`, i.e., $A_{2c10,2010}$ and $A_{3c10,3010}$, but weak or negligible for some other terms.

We additionally test $g =$ `0x0800` which itself does not appear in the high-ranked Fourier terms but is frequently observed in mixed relations. The result shows only a moderate modulation effect. For example,

$$\rho_{A_{5820,4020}|G_{0800}=0} = +0.1601, \qquad \rho_{A_{5820,4020}|G_{0800}=1} = +0.1099.$$

Although both branches remain biased, the gap is not as strong as those of the two previous branches. Note that we further checked the remaining single-bit γ-directions. Apart from `0x0200`, `0x0400`, and the weaker `0x0800`, no other direction gives meaningful direct evidence of a branch-coordinate effect in the selected mixed DL family.

Companion Terms. We also focus on the fact that there are several similar traditional DL distinguishers in the top-20 terms where only the existence of the pure-γ term differs. We therefore examine *companion terms* to explain the paired behavior among traditional DL terms. For a fixed pure-γ parity

$$G_g = \langle g, \gamma \rangle,$$

we call two mixed terms

$$A_{u,v} = \langle \mu, \gamma \rangle \oplus \langle \nu, \beta \rangle, \qquad A_{u\oplus g,v} = \langle \mu \oplus g, \gamma \rangle \oplus \langle \nu, \beta \rangle$$

companion terms with respect to G_g. In other words, companion terms have the same β-mask ν, while their γ-masks differ exactly by the tested direction g. They satisfy the algebraic identity

$$A_{u,v} \oplus G_g = A_{u\oplus g,v}.$$

If

$$A_2 = A_1 \oplus G_g,$$

then

$$G_g = 0 \quad \Longrightarrow \quad A_2 = A_1,$$

whereas

$$G_g = 1 \quad \Longrightarrow \quad A_2 = A_1 \oplus 1.$$

Therefore, companion terms must have identical branchwise correlations in the branch $G_g = 0$ and opposite signs in the branch $G_g = 1$. This paired behavior is a natural algebraic consequence of the relation between the two terms. For G_{0400}, the term $A_{2c10,2010}$ contains the gamma bit `0x0400`, while its companion $A_{2810,2010}$ does not. They satisfy

$$A_{2c10,2010} = A_{2810,2010} \oplus G_{0400}.$$

Their branchwise correlations are

mixed term	$\rho_{A\mid G_{0400}=0}$	$\rho_{A\mid G_{0400}=1}$
$A_{2810,2010}$	$+0.2144$	-0.0131
$A_{2c10,2010}$	$+0.2144$	$+0.0131$

As predicted, the two values are identical in the branch $G_{0400} = 0$ and have opposite signs in the branch $G_{0400} = 1$. Similarly,

$$A_{3c10,3010} = A_{3810,3010} \oplus G_{0400},$$

and we observe

mixed term	$\rho_{A\mid G_{0400}=0}$	$\rho_{A\mid G_{0400}=1}$
$A_{3810,3010}$	$+0.2142$	$+0.0131$
$A_{3c10,3010}$	$+0.2142$	-0.0131

The same companion behavior appears for G_{0200}. The branchwise correlations of direct-containing terms with the companion terms are

mixed term	$\rho_{A\mid G_{0200}=0}$	$\rho_{A\mid G_{0200}=1}$
$A_{1400,1000}$	$+0.1687$	-0.0003
$A_{1600,1000}$	$+0.1687$	$+0.0003$
$A_{1c00,1800}$	$+0.1678$	$+0.0003$
$A_{1e00,1800}$	$+0.1678$	-0.0003

These companion results confirm the internal consistency of the branch analysis: once a direct-containing mixed term shows a large branch-dependent bias, its companion term exhibits the deterministic equality and sign-reversal pattern induced by G_g. Note that the direct evidence for the relevance of g remains the large branch gap of the mixed terms whose γ-mask actually contains g, and the results of the companion terms are auxiliary evidence that they are algebraically coupled through the pure-γ terms.

ND scoring by branches. The preceding experiment shows that the positive distribution contains branch-dependent biases of mixed traditional DL terms. We next check whether the neural score reflects the same branch dependence. Let $\widetilde{s}$ be the neural score normalized over random samples with respect to the raw score s:

$$\widetilde{s} = \frac{s - \mathbb{E}[s \mid \text{random samples}]}{\mathrm{std}(s \mid \text{random samples})}.$$

For a mixed term A, we compute

$$d_{A\mid G_g=v} := \mathbb{E}\big[\widetilde{s}(-1)^A \mid \text{random samples},\ G_g = v\big]\,.$$

This measures the branchwise dependence of the neural score on the parity A. Its sign indicates which parity value is favored by the score, and its magnitude quantifies how strongly the score separates the two parity classes inside the branch $G_g = v$.

We again focus on the mixed terms whose γ-mask actually contains the tested bit g. For G_{0200}, the two mixed terms give

mixed term	$d_{A\|G_{0200}=0}$	$d_{A\|G_{0200}=1}$	gap
$A_{1600,1000}$	$+0.2714$	-0.0032	0.2746
$A_{1e00,1800}$	$+0.2491$	$+0.0287$	0.2204

The score of $\mathcal{ND}_7^{\text{Gohr}}$ strongly depends on these mixed parities in the branch $G_{0200} = 0$, whereas the dependence nearly disappears in the branch $G_{0200} = 1$. For G_{0400}, the strongest direct-containing terms are

mixed term	$d_{A\|G_{0400}=0}$	$d_{A\|G_{0400}=1}$	gap
$A_{3c10,3010}$	$+0.3006$	-0.0075	0.3081
$A_{2c10,2010}$	$+0.3293$	$+0.0256$	0.3037

Again, the score strongly uses these mixed terms in the branch $G_{0400} = 0$, whereas the dependence is very weak in the branch $G_{0400} = 1$.

