Improved Neural Distinguisher for PRESENT-80 using Inception and Efficient Channel Attention in Related-Key Multi-Pair Setting

Thanh-Phong Nguyen*†‖, Nguyen Tan Cam*†, Van-Than Huynh‡, Tan Nguyen§, Hieu-Minh Nguyen¶,
*University of Information Technology, Ho Chi Minh City, Vietnam
†Vietnam National University, Ho Chi Minh City, Vietnam
‡Owentis Vietnam, Ho Chi Minh City, Vietnam
§Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam
¶Academy of Cryptography Techniques, Hanoi, Vietnam
‖Corresponding author: ntphong2702@gmail.com

***Abstract*—This study presents a neural distinguisher for the lightweight block cipher PRESENT-80, targeting the related-key differential setting with multiple ciphertext pairs per sample. The proposed model combines an Inception-style multi-branch convolutional block, residual connections, and Efficient Channel Attention to extract richer differential features. Each input sample includes several ciphertext pairs encrypted under fixed differences in both plaintext and key, allowing the model to learn from structured ciphertext correlations across pairs. Experimental results show that our approach outperforms prior baselines from round 7 to 9 and retains distinguishing ability up to round 15, highlighting its robustness under deeper diffusion. This demonstrates the effectiveness of combining structured multi-pair inputs with lightweight attention mechanisms for data-constrained neural cryptanalysis.**

***Index Terms*—Neural differential cryptanalysis, Inception network, attention mechanism, related-key cryptanalysis, PRESENT cipher.**

I. Introduction

Differential cryptanalysis is one of the most powerful tools in evaluating the security of symmetric-key block ciphers. In recent years, neural differential cryptanalysis (NDC) has emerged as a promising approach that leverages deep learning to build distinguishers between real cipher outputs and random permutations [1].

The core idea of NDC is to train a neural network to identify subtle statistical patterns left by differential characteristics in ciphertexts. These distinguishers, though not always directly usable for key recovery, provide valuable insights into cipher structure, round security, and cryptanalytic resistance.

Despite recent advances, most neural distinguishers to date are limited to the single-ciphertext pair setting, where the model is trained to classify one pair of ciphertexts at a time. This setting, popularized by Gohr's ResNet-based approach [1], has shown impressive results in low-round distinguishability for lightweight ciphers such as Speck or PRESENT. However, in higher-round scenarios (e.g., $r \geq 8$), the information contained in a single ciphertext pair often becomes too sparse or noisy for effective learning.

Furthermore, most prior works assume a single-key setting, ignoring the potential of related-key differential patterns that emerge when the key difference $\Delta K \neq 0$ is exploited. Related-key attacks (RKA) [2] were originally proposed in classical differential cryptanalysis, where adversaries are allowed to analyze cipher outputs under multiple, carefully related keys. This model has been adapted in recent works [3]–[5] to the neural setting, resulting in related-key neural distinguishers (RKND) that improve distinguishability by leveraging both input and key differences.

Recent works [5], [6] also show that using multiple ciphertext pairs as input can enhance feature extraction, especially when combined with related-key settings. However, integrating these techniques into lightweight and generalizable architectures remains underexplored.

In parallel, attention mechanisms have demonstrated potential in neural cryptanalysis [7], particularly in the single-pair setting. However, lightweight attention modules such as Efficient Channel Attention (ECA) [8] remain underexplored in cryptanalytic applications. We hypothesize that integrating such attention mechanisms with structured convolutional backbones, such as Inception [9], can yield compact yet expressive models capable of capturing subtle differential features across ciphertext dimensions.

Our contributions are summarized as follows:

- We introduce a multi-pair input representation ($k \geq 2$) with fixed plaintext and key differences, allowing the model to extract more informative features that reflect underlying differential patterns in the related-key setting.
- We propose a lightweight neural architecture that integrates Inception-style convolutions with Efficient Channel Attention, enabling expressive yet compact feature extraction across ciphertext dimensions.
- We demonstrate that our model outperforms prior related-key neural distinguishers on PRESENT-80 in both accuracy and round depth, achieving robust distinguishability up to 15 rounds.

The remainder of this paper is organized as follows: Section II reviews prior works on neural and related-key distinguishers. Section III introduces our input representation and model architecture. Section IV presents experimental results and analysis. Finally, Section V concludes the paper and outlines future directions.

II. Related Works

A. Brief Description of PRESENT-80 Cipher

PRESENT is a standardized ultra-lightweight block cipher proposed at CHES 2007 [10], designed for hardware-constrained environments such as RFID tags and IoT devices. It operates on a 64-bit plaintext block and supports key sizes of 80 or 128 bits. This work focuses on the 80-bit variant, commonly referred to as PRESENT-80.

The cipher adopts a Substitution–Permutation Network (SPN) structure and consists of 31 encryption rounds. Each round includes three sequential steps:

- **addRoundKey:** XOR the 64-bit state with the 64-bit round key.
- **sBoxLayer:** Apply sixteen parallel 4-bit S-boxes to each nibble of the state.
- **Permutation Layer (pLayer):** Perform a fixed bitwise permutation for diffusion across bits.

