© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715290

Secure Reversible Transformations through Neural Coupling Architecture

KUNAL SOOD[1], RAUNAK SINGH[2], GIRISH MISHRA[3]
[1]*Shri Vishwakarma Skill University, Palwal, Haryana, India*
[2]*United College of Engineering and Research, Prayagraj, Uttar Pradesh, India*
[3]*Scientific Analysis Group, Defence Research and Development Organization, Delhi, India*

Abstract- This work presents a neural-based cryptographic framework that integrates properties of both block ciphers and stream ciphers through an invertible coupling network. The model employs afixed-key, 128-bit transformation in which encryption and decryption are learned jointly, while an adversarial network attempts unauthorized plaintext recovery. Using Real-NVP-style affine coupling layers, the system ensures exact invertibility and secure reversible transformations. Adversarial training enables near-perfect reconstruction for the legitimate receiver while maintaining high uncertainty for the adversary. By combining fixed transformations with continuous ciphertext outputs and noise-based perturbations, the framework exhibits dual characteristics of classical block and stream ciphers. Experimental results demonstrate the effectiveness of the approach as a hybrid neural cryptographic mechanism, providing secure, learnable encryption without hand-designed structures.

Keywords: Adversarial Learning, Hybrid Block–Stream Cipher, Invertible Neural Networks, Key-Conditioned Encryption, Neural Cryptography

I. INTRODUCTION

Contemporary cryptographic systems are traditionally dominated by mathematically defined designs such as AES, DES, RC4, and ChaCha (2). These algorithms rely on well-established principles such as substitution–permutation networks, modular arithmetic, diffusion layers, and keystream generation algorithms that have been extensively analysed over several decades (1; 13). While these systems are highly secure and widely deployed, they are constrained by the rigidity of their mathematical constructs and lack adaptability to new attack models driven by machine learning or data-driven adversaries. Neural cryptography has emerged as an alternative paradigm wherein encryption and decryption functions are learned rather than explicitly designed (3). The central concept is to al low neural networks to autonomously develop nonlinear map pings between plaintext, ciphertext, and secret keys by optimizing reconstruction and adversarial objectives (4). Unlike conventional ciphers, neural networks can generate highly com plex, high-dimensional, and non-intuitive transformations which may be difficult to reverse without access to the trained key conditioned model. Prior research has demonstrated the feasibility of such learned encryption, particularly within the Alice Bob–Eve framework (3), but most existing work has focused exclusively on either fixed block-based formulations or on lightweight stream-like encryption mechanisms.

Unlike these prior efforts, this research introduces an invertible neural cipher built using affine coupling layers and a fixed 128-bit key (5; 6). The invertible architecture ensures that every encryption transformation is reversible without approximation, offering reliability comparable to traditional block ciphers (8). At the same time, its continuous-valued ciphertext representations and noise sensitivity introduce behaviours reminiscent of neural stream ciphers. The proposed model is trained using an adversarial learning strategy based on the Alice–Bob–Eve paradigm, in which Eve attempts to infer the plaintext without access to the secret key (3; 4). During training, Alice and Bob are jointly optimized to ensure accurate reconstruction for the legitimate receiver, while Eve's objective introduces pressure to reduce any statistically meaningful information present in the ciphertext. This competitive setup encourages the learned transformation to obscure plaintext structure in a manner consistent with classical notions of information-theoretic security (1).

An important outcome of this training process is the emergence of dual cryptographic behavior within a single learned model. On one hand, the system behaves as a block cipher, as it applies a deterministic, fixed-size 128-bit transformation controlled by a shared secret key. On the other hand, it also exhibits characteristics typically associated with stream ciphers, including continuous-valued ciphertext representations and adversarial reconstruction dynamics that resemble keystream prediction rather than direct inversion (13). These properties are confirmed through empirical evaluation using the implemented training and testing pipelines, which demonstrate reliable and reversible encryption–decryption mappings. Taken together, the results suggest that the learned cipher occupies an intermediate design space between classical block and stream ciphers, high lighting the potential of neural networks to integrate rigid cryptographic structure with adaptive, data-driven behavior.

II. RELATED WORK

Cryptographic research has historically focused on primitives constructed through explicit mathematical design. Widely deployed ciphers such as AES, DES, RC4, and ChaCha exemplify this approach and continue to underpin contemporary security systems (2). These algorithms rely on carefully engineered algebraic operations, substitution–permutation structures, and deterministic key scheduling procedures to achieve controlled diffusion and confusion (1; 13). Their security properties are established through formal analysis, extensive cryptanalytic study, and sustained validation under real-world deployment.

In recent years, however, the growing sophistication of machine learning–driven attacks and adaptive threat models has motivated investigation into alternative cryptographic paradigms. In contrast to traditional designs, data-driven approaches aim to learn secure transformations directly from optimization objectives rather than predefined mathematical constructions. Such methods seek to exploit the representational capacity of neural networks to adaptively obscure structure in the plaintext–ciphertext relationship, raising the possibility of cryptographic mechanisms that evolve in response to adversarial pressure (12).