The meaningful empirical observation is that, among the γ bits actually appearing in the selected top mixed DL terms, `0x0200` and `0x0400` produce large branch gaps both in the positive-sample bias and the neural score, while the others are weaker and even almost inactive. This supports the explanation that the corresponding pure-γ terms can be high-ranked in the Fourier analysis even though they are nearly balanced as standalone DL distinguishers.

We further check whether the same companion structure is reflected in the neural score. If two companion terms satisfy

$$A_2 = A_1 \oplus G_g,$$

then their score-dependence statistics must satisfy the patterns:

$$G_g = 0 \quad \Longrightarrow \quad d_{A_2|G_g=0} = d_{A_1|G_g=0},$$

whereas

$$G_g = 1 \quad \Longrightarrow \quad d_{A_2|G_g=1} = -d_{A_1|G_g=1}.$$

For G_{0400}, the companion pairs give

mixed term	$d_{A\|G_{0400}=0}$	$d_{A\|G_{0400}=1}$
$A_{2810,2010}$	$+0.3293$	-0.0256
$A_{2c10,2010}$	$+0.3293$	$+0.0256$
$A_{3810,3010}$	$+0.3006$	$+0.0075$
$A_{3c10,3010}$	$+0.3006$	-0.0075

The two values of $d_{A|G_{0400}}$ are identical in the branch $G_{0400} = 0$ and have opposite signs in the branch $G_{0400} = 1$ for both mixed terms, exactly as predicted by the companion relation. Similarly, the companion pairs for G_{0200} give the same effect:

mixed term	$d_{A\|G_{0200}=0}$	$d_{A\|G_{0200}=1}$
$A_{1400,1000}$	$+0.2714$	$+0.0035$
$A_{1600,1000}$	$+0.2714$	-0.0035
$A_{1c00,1800}$	$+0.2491$	-0.0287
$A_{1e00,1800}$	$+0.2491$	$+0.0287$

The pure-β term `0x00080008`. The mask `0x00080008` exhibits a substantial rank discrepancy between the two neural distinguishers: it is ranked 5th in the Fourier spectrum of $\mathcal{ND}_7^{\text{Yuan}}$, but only 27th in that of $\mathcal{ND}_7^{\text{Gohr}}$. This suggests that the corresponding pure-β marginal feature is exposed much more directly in the difference-oriented representation than in the raw-pair representation. Thus, it requires a separate explanation from the pure-γ branch terms discussed above.

For this traditional DL mask, we have the corresponding parity

$$T = \langle \texttt{0x0002}, \beta \rangle.$$

Let us denote this parity by Z, which is a pure-β marginal parity. Recall that the pure-γ terms G_{0400} and G_{0200} are almost balanced as standalone parities over positive samples, and their relevance is revealed only after conditioning mixed terms on the branches $G_g = 0$ and $G_g = 1$. By contrast, Z already has a visible marginal bias over 10^9 positive samples:

$$\Pr[Z = 0 \mid \text{label} = 1] = 0.4564,$$

or equivalently

$$\mathbb{E}[(-1)^Z \mid \text{label} = 1] = -0.0873.$$

$Z = 1$ occurs more often than $Z = 0$ in the real distribution and it is therefore natural that a difference-oriented distinguisher exposes it as a high-ranked traditional DL feature.

For the case of $\mathcal{ND}_7^{\text{Gohr}}$, several variant DL terms have larger absolute coefficients and appear before Z in the Fourier spectrum. In particular, 10 variant terms have coefficients larger than the magnitude of `0x00080008`. This suggests that the lower rank of `0x00080008` in the $\mathcal{ND}_7^{\text{Gohr}}$ spectrum is consistent with a relative shift of Fourier mass toward value-dependent features, including the variant DL terms associated with local constraints.

5 Neural-Inspired Analysis for the Boomerang Right-Quartet Condition

Recent works have used neural distinguishers not only as attack components, but also as tools for extracting and interpreting cryptanalytic features of block ciphers [ZYS+26, YWZW26]. Motivated by this viewpoint, we ask whether a boomerang right-quartet condition leaves detectable structure in the ciphertext. The essential object tested by a boomerang distinguisher is a right-quartet condition rather than a visible relation on the original ciphertext pair. We investigate whether the right-quartet condition leaves a statistical feature on the original ciphertext pair of `SPECK32/64` by analyzing a corresponding neural distinguisher.