Round keys are derived from the 80-bit master key through left rotation by 61 bits, S-box substitution on the leftmost nibble, and round counter XOR into bits `K[19:15]`. The leftmost 64 bits are used as the round key for each round.

A high-level pseudocode of the encryption process is provided in Algorithm 1 [10].

Algorithm 1 PRESENT Algorithm

GenerateRoundKey()
$i = 1$
while $1 \leq i \leq 31$ **do**
  addRoundKey(STATE, K_i)
  sBoxLayer(STATE)
  pLayer(STATE)
  $i = i + 1$
end while
addRoundKey(STATE, K_{32})

The simplicity of PRESENT's design, especially its deterministic bit permutation and linear key schedule, makes it well-suited for differential and related-key cryptanalysis. These characteristics are particularly advantageous for neural-based distinguishers operating in constrained or structured settings.

B. Differential and Related-Key Notations

Differential cryptanalysis is a fundamental technique introduced by Biham and Shamir [11], which studies how differences in the inputs of a block cipher propagate through encryption rounds to affect the output differences. The key idea is to track the evolution of structured differences, or trails, through the cipher's internal transformations.

Given two plaintexts P and P^* encrypted under the same key K, we define the plaintext difference as:

$$\Delta P = P \oplus P^*$$

and the corresponding ciphertext difference as:

$$\Delta C = C \oplus C^*, \quad \text{where } C = E_K(P),\ C^* = E_K(P^*)$$

These quantities represent the input and output differences of a ciphertext pair.

A differential trail, also known as a characteristic, is a sequence of internal differences across rounds:

$$\Omega = (\beta_0, \beta_1, \ldots, \beta_r)$$

where $\beta_0 = \Delta P$ and $\beta_r = \Delta C$. Each β_i reflects how the difference evolves through round i of the cipher. While classical attacks attempt to identify high-probability characteristics to build distinguishers [11], neural-based approaches [1] aim to learn such differential patterns directly from data.

In the related-key setting [2], inputs are encrypted under two keys K and $K^* = K \oplus \Delta K$. This allows the adversary to study the combined propagation of differences through both the cipher and the key schedule. The resulting output difference becomes:

$$\Delta C = E_K(P) \oplus E_{K^*}(P^*)$$

Recent works [5] show that fixing $(\Delta P, \Delta K)$ allows structured patterns to emerge in the ciphertext space, even when the underlying differential characteristics are unknown.

In this work, we follow this approach and generate data by encrypting multiple plaintext pairs (P, P^*) with a fixed plaintext difference ΔP and key difference ΔK. The neural model is then trained to detect distinguishable structures in ΔC across many ciphertext pairs, without explicitly computing differential probabilities or trails.

C. Neural Distinguishers with Single Ciphertext Pair

The use of deep learning for differential cryptanalysis was first introduced by Gohr [1], who proposed a ResNet-based neural distinguisher for reduced-round Speck32/64. In this setting, each input sample consists of a single ciphertext pair, along with its XOR difference, and the network is trained to distinguish between real encryptions and random permutations. This work demonstrated that neural networks can effectively learn statistical biases induced by differential trails, even in the absence of explicit analytical models.

Following Gohr's framework, AutoND [12] further automated the training pipeline by introducing a cipher-agnostic infrastructure for generating data, finding good input differences, and evaluating models across various cipher families. While these contributions significantly improved the usability and reproducibility of NDC experiments, they still operate under the single-pair, single-key assumption.

However, this setup has limitations when applied to high-round distinguishers. As the number of rounds increases, the differential signal in a single ciphertext pair becomes weaker and more difficult to learn, often leading to models that fail to

generalize beyond low rounds. This motivates the exploration of alternative data representations and richer input structures, as we pursue in this work.

D. Multiple-Ciphertext Pair Approaches

To address the limitations of single-pair neural distinguishers in high-round settings, Chen et al. [6] introduced the ND_k framework, where each input sample is constructed from k ciphertext pairs instead of just one. By aggregating multiple differential observations, ND_k allows the neural network to capture more robust statistical features and mitigate the noise present in higher rounds. Their model employs derived features such as bitwise XOR and modular differences between pairs, achieving improved accuracy on reduced-round variants of the Speck cipher.

The ND_k paradigm highlights the benefit of structured input representations in neural differential cryptanalysis. Rather than relying solely on deeper models or larger datasets, it leverages data-level redundancy to amplify the distinguishing signal. This approach has since inspired various extensions, including multi-pair attention models and related-key variants.

Our work builds upon the ND_k formulation by combining it with lightweight attention and convolutional mechanisms, aiming to enhance differential feature extraction in a compact and efficient way.