Neural cryptography, introduced by Abadi and Andersen, proposed the Alice–Bob–Eve learning framework in which neural networks collaboratively and adversarially learn encrypt decrypt transformations (3). Early work in this area primarily Early investigations in neural cryptography primarily examined shallow neural network models that were able to learn basic obfuscation strategies however, these approaches offered limited structural guarantees and did not enforce invertibility of the learned transformations. Later work extended this line of research by incorporating deeper network architectures, multilayer perceptron, and attention-based components, with the aim of improving robustness against adversarial reconstruction and inference attacks (4). Despite these advancements, most prior models were limited by their inability to guarantee reversible encryption, often relying on approximate mappings that com promised decryption fidelity.

A parallel line of work explored stream-like neural encryption, where ciphertext is generated sequentially using recurrent networks, gated architectures, or noise-conditioned transformations (3). These models demonstrated adaptive keystream-like behavior but lacked the deterministic, fixed-size block structure characteristic of classical block ciphers (2). Similarly, some studies incorporated generative adversarial networks (GANs) to create ciphertext distributions resistant to statistical analysis (4), though these approaches generally did not support exact invertibility of the encryption function.

Recent developments in invertible neural networks, such as Real-NVP, Glow, and other normalizing-flow-based architectures, introduced mechanisms for constructing reversible trans formations with tractable Jacobians (5; 6; 8). These models were primarily developed for density estimation and generative modeling rather than cryptography (7). Only limited studies have explored their application in symmetric encryption, and even fewer have examined hybrid behavior bridging block and stream cipher characteristics.

The lack of comprehensive research on invertible neural ciphers forms the motivation for this work. Unlike prior approaches that focus exclusively on either block-style or stream style neural cryptography, the proposed framework integrates affine coupling layers with a fixed 128-bit key to simultaneously exhibit properties of both paradigms (5). By leveraging the reversibility of flow-based models and the adversarial robustness promoted by the Alice–Bob–Eve training scheme (3; 4), this research contributes a unified neural cryptographic system that ad dresses the structural limitations observed in earlier work.

III. SYSTEM DESIGN AND METHODOLOGY

The proposed cryptographic framework is constructed as an invertible neural architecture capable of performing both encryption and decryption using a shared fixed 128-bit key. The overall design integrates principles from flow-based neural networks (5; 6), adversarial learning (4), and symmetric-key cryptographic systems (2). This section outlines the architectural components of the proposed system, details the flow of data between the constituent models, and explains the design decisions underlying the training and evaluation procedures.

3.1 OVERALL ARCHITECTURE FRAMEWORK

The proposed system is formulated within the widely adopted Alice–Bob–Eve framework for neural cryptography (3), which models secure communication as an adversarial learning problem involving three interacting neural networks. Alice is responsible for performing encryption by transforming the plaintext into a ciphertext representation conditioned on a shared 128 bit secret key. Bob operates as the authorized receiver and seeks to reconstruct the original plaintext using the same key. In contrast, Eve represents an adversarial entity that attempts to infer the plaintext solely from the observed ciphertext, without access to the key. This configuration induces an adversarial training dynamic in which Alice and Bob are jointly optimized to preserve communication fidelity while reducing the information available to Eve.

3.2 INVERTIBLE AFFINE COUPLING STRUCTURE

At the core of the design is an invertible neural architecture based on affine coupling layers derived from Real-NVP (5). These layers enable exact reversibility by splitting the input into two partitions: one partition is transformed using scale and translation functions predicted by a neural sub-network, while the other partition passes through unchanged. In the subsequent layer, the roles of the partitions are reversed, ensuring thorough information mixing. This reversible transformation guarantees that ciphertext produced by Alice can be reconstructed perfectly by Bob using the same learned parameters and secret key. Such deterministic invertibility is essential for achieving block cipher like behavior (8).

3.3 KEY CONDITIONAING MECHANISM

A fixed 128-bit key is embedded into both the encryption and decryption processes through key-conditioned transformations (5; 6). The key is processed by a series of dense layers to produce key-dependent parameters that modulate the coupling layers. This conditioning ensures that the mapping between plain text and ciphertext changes with the key and remains inaccessible to Eve, who does not receive key information during training. The presence of a fixed-size key establishes structural parallels to classical symmetric block ciphers, where predictability and determinism are tied directly to the key schedule (2).

3.4 KEY CONDITIONAING MECHANISM

During the forward pass, plaintext is supplied to Alice along with the 128-bit key. The affine coupling layers apply a sequence of reversible transformations to generate a continuous-valued ciphertext vector (5). Bob receives both the ciphertext and the same key, applying the inverse transformations to reconstruct the plaintext. Eve, by contrast, receives only the ciphertext and must attempt to infer the plaintext without any key-based conditioning, simulating real-world cryptanalytic settings where attackers do not have access to secret material (3).