5.1 A Boomerang Right-Quartet Neural Distinguisher

Our experiment feeds only the original ciphertext pair

$$(C, C') = (C_L, C_R, C'_L, C'_R)$$

to the neural network and ask whether the right-quartet event can be inferred statistically from this pair alone. For

$$P' = P \oplus \Delta, \quad C = E_K^{(r)}(P), \quad C' = E_K^{(r)}(P'),$$

define the right-quartet event

$$\mathcal{R}_{K,\Delta,\nabla}^{(r)} := \left\{ (E_K^{(r)})^{-1}(C \oplus \nabla) \oplus (E_K^{(r)})^{-1}(C' \oplus \nabla) = \Delta \right\}.$$

The positive distribution $\mathcal{D}^+$ is defined as the distribution of the original ciphertext pair (C, C') generated under the input difference $P' = P \oplus \Delta$, conditioned on the right-quartet event $R^{(r)}_{K,\Delta,\nabla}$. Similarly, the negative distribution $\mathcal{D}^-$ is defined as the distribution of (C, C') generated under the same input difference, conditioned on $\neg R^{(r)}_{K,\Delta,\nabla}$. Thus, the label is determined by whether the boomerang right-quartet event occurs, whereas the neural network observes only the unmodified ciphertext pair (C, C'). To eliminate distinguishability coming from the marginal distribution of the observed ciphertext difference, we match the two classes with respect to

$$\delta_C := C \oplus C'.$$

Concretely, after generating positive and negative pools, we match the histogram of the two datasets so that the final positive and negative datasets have the same empirical multiset of δ_C. The resulting task therefore isolates information in the raw pair beyond the marginal output-difference distribution.

We instantiate this construction with the 10-round `SPECK32/64` boomerang distinguisher reported in [WWS23]. The parameters are

$$r = 10, \qquad \Delta = (\texttt{0x2800}, \texttt{0x0010}), \qquad \nabla = (\texttt{0x8102}, \texttt{0x8108}).$$

Using this setting, we generate 10^6 training samples and use the same network architecture as in $\mathcal{ND}^{\text{Gohr}}$. The network is given neither the plaintexts nor the decryption outputs (Q, Q'), but receives only the original ciphertext pair (C, C'). The resulting neural distinguisher reaches an accuracy of 66.7% on the matched classification task. The observed distinguishability indicates that conditioning on the right-quartet event induces a residual value-dependent bias in the distribution of raw ciphertext pairs.

Let us denote our 10-round boomerang right-quartet neural distinguisher $\mathcal{BRQ}_{10}$. We apply Fourier analysis to $\mathcal{BRQ}_{10}$ for a global interpretation in terms of the parities of the ciphertext pair. The resulting spectrum is sparse and dominated by specific value-dependent parities. The results are summarized in Table 6 and 7.

Table 6: Accuracy of raw $\mathcal{BRQ}_{10}$ and the corresponding result of Fourier analysis.

τ	T #Terms	ACC
0.01	29	0.626
0	-	0.667

5.2 A Local Explanation of the Dominant Terms.

We now give a local explanation for the dominant Fourier terms in Table 7. The largest coefficients can be traced to the first inverse step of the lower boomerang characteristic. In the 10-round boomerang trail used here, ∇ is expected to propagate by one inverse round as

$$\texttt{(0x8102,0x8108)} \rightarrow \texttt{(0x8000,0x8002)}.$$

Let $C = (C_L, C_R)$ be one ciphertext. We write the subtraction output before the inverse rotation as

$$x = z \boxminus y, \qquad y = (C_L \oplus C_R) \ggg 2, \qquad z = C_L \oplus k_9.$$

For $\bar{C} = C \oplus \nabla$, where $\nabla = (\texttt{0x8102}, \texttt{0x8108})$, we have

$$\bar{z} = z \oplus n, \qquad \bar{y} = y \oplus m,$$

Table 7: Top 20 Fourier terms of $\mathcal{BRQ}_{10}$ rewritten in the (β, γ, y) variables and sorted by the absolute value of the Fourier coefficient.

ID	Γ	Γ'	(β, γ, y)	Coefficient
1	0x06000600	0x00000000	$y_7 \oplus y_8$	-0.4782
2	0x00000000	0x06000600	$\beta_7 \oplus \beta_8 \oplus y_7 \oplus y_8$	-0.4455
3	0x05000500	0x00000000	$y_6 \oplus y_8$	-0.2645
4	0x00000000	0x05000500	$\beta_6 \oplus \beta_8 \oplus y_6 \oplus y_8$	-0.2568
5	0x03000300	0x06000600	$\beta_7 \oplus \beta_8 \oplus y_6 \oplus y_8$	$+0.1884$
6	0x06000600	0x03000300	$\beta_6 \oplus \beta_7 \oplus y_6 \oplus y_8$	$+0.1837$
7	0x07800700	0x04800400	$\gamma_7 \oplus \beta_8 \oplus y_6 \oplus y_7$	$+0.1260$
8	0x05000500	0x06000600	$\beta_7 \oplus \beta_8 \oplus y_6 \oplus y_7$	-0.1242
9	0x06000600	0x05000500	$\beta_6 \oplus \beta_8 \oplus y_6 \oplus y_7$	-0.1233
10	0x04800400	0x07800700	$\gamma_7 \oplus \beta_6 \oplus \beta_7 \oplus \beta_8 \oplus y_6 \oplus y_7$	$+0.1230$
11	0x00000000	0x04800480	$\beta_5 \oplus \beta_8 \oplus y_5 \oplus y_8$	-0.1106
12	0x04800480	0x00000000	$y_5 \oplus y_8$	-0.1061
13	0x03000300	0x00000000	$y_6 \oplus y_7$	$+0.1021$
14	0x00000000	0x03000300	$\beta_6 \oplus \beta_7 \oplus y_6 \oplus y_7$	$+0.0997$
15	0x03000300	0x07800780	$\beta_5 \oplus \beta_6 \oplus \beta_7 \oplus \beta_8 \oplus y_5 \oplus y_8$	-0.0769
16	0x07800780	0x03000300	$\beta_6 \oplus \beta_7 \oplus y_5 \oplus y_8$	-0.0710
17	0x06000600	0x02800280	$\beta_5 \oplus \beta_7 \oplus y_5 \oplus y_8$	$+0.0694$
18	0x04800400	0x04800400	$\gamma_7 \oplus \beta_8$	-0.0663
19	0x02800280	0x06000600	$\beta_7 \oplus \beta_8 \oplus y_5 \oplus y_8$	$+0.0571$
20	0x07800700	0x03000380	$\gamma_7 \oplus \beta_5 \oplus \beta_6 \oplus \beta_7 \oplus y_5 \oplus y_8$	$+0.0552$