E. Related-Key Attacks in Differential Cryptanalysis

Related-key attacks (RKA) were first introduced by Biham [2] as an extension of differential and linear cryptanalysis. In this model, the adversary is allowed to query encryption oracles under multiple secret keys that are related by a known relation—such as fixed XOR differences or affine transformations. While this setting may appear artificial in idealized environments, it becomes increasingly relevant in practice, especially in lightweight cryptographic systems with simplified key schedules, key reuse, or shared secrets.

The fundamental intuition behind RKA is that injecting a controlled difference into both the plaintext and the key can amplify statistical biases in the ciphertexts. This combined differential propagation through both the cipher rounds and the key schedule can expose distinguishable patterns that would otherwise remain hidden in the single-key model.

Although classical RKA techniques do not rely on machine learning, they form the theoretical basis for modern adaptations such as related-key neural distinguishers, where deep learning models are trained to recognize such differential patterns in ciphertext data generated under related keys. These modern approaches will be discussed in the next section.

F. Related-Key Neural Distinguishers

Related-key neural distinguishers (RKND) represent a recent line of research that combines related-key differential cryptanalysis with deep learning. Rather than relying on handcrafted statistical analysis, these models aim to automatically learn distinguishing features from ciphertexts generated under related keys—where the key difference is fixed and known.

The first systematic exploration of RKND was carried out by Lu et al. [3], who applied residual neural networks to train distinguishers for the SIMON and SIMECK block ciphers. Their work demonstrated that using related-key data not only improved classification accuracy but also enabled distinguishers to generalize to deeper rounds—up to 21 rounds for Simeck64/128—beyond what single-key models could achieve.

More recently, Pooja et al. [4] applied a similar concept to the PRESENT cipher. Using a simple multilayer perceptron (MLP) architecture, they trained RKND models on fixed key differences and achieved up to 5-round distinguishability, showcasing that even lightweight models can benefit from the related-key setting.

Expanding this idea, Su et al. [5] proposed a general RKND framework that supports multiple ciphertext pairs per sample. Their method focuses on DES and PRESENT, and demonstrates consistent improvements in both accuracy and round coverage using enhanced data formats and a Deep Residual Shrinkage Network (DRSN) architecture, which incorporates adaptive thresholding to suppress noise and highlight important features.

These works collectively indicate that related-key settings provide a promising direction for improving the performance of neural differential distinguishers, particularly when combined with richer input representations such as multiple ciphertext pairs.

G. Attention Mechanisms in Cryptanalysis

The use of attention mechanisms in NDC remains relatively recent. Deng et al. [7] were the first to introduce attention in this context, proposing a Vision-Transformer-based (ViT) architecture for building distinguishers on lightweight ciphers such as SPECK. Their model treats binary ciphertexts as input sequences and applies multi-head self-attention across bit positions. While this approach opened the door for attention in cryptanalysis, it was designed solely for single-pair inputs and carries the complexity of transformer-style designs.

In a more cipher-specific direction, Guo et al. [13] introduced GA-CAM, an attention-enhanced neural distinguisher targeting PRESENT and SKINNY. Their model integrates channel and spatial attention modules, enabling it to focus on salient differential features. GA-CAM demonstrated improved accuracy and offered visual interpretability through attention maps. However, it was primarily evaluated in fixed-key scenarios and not within the multi-pair or related-key settings.

Beyond these efforts, lightweight attention modules from the computer vision domain have also gained traction. Among them, the Efficient Channel Attention (ECA) module [8] offers a particularly attractive trade-off between performance and complexity. Unlike traditional channel attention mechanisms that rely on fully connected layers, ECA uses 1D convolution to model local cross-channel interactions with minimal overhead. This makes it well-suited for integration into compact neural distinguishers, particularly when handling multi-pair binary inputs.

Despite these developments, attention mechanisms remain underexplored in the context of multi-pair and related-key neural distinguishers. To the best of our knowledge, no prior work has studied the integration of lightweight attention into ND_k architectures. Our work fills this gap by embedding ECA into an Inception-based ND_k model, designed specifically for PRESENT under related-key conditions.

H. Summary of Related Works

This section reviews prior works in neural differential cryptanalysis. Table I compares key features, including related-key (RK) support, multiple ciphertext pairs (ND_k), attention mechanisms, rounds, and architectures. Our approach integrates multi-pair inputs, RK settings, and ECA for enhanced distinguishability on PRESENT.

TABLE I: Comparison of Prior Works in Neural Differential Cryptanalysis

Work	RK	ND_k	Attention	Rounds	Architecture
Gohr [1]	✗	✗	✗	7	ResNet
Chen et al. [6]	✗	✓	✗	8	DbitNet
Pooja et al. [4]	✓	✗	✗	6	MLP
Su et al. [5]	✓	✓	✗	9	ResNet
Guo et al. [13]	✗	✗	GA-CAM	8	CNN + Att.
Ours	✓	✓	ECA	15	Inception + ECA

III. Methodology

A. Input Representation

Each training sample is constructed using multiple ciphertext pairs, generated under a fixed input and key difference $(\Delta P, \Delta K)$, where $\Delta K \neq 0$. We adopt the ND_k input format $[\Delta C \parallel C \parallel C^*]$ per ciphertext pair, as proposed by Su et al. [5], where $\Delta C = C \oplus C^*$.

This triplet format enables the model to extract differential features across each pair, capturing both ciphertext-level variations and pairwise correlations. For each sample, k such triplets are concatenated, resulting in a 3D tensor of shape $(k, 3, w)$, where w is the block size (e.g., $w = 64$ for PRESENT). This is flattened into a vector of length $3kw$ before being passed into the model.

Positive samples (label 1) are generated by encrypting plaintexts under related keys and input differences, while negative samples (label 0) use independently sampled plaintexts and keys without fixed differences. Algorithm 2 describes the full process.

Algorithm 2 Dataset Generation for Related-Key ND_k

Input: Number of samples n, number of pairs k, round number r, input difference ΔP, key difference ΔK.
Output: Dataset (X, Y) of shape $(n, 3kw)$.
for $i = 1$ to n **do**
  Randomly assign label $y \in \{0, 1\}$
  Initialize sample $S \leftarrow [\,]$
  for $j = 1$ to k **do**
    Sample random plaintext P and key K
    if $y = 1$ **then**
      $P^* \leftarrow P \oplus \Delta P$
      $K^* \leftarrow K \oplus \Delta K$
    else
      Sample random P^* and K^* independently
    end if
    $C \leftarrow E_K(P, r)$
    $C^* \leftarrow E_{K^*}(P^*, r)$
    $\Delta C \leftarrow C \oplus C^*$
    Append $[\Delta C \parallel C \parallel C^*]$ to S
  end for
  $X[i] \leftarrow \text{Flatten}(S)$
  $Y[i] \leftarrow y$
end for
return (X, Y)

B. Design Motivation and Model Choice

Designing an effective neural distinguisher in the related-key, multiple-pair setting presents several challenges. Unlike traditional single-pair distinguishers, our model must process a richer input representation that captures differential structures across several ciphertext pairs, each generated under a fixed key and plaintext difference. This calls for a model architecture that can handle multiple channels, detect local and cross-pair correlations, and remain efficient enough for practical use on lightweight ciphers such as PRESENT.

To address similar concerns in computer vision balancing local detail extraction and computational efficiency, Szegedy et al. [14] introduced the Inception architecture, which became foundational in large-scale image classification (e.g., ImageNet). Inception applies multiple convolutional filters of varying sizes in parallel, allowing the model to capture patterns at different spatial resolutions simultaneously.

Inspired by this idea, Zhang et al. [9] adapted Inception-style blocks to the domain of neural differential cryptanalysis. Their work demonstrated that such multi-branch convolutional modules can improve the ability of neural distinguishers to extract differential features across various dimensions of ciphertext inputs.

Following this line, we adopt an Inception-style block as a core component of our model. When applied to multiple ciphertext pairs, this structure enables the network to process local differential patterns (bitwise differences) and more global structural biases that may arise under related-key encryption. The varying kernel sizes mimic the need to extract multi-scale interactions among ciphertext components distributed across different positions.

In parallel, we incorporate the ECA module [8] to enhance feature discrimination. ECA captures local cross-channel dependencies without introducing heavy parameter overhead. By applying adaptive 1D convolution on globally pooled statistics, ECA allows the model to focus on the most informative channels—particularly beneficial when differences induced by the cipher's key schedule or S-box diffusion manifest subtly in specific positions.

Overall, our architectural choices reflect a balance between

expressiveness and efficiency. The combination of Inception-style multi-scale feature extraction and lightweight channel-wise attention is well suited for distinguishing subtle patterns across multiple ciphertext pairs in resource-constrained cryptographic learning tasks.

C. Model Architecture

The proposed neural distinguisher is designed to effectively process multiple ciphertext pairs under a related-key setting. The architecture integrates three key components: an Inception-style convolutional block, a lightweight channel attention mechanism (ECA) (Fig. 1), and a residual convolutional tower. This combination aims to balance expressiveness and efficiency, allowing the model to learn both local differential features and global statistical patterns. among ciphertext pairs (Fig. 2).

Input Tensor: Each input sample is a tensor of shape $(k, 3, w)$, where k is the number of ciphertext pairs, 3 corresponds to the components $[\Delta C, C, C^*]$, and w is the block size (64 for PRESENT). The tensor is reshaped into a 1D sequence of $k \cdot w$ vectors, each with 3 channels, and processed using 1D convolutions across the pair axis.

Inception Block: The first module is a multi-branch convolutional block inspired by the Inception architecture [9]. It consists of parallel 1D convolutions with kernel sizes $\{1, 3, 5, 7\}$, enabling the model to extract features at multiple scales. Each branch applies batch normalization before the outputs are concatenated along the feature dimension. This design allows the model to capture both fine-grained and coarse-grained differential structures from the input sequence.