with

$$n = \texttt{0x8102}, \qquad m = (\texttt{0x8102} \oplus \texttt{0x8108}) \ggg 2 = \texttt{0x8002}.$$

The lower transition requires the left-word difference 0x8000 after applying the inverse rotation $\lll 7$. Equivalently, the two subtraction outputs before this rotation must satisfy

$$x \oplus \bar{x} = \texttt{0x8000} \ggg 7 = \texttt{0x0100}.$$

Let

$$d := \texttt{0x0100}.$$

Then the local condition is

$$(z \boxminus y) \oplus ((z \oplus n) \boxminus (y \oplus m)) = d. \tag{2}$$

The important point is that

$$n \oplus m = \texttt{0x8102} \oplus \texttt{0x8002} = \texttt{0x0100} = d.$$

Hence the required output difference d is already equal to the direct XOR contribution of the masks n and m. Therefore Equation 2 is equivalent to requiring that the two borrow chains introduce no additional difference on the output bits.

Let b_i and $\bar{b}_i$ denote the borrow bits entering bit i in $z \boxminus y$ and $(z \oplus n) \boxminus (y \oplus m)$, respectively. We use the convention $b_0 = \bar{b}_0 = 0$. The bitwise outputs of the two subtractions are

$$x_i = z_i \oplus y_i \oplus b_i, \qquad \bar{x}_i = (z_i \oplus n_i) \oplus (y_i \oplus m_i) \oplus \bar{b}_i.$$

Thus

$$x_i \oplus \bar{x}_i = n_i \oplus m_i \oplus b_i \oplus \bar{b}_i.$$

Since $d_i = n_i \oplus m_i$, the condition $x \oplus \bar{x} = d$ is equivalent to

$$b_i = \bar{b}_i \qquad \text{for all output positions } 0 \le i \le 15. \tag{3}$$

In other words, the required local transition is realized exactly when the two borrow chains remain synchronized throughout all output bits.

It remains to determine when Equation 3 is preserved. Let $B(a, b, c)$ denote the one-bit borrow update function for the subtraction $a - b - c$. Then

$$b_{i+1} = B(z_i, y_i, b_i), \qquad \bar{b}_{i+1} = B(z_i \oplus n_i, y_i \oplus m_i, \bar{b}_i),$$

where

$$B(a, b, c) = ((1 \oplus a)b) \oplus ((1 \oplus a)c) \oplus bc.$$

The active bit positions of n and m are

$$(n_i, m_i) = \begin{cases} (1,1), & i = 1, 15, \\ (1,0), & i = 8, \\ (0,0), & \text{otherwise.} \end{cases}$$

At positions where $(n_i, m_i) = (0, 0)$, equality of incoming borrows directly implies equality of outgoing borrows. Hence only the active positions $i = 1, 8, 15$ can disturb Equation 3.

At bit 1, both subtraction inputs are flipped. Assuming $b_1 = \bar{b}_1$, the outgoing borrows remain equal if and only if

$$B(z_1, y_1, b_1) = B(z_1 \oplus 1, y_1 \oplus 1, b_1).$$

Using the above expression of B, we have the identity

$$B(a, b, c) \oplus B(a \oplus 1, b \oplus 1, c) = a \oplus b.$$

Therefore the equality of the outgoing borrows at bit 1 holds if and only if

$$z_1 \oplus y_1 = 0, \qquad \text{that is,} \qquad z_1 = y_1.$$

This gives the first nontrivial local compatibility condition.

Between bits 2 and 7, the masks are inactive, so the equality of borrows propagates unchanged up to bit 8. At bit 8, only the z-side is flipped:

$$n_8 = 1, \qquad m_8 = 0.$$

Assuming $b_8 = \bar{b}_8$, synchronization at the next bit requires

$$B(z_8, y_8, b_8) = B(z_8 \oplus 1, y_8, b_8).$$

The corresponding identity is

$$B(a, b, c) \oplus B(a \oplus 1, b, c) = b \oplus c.$$

Thus the outgoing borrows at bit 8 are equal if and only if

$$y_8 \oplus b_8 = 0, \qquad \text{that is,} \qquad y_8 = b_8.$$

This is the second nontrivial local compatibility condition.

Finally, bit 15 also has $(n_{15}, m_{15}) = (1, 1)$. However, this bit can only affect the final borrow b_{16}, which does not enter any output bit of the 16-bit subtraction. Since Equation 2

constrains only output bits $0, \ldots, 15$, bit 15 imposes no additional condition. Consequently, up to the unused final borrow, the local transition is governed by the two constraints

$$z_1 = y_1, \qquad y_8 = b_8.$$

The first is a low-bit compatibility condition induced by the simultaneous flip $(n_1, m_1) = (1, 1)$, whereas the second is induced by the asymmetric flip $(n_8, m_8) = (1, 0)$. The latter condition is the source of the dominant Fourier parities: b_8 is the borrow entering bit 8 in $z \boxminus y$, and therefore depends only on the lower eight bits of z and y.

Under the standard local uniform model,

$$\Pr[b_8 = y_7 \mid z_1 = y_1] = \frac{3}{4}, \qquad \Pr[b_8 = y_6 \mid z_1 = y_1] = \frac{5}{8}, \qquad \Pr[b_8 = y_5 \mid z_1 = y_1] = \frac{9}{16}.$$

The detailed derivation is provided in Appendix D. Consequently, whenever the positive right-quartet distribution is concentrated on this lower inverse transition, the condition $y_8 = b_8$ induces the biased parities

$$y_7 \oplus y_8, \qquad y_6 \oplus y_8, \qquad y_5 \oplus y_8$$

with decreasing strength. This accounts for the terms ID1, ID3, and ID12 of Table 7. Note that the same reasoning applies to the C'-branch. Since $\beta = y \oplus y'$, a parity on y' can be rewritten in the (β, y) notation as

$$y'_7 \oplus y'_8 = \beta_7 \oplus \beta_8 \oplus y_7 \oplus y_8,$$

and similarly for the lower-index variants. This accounts for the paired Fourier terms in Table 7, such as

$$\beta_7 \oplus \beta_8 \oplus y_7 \oplus y_8, \qquad \beta_6 \oplus \beta_8 \oplus y_6 \oplus y_8.$$

The label 1 distribution is defined by the full right-return event and may contain contributions from multiple compatible trails. Nevertheless, if a significant fraction of right quartets follows the dominant lower transition

$$(\texttt{0x8102}, \texttt{0x8108}) \to (\texttt{0x8000}, \texttt{0x8002}),$$

then Equation 3 provides a natural source of the value-dependent y-parities detected as the largest Fourier coefficients of $\mathcal{BRQ}_{10}$. Thus, in analogy with the C3-triggered explanation of variant DL terms, the boomerang right-quartet neural distinguisher appears to exploit a local borrow constraint of the ARX operation that is visible through the raw ciphertext pair.

Terms involving γ_7. The same local mechanism also explains why several terms in Table 7 contain the additional bit γ_7. Repeating the same expansion for the C'-branch gives

$$y'_8 = b'_8,$$

where b'_8 denotes the borrow entering bit 8 in the corresponding subtraction for C'. Since $y'_8 = y_8 \oplus \beta_8$, XORing the two branch conditions yields

$$\beta_8 = b_8 \oplus b'_8. \tag{4}$$

Thus, β_8 measures whether the bit-8 borrows in the two branches differ.

The bit b_8 is determined by the comparison of the lower eight bits in the subtraction. Hence the most influential observable difference bit for changing this borrow is the bit

immediately below the borrow boundary, namely γ_7. Consequently, the two-branch compatibility condition in Equation 4 has a natural low-order projection onto

$$\gamma_7 \oplus \beta_8.$$

This gives a natural explanation for the appearance of the term $\gamma_7 \oplus \beta_8$.

The remaining terms containing γ_7 are consistent with value-conditioned refinements of the same borrow-difference relation. For instance,

$$\gamma_7 \oplus \beta_8 \oplus y_6 \oplus y_7$$

adds the local y-context that affects the bit-8 borrow in the C-branch. Similarly,

$$\gamma_7 \oplus \beta_6 \oplus \beta_7 \oplus \beta_8 \oplus y_6 \oplus y_7 = \gamma_7 \oplus \beta_8 \oplus y'_6 \oplus y'_7$$

is the analogous refinement expressed through the C'-branch value bits. Therefore these terms should not be interpreted as independent difference-only distinguishers. They are better understood as low-order Fourier projections of the two-branch borrow compatibility condition, with the additional y- or y'-bits making the value conditioning explicit.

6 Conclusion

We connected the local-constraint interpretation and Fourier analysis of neural distinguishers for `SPECK32/64`. For the 7-round raw-pair differential neural distinguisher, the dominant Fourier parities can be rewritten in the local variables (β, γ, y), separating y-dependent variant DL terms from difference-dependent traditional DL terms. The dominant variant terms are concentrated on C3-triggered branches, while the traditional part contains mixed (γ, β)-relations, pure-γ branch effects and pure-β marginal features.

We also applied this viewpoint to a boomerang right-quartet neural distinguisher. After matching the classes by the observed ciphertext difference, the remaining distinguishability is explained by value-dependent structure in the raw pair. The dominant y-parities of $\mathcal{BRQ}_{10}$ are traced to the first lower inverse transition, whose borrow synchronization yields specific one-bit constraints. Thus the extracted Fourier terms can be interpreted as observable projections of concrete local ARX constraints. This interpretation is not a one-to-one correspondence for every term, but a common framework in which Fourier features and local ARX constraints can be compared.

References

[BGPT21] Adrien Benamira, David Gerault, Thomas Peyrin, and Quan Quan Tan. A deeper look at machine learning-based cryptanalysis. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 805–835. Springer, 2021.

[BLN17] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-linear cryptanalysis revisited. *Journal of Cryptology*, 30(3):859–888, 2017.