ECA Attention: Following the Inception block, the concatenated features are passed through an ECA module [8]. ECA dynamically reweights each feature channel by capturing local cross-channel interactions without dimensionality reduction. Specifically, it first applies global average pooling (GAP) across the sequence length to obtain a summary vector of channel statistics. This vector is then passed through a lightweight 1D convolution with an adaptively chosen kernel size $k = \psi(C)$ (where C is the number of channels), enabling local context modeling across neighboring channels.

Unlike conventional attention mechanisms that rely on fully connected layers, ECA introduces no dimensionality bottleneck and maintains computational efficiency. The resulting attention weights are activated through a sigmoid function and broadcast-multiplied with the original feature map.

This design is particularly well-suited for neural differential distinguishers, where subtle channel-wise dependencies between ciphertext features carry meaningful differential information. By adaptively emphasizing important channels, ECA improves the model's ability to generalize across variations in input pairs and round settings while keeping parameter overhead minimal.

Residual Tower: The attention-enhanced representation is then fed into a stack of residual convolutional blocks. Each block consists of Conv1D–BatchNorm–ReLU layers with skip connections. This tower increases the depth of the network while preserving gradient flow, enabling the model to learn higher-level abstractions over ciphertext distributions without degradation.

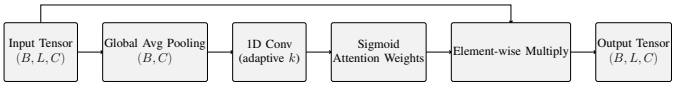

Fig. 1: Structure of the ECA module. Attention weights are generated by applying 1D convolution with adaptive kernel size to the global average pooled feature map, then reweighted back onto the input via element-wise multiplication.

Output Head: The final output is obtained by flattening the feature map and passing it through a fully connected layer with sigmoid activation. The model is trained using binary cross-entropy loss to distinguish between related and unrelated ciphertext samples.

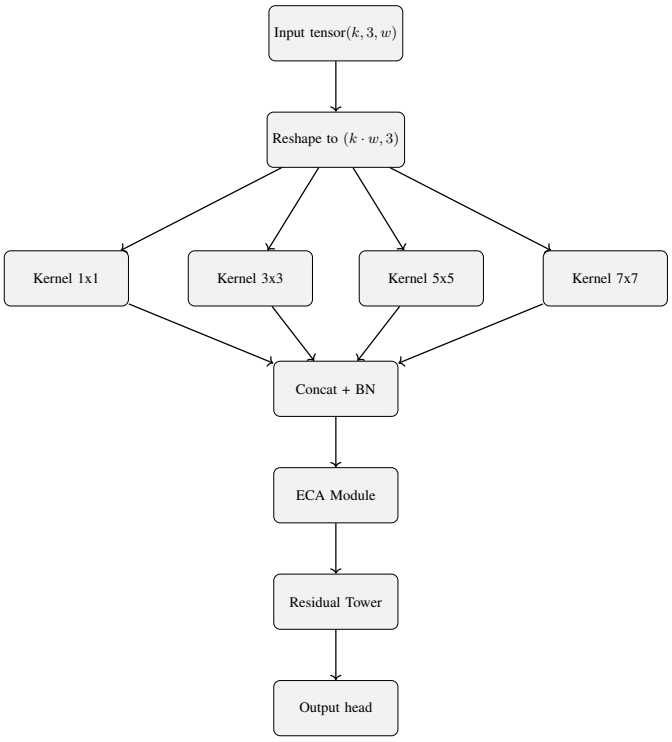

Fig. 2: Model architecture: ND_k input is reshaped and passed through a multi-branch Inception block, ECA attention, residual tower, and dense classifier.

D. Training and Architecture Hyperparameters

Table II and Table III summarize the core hyperparameters used in our experiments, covering both training configuration and model architecture. These settings were tuned empirically to achieve a balance between convergence speed, generalization, and training stability.

TABLE II: Training Hyperparameters

Parameter	Value
Optimizer	Adam (AMSGrad)
Initial Learning Rate	0.001
Learning Rate Schedule	Cyclic (triangular)
Number of Epochs	20
Batch Size	5,000
Loss Function	Binary Cross-Entropy
Validation Split	10%
Number of Training Samples	10^7
Dropout Rate	0.3
L2 Weight Decay	1×10^{-4}

TABLE III: Model Architecture Hyperparameters

Parameter	Value
Inception Filters	32
Inception Kernel Sizes	1, 3, 5, 7
Number of Residual Blocks	5
Residual Block Kernel Sizes	3, 5, 7, 9, 11
Convolution Filters per Block	128
Fully Connected Layers	$512 \rightarrow 64 \rightarrow 64$
Activation Function	ReLU
Final Output Activation	Sigmoid
Kernel Initializer	He Normal
Batch Normalization	Yes

IV. Experimenting

A. Setup and Protocol

All experiments are conducted on a workstation equipped with an NVIDIA A100 GPU and 40 GB of RAM, using Python 3.10 and TensorFlow 2.13. Training and data generation are fully GPU-accelerated through a custom CuPy-based implementation of the PRESENT-80 cipher, enabling high-throughput data generation and integration with model training.