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

[BODKW19] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. Dlct: a new tool for differential-linear cryptanalysis. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 313–342. Springer, 2019.

[BTCS+15] Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. The simon and speck lightweight block ciphers. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2015. doi:10.1145/2744769.2747946.

[BV14] Alex Biryukov and Vesselin Velichkov. Automatic search for differential trails in arx ciphers. In *Cryptographers' track at the RSA conference*, pages 227–250. Springer, 2014.

[CHP+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 683–714. Springer, 2018.

[CY21] Yi Chen and Hongbo Yu. Bridging machine learning and cryptanalysis via edlct. *Cryptology ePrint Archive*, 2021.

[Din14] Itai Dinur. Improved differential cryptanalysis of round-reduced speck. In *International Conference on Selected Areas in Cryptography*, pages 147–164. Springer, 2014.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time attack on the a5/3 cryptosystem used in third generation gsm telephony. *Cryptology ePrint Archive*, 2010.

[GL89] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989.

[Goh19] Aron Gohr. Improving attacks on round-reduced Speck32/64 using deep learning. pages 150–179, 2019. doi:10.1007/978-3-030-26951-7_6.

[LH94] Susan K Langford and Martin E Hellman. Differential-linear cryptanalysis. In *Annual International Cryptology Conference*, pages 17–25. Springer, 1994.

[NSLL22] Zhongfeng Niu, Siwei Sun, Yunwen Liu, and Chao Li. Rotational differential-linear distinguishers of ARX ciphers with arbitrary output linear masks. pages 3–32, 2022. doi:10.1007/978-3-031-15802-5_1.

[O'D21] Ryan O'Donnell. Analysis of boolean functions, 2021. URL: https://arxiv.org/abs/2105.10386, arXiv:2105.10386.

[Wag99] David Wagner. The boomerang attack. In *International Workshop on Fast Software Encryption*, pages 156–170. Springer, 1999.

[WWS23] Dachao Wang, Baocang Wang, and Siwei Sun. Sat-aided automatic search of boomerang distinguishers for arx ciphers. *IACR Transactions on Symmetric Cryptology*, pages 152–191, 2023.

[YWZW26] Yufei Yuan, Wenling Wu, Lei Zhang, and Ruichen Wu. Fourier analysis of neural distinguishers. *IACR Transactions on Symmetric Cryptology*, 2026(1):441–467, 2026.

[ZYS+26] Liu Zhang, Yiran Yao, Danping Shi, Dongchen Chai, Jian Guo, and Zilong Wang. Neural-inspired advances in integral cryptanalysis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 451–481. Springer, 2026.

A Local Constraints of Modular Addition of SPECK32/64

We recall only the C3 case of the one-bit local-constraint interpretation in [BLYZ23] since this is the case used in our analysis. The complete list of one-bit constraints, carry-deduction rules, and C1–C4 cases can be found in [BLYZ23].

Consider an XOR-differential

$$\delta = (\alpha, \beta \mapsto \gamma)$$

through addition modulo 2^n. The one-bit condition $\mathsf{Cyc}_{i+1,i}$ is the case

$$\mathsf{eq}(\alpha, \beta, \gamma)_i = 0, \qquad \alpha_i \oplus \beta_i = 1, \qquad \alpha_i \oplus \mathrm{xor}(\alpha, \beta, \gamma)_i = 1.$$

In this case, the validity of the one-bit modular-addition transition from bit i to bit $i+1$ is checked by

$$\mathrm{xor}(\alpha, \beta, \gamma)_{i+1} \oplus \beta_i = y_i \oplus c_i.$$

Under $\mathsf{Cyc}_{i+1,i}$, the unknown carry c_i is the only obstacle to checking the condition from the known value y.

The C3 case occurs when this carry c_i can be deduced from the previous bit as

$$c_i = y_{i-1}.$$

Bao *et al*. identify three lower-bit cases that imply this deduction and they are summarized in Table 8.

Table 8: Lower-bit cases used in the $\mathsf{C3}_i$ condition.

Case	Additional condition	Carry deduction
$\mathsf{Cxy0}_{i,i-1}$	$\mathrm{xor}(\alpha, \beta, \gamma)_i \oplus \alpha_{i-1} = 0$	$c_i = y_{i-1}$
$\mathsf{Cxc1}_{i,i-1}$	$\mathrm{xor}(\alpha, \beta, \gamma)_i \oplus \alpha_{i-1} = 1$	$c_i = y_{i-1}$
$\mathsf{Cyc0}_{i,i-1}$	$\mathrm{xor}(\alpha, \beta, \gamma)_i \oplus \beta_{i-1} = 0$	$c_i = y_{i-1}$

The $\mathsf{C3}_i$ event is defined by

$$\mathsf{C3}_i := \mathsf{Cyc}_{i+1,i} \wedge \left(\mathsf{Cxy0}_{i,i-1} \vee \mathsf{Cxc1}_{i,i-1} \vee \mathsf{Cyc0}_{i,i-1}\right).$$

Whenever $\mathsf{C3}_i$ holds, substituting $c_i = y_{i-1}$ into the $\mathsf{Cyc}_{i+1,i}$ relation gives

$$\mathrm{xor}(\alpha, \beta, \gamma)_{i+1} \oplus \beta_i = y_i \oplus y_{i-1}. \tag{5}$$

Under $\mathsf{C3}_i$, Equation 5 can be written as

$$\beta_{i+1} \oplus \gamma_{i+1} \oplus \beta_i \oplus y_i \oplus y_{i-1} = \alpha_{i+1}.$$

Therefore, if α_{i+1} is biased in positive samples generated from the fixed input difference, then the $\mathsf{C3}_i$-triggered branch transfers this latent bias to the observable parity

$$\beta_{i+1} \oplus \gamma_{i+1} \oplus \beta_i \oplus y_i \oplus y_{i-1}.$$

B Fourier Analysis of the Difference-Oriented Neural Distinguisher

We report the result of Fourier analysis of the difference-oriented neural distinguisher $\mathcal{ND}_7^{\mathrm{Yuan}}$. This distinguisher takes the ciphertext difference $C \oplus C'$ as input. Therefore, its extracted Fourier terms are difference-dependent traditional DL terms. The results are included to compare the dominant traditional masks with those of $\mathcal{ND}_7^{\mathrm{Gohr}}$ and to highlight the rank discrepancy of the pure-β term `0x00080008`.

Table 9: Accuracy of raw $\mathcal{ND}_7^{\text{Yuan}}$ and the corresponding result of Fourier analysis.

τ	#Terms	ACC
0.01	83	0.577
0	-	0.584

Table 10: Top 20 Fourier terms of $\mathcal{ND}_7^{\text{Yuan}}$ sorted by the absolute value of the Fourier coefficient.

ID	Γ	Γ'	Coefficient
1	0x38103010	0x38103010	-0.2398
2	0x2c102010	0x2c102010	-0.2298
3	0x70206020	0x70206020	-0.2105
4	0x58204020	0x58204020	-0.2054
5	0x00080008	0x00080008	$+0.2008$
6	0x3c103010	0x3c103010	-0.1674
7	0x50304030	0x50304030	-0.1576
8	0x78306030	0x78306030	-0.1552
9	0x1e001800	0x1e001800	-0.1526
10	0x1c001800	0x1c001800	-0.1515
11	0x102c1020	0x102c1020	-0.1482
12	0x10381030	0x10381030	-0.1455
13	0x16001000	0x16001000	-0.1437
14	0x28102010	0x28102010	-0.1336
15	0x14001000	0x14001000	-0.1028
16	0x02000000	0x02000000	$+0.0800$
17	0x04000000	0x04000000	$+0.0791$
18	0x08000800	0x08000800	$+0.0778$
19	0x14081008	0x14081008	-0.0757
20	0x0a080808	0x0a080808	-0.0731

C Decomposition of the C3 Trigger

We further decompose the C3 event to identify where the conditional alignment is concentrated. Write

$$A_i := \mathsf{Cyc}_{i+1,i},$$

and

$$B_i := \mathsf{Cxy0}_{i,i-1} \vee \mathsf{Cxc1}_{i,i-1} \vee \mathsf{Cyc0}_{i,i-1}.$$

Then the condition for $\mathsf{C3}_i$ becomes $A_i \wedge B_i$. We measure the conditional bias of T_j on the four regions

$$A_i \wedge B_i, \quad A_i \wedge \neg B_i, \quad \neg A_i \wedge B_i, \quad \neg A_i \wedge \neg B_i.$$

The results are shown in Table 11.

Table 11: Decomposition of the C3 trigger on real pairs.

ID	C3 index	q_{A_i}	q_{B_i}	$q_{A_i \wedge B_i}$	$q_{A_i \wedge \neg B_i}$	$q_{\neg A_i \wedge B_i}$	$q_{\neg A_i \wedge \neg B_i}$
16	11	0.6351	0.5653	0.7620	0.5631	0.5000	0.4998
17	11	0.6351	0.5654	0.7620	0.5631	0.4999	0.5005
18	11	0.6351	0.5651	0.7620	0.5631	0.4995	0.4999
19	11	0.6351	0.5661	0.7620	0.5631	0.5009	0.5003
20	10	0.5917	0.5541	0.7005	0.5024	0.4996	0.5001

For ID16–ID19, the Cyc component already produces a visible partial alignment:

$$q_{A_{11}} = 0.6351.$$

When B_{11} is also imposed, the probability increases to

$$q_{A_{11} \wedge B_{11}} = 0.7620.$$

On the other hand, B_{11} alone is not informative outside the Cyc branch:

$$q_{\neg A_{11} \wedge B_{11}} \approx 0.5.$$

Therefore, B_{11} should be interpreted as a selector that becomes effective only inside the $\mathsf{Cyc}_{12,11}$ branch for ID16–ID19. This explains why the full C3 condition gives a stronger alignment than conditioning on A_{11} alone, and why B_{11} is informative only inside the $\mathsf{Cyc}_{12,11}$ branch.

For ID20, the mechanism is even more conjunctive. Although the marginal quantities $q_{A_{10}}$ and $q_{B_{10}}$ are above 0.5, this is due to their overlap on $A_{10} \wedge B_{10}$. Once the intersection is removed, neither factor is informative:

$$q_{A_{10} \wedge \neg B_{10}} = 0.5024, \qquad q_{\neg A_{10} \wedge B_{10}} = 0.4996.$$

In contrast, the full C3 condition gives

$$q_{A_{10} \wedge B_{10}} = 0.7005.$$

Thus, the observed alignment is concentrated on the simultaneous occurrence of the Cyc component and the lower selector for ID20.