Each sample consists of k ciphertext pairs under a fixed plaintext difference $\Delta P = \texttt{0x00000080}$ and a related-key difference $\Delta K = \texttt{0x00000000000000800000}$. This ΔK was selected using a PCA-based clustering approach inspired by Su et al. [5], which identifies key differences that maximize intra-class similarity in differential features.

Positive samples (label 1) are generated using related-key pairs (P, K) and (P^*, K^*) satisfying $P^* = P \oplus \Delta P$, $K^* = K \oplus \Delta K$. Negative samples use independent random values. All datasets are generated on-the-fly using our GPU pipeline.

Model accuracy is evaluated over five independent test sets, each with 10^6 samples, and we report both the average and standard deviation. Test conditions match the training configuration unless otherwise specified.

B. Evaluation Strategy

To ensure a meaningful comparison, we restrict all evaluations to the related-key setting. We benchmark against the framework of Su et al. [5], which uses ND_k inputs and a ResNet architecture. Their results on PRESENT-80 for rounds 7–9 serve as the primary baseline.

We examine the proposed model under two evaluation scenarios:

- **Effect of Number of Pairs (k):** We vary $k = 1, 2, 4, 8$ to analyze the impact of multiple ciphertext pairs on distinguishability.
- **Round Generalization:** We assess whether a model trained on $r = 9$ generalizes to higher rounds up to $r = 15$.

C. Multi-Round Comparison Across Different Ciphertext Pair Count (k)

To analyze the combined effect of round number and number of ciphertext pairs, we evaluate both our model and Su et al.'s RKND baseline [5] across $r = 7, 8, 9$ and $k \in \{1, 2, 4, 8\}$. Table IV summarizes the classification accuracy (%) on PRESENT-80. Each result is averaged over five test sets of 10^6 samples.

TABLE IV: Accuracy (%) Comparison Over (k, r) for Related-Key ND_k Models

Pairs (k)	Round 7	Round 8	Round 9
1	Su et al: 71.0 Ours: 95.0	Su et al: 57.0 Ours: 68.0	Su et al: – Ours: 54.0
2	Su et al: 80.0 Ours: 99.0	Su et al: 60.0 Ours: 76.0	Su et al: 50.2 Ours: 56.1
4	Su et al: 89.0 Ours: 99.0	Su et al: 65.0 Ours: 85.0	Su et al: 52.9 Ours: 59.1
8	Su et al: 96.0 Ours: 99.0	Su et al: 72.0 Ours: 92.0	Su et al: 54.4 Ours: 62.9

The results show that increasing the number of ciphertext pairs k consistently improves model accuracy across all rounds for both methods. Notably, our model consistently outperforms Su et al.'s baseline in every setting where a comparison is available.

The advantage of our architecture becomes more apparent as k increases. At round 9, for example, our model achieves 62.9% accuracy at $k = 8$, compared to 54.4% for Su et al., a relative gain of over 8.5 percentage points. This supports our hypothesis that multi-scale convolution and lightweight attention improve the model's ability to capture subtle related-key differential patterns, especially in deeper rounds where the distinguishing signal is weaker.

D. Effect of Ciphertext Pair Count (k) on Model Accuracy

We further investigate the role of the number of ciphertext pairs k used as input to our model. Table V summarizes the classification accuracy of our Inception + ECA architecture on PRESENT-80 for $r = 7$, $r = 8$, and $r = 9$, while varying k from 1 to 8.

TABLE V: Accuracy (%) of Our Model Across Different k Values

Pairs (k)	Round 7	Round 8	Round 9
1	95.0	68.0	54.0
2	99.0	76.0	56.1
4	99.0	85.0	59.1
8	99.0	92.0	62.9

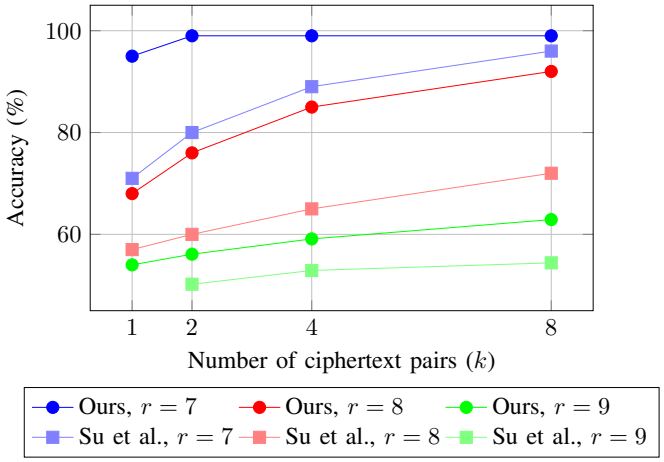

Fig. 3: Accuracy comparison of Su et al. and our model across different ciphertext pairs (k) and round numbers (r) on PRESENT-80. Circle markers denote our model; square markers represent Su et al.

The results in Fig. 3 confirm that increasing the number of ciphertext pairs per sample leads to consistent improvements in classification accuracy. The gains are particularly significant at higher rounds, where the distinguishing signal becomes weaker and redundancy across pairs becomes more beneficial.