D Probabilities of the Local Uniform Model

We derive the agreement probabilities between the borrow b_8 and the upper bits of the low byte of y that are used in Section 5.2. Recall that b_8 is the borrow entering bit 8 in the subtraction $z \boxminus y$. Therefore b_8 is completely determined by the lower eight bits of z and y. Let

$$Z := z_{[0,7]}, \qquad Y := y_{[0,7]},$$

where both are interpreted as integers in $\{0, \ldots, 255\}$. Under the local uniform model, the pair (Z, Y) is treated as uniformly distributed over all low-byte values satisfying the already derived compatibility condition

$$z_1 = y_1.$$

We denote this event by

$$\Omega := \{(Z, Y) : Z_1 = Y_1\}.$$

Thus $|\Omega| = 2^{15}$, since for each Y there are exactly 128 possible values of Z with the same bit 1. The borrow b_8 is the borrow-out of the low eight-bit subtraction $Z - Y$. Hence

$$b_8 = 1 \quad \Longleftrightarrow \quad Z < Y.$$

Equivalently, for a fixed value $Y = t$, $b_8 = 1$ if and only if Z lies below t. Because of the conditioning $Z_1 = Y_1 = t_1$, the relevant number of admissible values is

$$N(t) := \#\{Z \in \{0, \ldots, 255\} : Z < t,\ Z_1 = t_1\}.$$

Then

$$\Pr[b_8 = 1 \mid Y = t, \Omega] = \frac{N(t)}{128}.$$

We now compute $N(t)$. Write

$$t = 4q + r, \qquad 0 \le q \le 63, \qquad r \in \{0, 1, 2, 3\}.$$

The bit t_1 is determined by r: it is 0 for $r = 0, 1$, and 1 for $r = 2, 3$. Among every complete block

$$\{4u, 4u + 1, 4u + 2, 4u + 3\},$$

there are two values with bit 1 equal to 0, and two values with bit 1 equal to 1. A direct count gives

$$N(4q) = 2q, \qquad N(4q + 1) = 2q + 1,$$

and

$$N(4q + 2) = 2q, \qquad N(4q + 3) = 2q + 1.$$

Equivalently,

$$N(4q + r) = 2q + (r \bmod 2).$$

We are interested in

$$\Pr[b_8 = y_j \mid \Omega]$$

for $j \in \{5, 6, 7\}$. The same calculation also applies to $j = 4$. Since $j \ge 2$, the bit y_j of $Y = t = 4q + r$ is exactly the bit q_{j-2} of q; the lower residue r does not affect it. Thus the condition $y_j = e$ is equivalent to

$$q_{j-2} = e.$$

First consider the case $y_j = 1$. Averaging over all $Y = t$ with $y_j = 1$, we obtain

$$\Pr[b_8 = 1 \mid y_j = 1, \Omega] = \frac{1}{128} \sum_{\substack{0 \le t < 256 \\ t_j = 1}} \frac{N(t)}{128}.$$

Using $t = 4q + r$, this becomes

$$\Pr[b_8 = 1 \mid y_j = 1, \Omega] = \frac{1}{2^{14}} \sum_{\substack{0 \le q < 64 \\ q_{j-2} = 1}} \sum_{r=0}^{3} \big(2q + (r \bmod 2)\big).$$

Since

$$\sum_{r=0}^{3} \big(2q + (r \bmod 2)\big) = 8q + 2,$$

and

$$\sum_{\substack{0 \le q < 64 \\ q_{j-2} = 1}} q = 1008 + 16 \cdot 2^{j-2},$$

we get

$$\Pr[b_8 = 1 \mid y_j = 1, \Omega] = \frac{8(1008 + 16 \cdot 2^{j-2}) + 64}{2^{14}} = \frac{127}{256} + 2^{j-9}.$$

Similarly, for $y_j = 0$,

$$\sum_{\substack{0 \le q < 64 \\ q_{j-2} = 0}} q = 1008 - 16 \cdot 2^{j-2},$$

and hence

$$\Pr[b_8 = 1 \mid y_j = 0, \Omega] = \frac{127}{256} - 2^{j-9}.$$

Therefore

$$\Pr[b_8 = 0 \mid y_j = 0, \Omega] = 1 - \Pr[b_8 = 1 \mid y_j = 0, \Omega] = \frac{129}{256} + 2^{j-9}.$$

Combining the two cases gives

$$\begin{aligned}\Pr[b_8 = y_j \mid \Omega] &= \frac{1}{2}\Pr[b_8 = 1 \mid y_j = 1, \Omega] + \frac{1}{2}\Pr[b_8 = 0 \mid y_j = 0, \Omega] \\ &= \frac{1}{2}\left(\frac{127}{256} + 2^{j-9}\right) + \frac{1}{2}\left(\frac{129}{256} + 2^{j-9}\right) \\ &= \frac{1}{2} + 2^{j-9}.\end{aligned}$$

Thus,

$$\Pr[b_8 = y_7 \mid z_1 = y_1] = \frac{1}{2} + 2^{-2} = \frac{3}{4},$$

$$\Pr[b_8 = y_6 \mid z_1 = y_1] = \frac{1}{2} + 2^{-3} = \frac{5}{8},$$

and

$$\Pr[b_8 = y_5 \mid z_1 = y_1] = \frac{1}{2} + 2^{-4} = \frac{9}{16}.$$

For completeness, the next value is

$$\Pr[b_8 = y_4 \mid z_1 = y_1] = \frac{1}{2} + 2^{-5} = \frac{17}{32}.$$