At round 9, for example, increasing k from 1 to 8 improves accuracy from 54.0% to 62.9%—a relative increase of nearly 9 percentage points. This supports the hypothesis that ND_k representations with $k > 1$ provide richer structural features and facilitate more robust learning under related-key differentials. Fig. 4 illustrates the trend, showing consistent accuracy improvements with larger k.

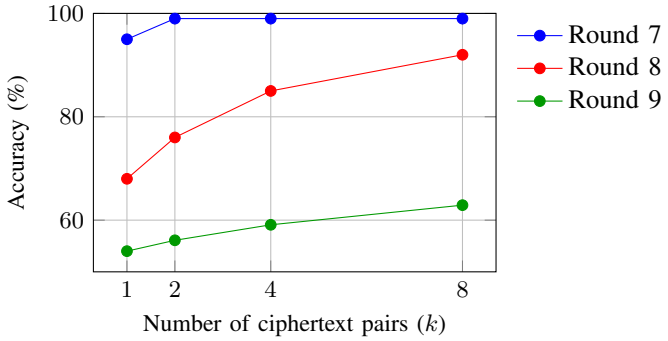

Fig. 4: Effect of ciphertext pair count (k) on model accuracy. Accuracy increases consistently with k, especially at higher rounds. Legend shown separately for compact layout.

E. Generalization to Higher Rounds

In this experiment, we evaluate whether a model trained at a moderate round number can generalize to deeper cipher configurations. Specifically, we train our model on round $r = 9$ using 8 related-key ciphertext pairs (ND_8), and evaluate it on unseen data generated at higher rounds $r \in \{10, 11, \ldots, 16\}$, without any fine-tuning.

Table VI reports the average accuracy and standard deviation across five independent test runs per round. The model shows a sharp accuracy drop from round 9 to 10, but maintains accuracy consistently above 50% up to round 15, suggesting that its predictions are not entirely random. At round 16, accuracy reaches parity, indicating that distinguishability vanishes.

TABLE VI: Test accuracy (%) of model trained at round 9, evaluated on higher rounds

Round	10	11	12	13	14	15	*16*
Accuracy	51.52	50.21	50.05	50.01	50.05	50.06	*49.99*
Std. Dev.	0.03	0.02	0.02	0.01	0.04	0.03	*0.03*

These results suggest that the learned representations are not strictly tied to the training round, but capture persistent structural features across rounds. This is in contrast to ND_1 models like GohrNet and DBitNet, which typically degrade to 50% much earlier. We hypothesize that the use of multiple related-key ciphertext pairs enables the model to extract deeper statistical cues that remain partially valid even under increased round diffusion.

The sharp accuracy decline between $r = 9$ and $r = 10$ suggests a transition point in statistical learnability. However, the model still maintains distinguishability slightly above random up to round 15 (Fig. 5), indicating residual feature from related-key structure. This contrasts with single-pair ND_1 models such as GohrNet, where accuracy typically degrades to random levels (50%) after just one or two additional rounds.

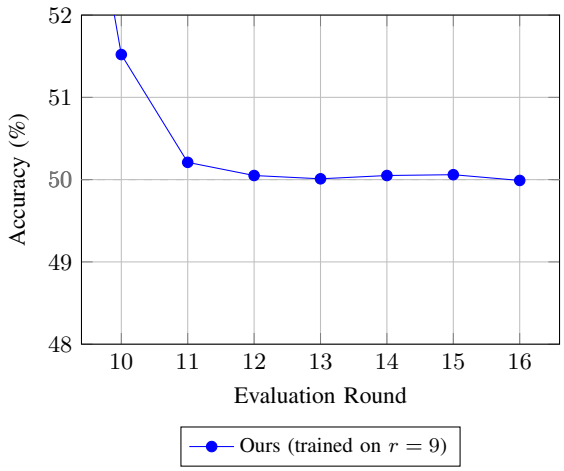

Fig. 5: Zoomed-in accuracy around 50% for model trained at $r = 9$ and evaluated on higher rounds. Dashed line at 50% marks the random guessing threshold.

F. Discussion

The experimental results validate the effectiveness of combining related-key differentials with multiple ciphertext pairs in neural cryptanalysis. Compared to the prior related-key ND_k

model by Su et al. [5], our proposed architecture achieves consistently higher accuracy across rounds 7–9 and all tested values of k. The improvement is most pronounced at round 9, where our model surpasses Su et al. by over 8% at $k = 8$ (Table IV).

In the round generalization experiment (Table VI), we observe that a model trained at round 9 retains distinguishability up to round 15. Although accuracy decays rapidly after round 10, it remains slightly above the 50% threshold for several deeper rounds, suggesting that the learned features are not entirely round-specific. This robustness contrasts with earlier single-pair models such as GohrNet or DBitNet, which typically degrade to random guessing beyond one or two additional rounds.

We attribute the strong performance of our model to three key factors:

- **Multi-pair input** ($k > 1$): Aggregates differential information across ciphertext pairs, improving the signal-to-noise ratio available for learning.
- **Related-key configuration** ($\Delta K \neq 0$): Introduces structured variations in ciphertext space that the network can learn to exploit.
- **Efficient architecture**: The combination of Inception-style parallel convolutions, residual connections, and ECA attention enables the model to extract multi-scale and channel-aware features while remaining compact.

Nevertheless, the model's advantage diminishes as the cipher rounds increase beyond its training range. By round 16, accuracy converges to 50%, indicating that the cipher's diffusion effectively suppresses all learnable patterns, even under related-key inputs.

These findings illustrate both the benefits and limitations of neural distinguishers in the related-key multi-pair setting. While our approach improves distinguishability in mid-round configurations, further work is needed to close the gap toward higher-round key recovery or generalized distinguishers beyond round 15.

V. Conclusion and Future Work

We proposed a neural distinguisher for the lightweight block cipher PRESENT-80, operating in the related-key setting with multiple ciphertext pairs per sample. Our model leverages an Inception-style convolutional block, residual connections, and efficient channel attention to process structured differential inputs. The combination of multi-pair representation and related-key differences enables richer statistical learning and enhances model robustness.

Experimental results demonstrate that our approach outperforms prior related-key baselines across rounds 7–9 and maintains distinguishability up to round 15, even without retraining. These findings confirm the utility of structured input design and lightweight architectural enhancements in improving differential neural cryptanalysis under data-constrained conditions.

For future work, we plan to extend this methodology to other lightweight ciphers such as GIFT and SKINNY, and integrate the model into the AutoND pipeline for automated input-difference discovery. More importantly, we are interested in exploring whether high-performing related-key neural distinguishes can be adapted to support key-recovery attacks. While RKNDs currently act as binary classifiers, further investigation into their integration with key search or ranking strategies could bridge the gap between distinguishers and full cryptanalysis.

Reproducibility and Code Availability

All code, model definitions, and training scripts used in this paper are available at: https://github.com/LongHaiTown/Related-key-mcp-attention-Inception-based-ND

References

[1] A. Gohr, "Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning," in *Advances in Cryptology – CRYPTO 2019*, Springer, pp. 150–179, 2019. [Online]. Available: https://eprint.iacr.org/2019/037

[2] E. Biham, "New types of cryptanalytic attacks using related keys," Journal of Cryptology, vol. 7, no. 4, pp. 229–246, 1994.

[3] J. Lu, G. Liu, B. Sun, C. Li, and L. Liu, "Improved (related-key) differential-based neural distinguishers for SIMON and SIMECK block ciphers," Computer Journal, vol. 67, no. 2, pp. 537–547, 2024.

[4] Pooja, S. Shantanu, and G. Mishra, "Related-Key Neural Distinguisher for Round-Reduced PRESENT Cipher," in Advances in Data-Driven Computing and Intelligent Systems, Springer, 2024, pp. 393–403.

[5] R.-T. Su, J.-J. Ren, and S.-Z. Chen, "Improved Framework of Related-key Differential Neural Distinguisher and Applications to the Standard Ciphers," *Cryptology ePrint Archive*, Report 2025/537. [Online]. Available: https://eprint.iacr.org/2025/537

[6] Y. Chen, Y. Shen, H. Yu, and S. Yuan, "A New Neural Distinguisher Considering Features Derived From Multiple Ciphertext Pairs," *The Computer Journal*, vol. 66, no. 6, pp. 1419–1433, 2023.

[7] H. Deng, X. Cao, and Y. Cheng, "Attention in Differential Cryptanalysis on Lightweight Block Cipher SPECK," in *Proc. 20th Int. Conf. on Privacy, Security and Trust (PST)*, IEEE, 2023, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/10320201

[8] Q. Wang et al., "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 11531–11539, 2020. doi: 10.1109/CVPR42600.2020.01155

[9] L. Zhang and Z. Wang, "Improving Differential-Neural Cryptanalysis," *Cryptology ePrint Archive*, Report 2022/183, 2022. [Online]. Available: https://eprint.iacr.org/2022/183

[10] A. Bogdanov et al., "PRESENT: An ultra-lightweight block cipher," in *Proc. Int. Workshop Cryptogr. Hardw. Embedded Syst. (CHES)*, Vienna, Austria, 2007, pp. 450–466.

[11] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. New York, NY, USA: Springer, 1993.

[12] E. Bellini, D. Gerault, A. Hambitzer, and M. Rossi, "A cipher-agnostic neural training pipeline with automated finding of good input differences," *IACR Trans. Symmetric Cryptol.*, vol. 2023, no. 3, pp. 184–212, 2023.

[13] Y. Guo, Y. Lu, W. Liu, W. Chen, and B. Yu, "Improved differential neural distinguishers for present and skinny," *Physica Scripta*, vol. 100, no. 6, p. 066003, 2025.

[14] C. Szegedy et al., "Going Deeper With Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.