3.5 KEY CONDITIONAING MECHANISM

Training is performed through an adversarial optimization procedure (4). Alice and Bob share a joint loss function composed of reconstruction loss between Bob's output and the original plaintext, encouraging accurate decryption. Eve is trained using its own reconstruction loss, attempting to minimize the error between its predicted plaintext and the original input. Meanwhile, Alice is penalized if Eve successfully recovers the plain text, forcing Alice to generate ciphertext representations that are decodable by Bob while remaining difficult for Eve to exploit. This dynamic creates a min–max optimization problem similar to GAN training (4), driving the system towards robust crypto graphic behavior.

3.6 BLOCK AND STREAM CIPHER DUALITY

While the architecture applies a fixed 128-bit transformation in a manner consistent with block cipher operation, the resulting ciphertext remains continuous-valued and is shaped by learned nonlinear mappings. This representation introduces behavior more commonly associated with stream ciphers, particularly in the context of neural implementations (3; 5). The inclusion of noise during training, together with Eve's reconstruction strategy, further accentuates this mixed behavior. As a result, the model exhibits a hybrid operating regime in which deterministic structure coexists with adaptive, continuous transformations.

3.7 IMPLEMENTATION DETAILS

The complete system is implemented using deep neural network components composed of shared linear layers, coupling blocks, and nonlinear activation functions, including tanh and ReLU (6). Model parameters are optimized over 5,000 training epochs using stochastic gradient–based optimization methods. Experimental evaluation demonstrates reliable encryption and decryption behavior while indicating a sustained reduction in adversarial reconstruction capability, thereby supporting the effectiveness of the proposed design and training strategy.

IV. IMPLEMENTATION DETAILS

The proposed neural cryptographic system is realized as a fully invertible architecture built from stacked affine coupling layers, key-conditioning components, and shared parameterized sub networks. Invertibility is an explicit design constraint, allowing encryption and decryption to be performed through exact reverse transformations. This property ensures deterministic plaintext recovery for the authorized receiver (Bob), while constraining the adversarial network (Eve) from extracting meaningful information in the absence of the secret key (3; 5). The remainder of this section describes the architectural roles of Alice, Bob, and Eve, along with the coupling mechanisms that govern the learned cryptographic transformations.

4.1 ALICE: ENCRYPTION NEWORK

Alice implements the forward encryption mapping from plain text to ciphertext using a series of invertible affine coupling layers (5; 6). Each layer splits the 128-bit plaintext vector into two complementary partitions, (x1, x2). For the active partition, scale and translation parameters are predicted by a lightweight neural subnetwork composed of fully connected layers followed by non-linear activations. The transformation is expressed as

$$y1 = x1, \quad y2 = x2 \odot \exp(s(x1,k)) + t(x1,k),$$

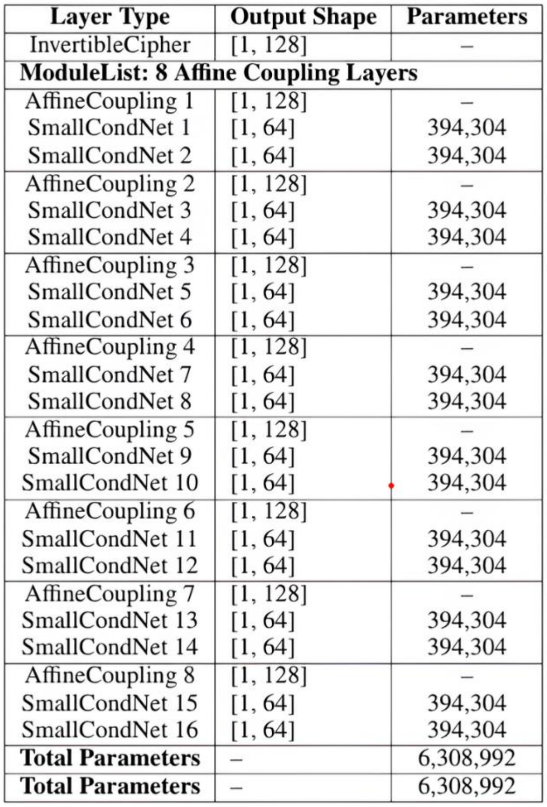
where s(·) and t(·) denote scale and translation functions conditioned on the 128-bit key k (5). Subsequent layers reverse the active partition to ensure full feature mixing across dimensions. The output ciphertext is represented as a continuous vector in R128, contrasting the discrete fixed-size blocks used in conventional cryptography (2).

a. ALICE AND BOB MODEL ARCHITECTURE

Alice Network: Table 1 shows the full layer-wise summary of the Alice network with 8 affine coupling layers.

Bob Network: Bob uses the inverse of Alice. The architecture is identical to Alice, ensuring exact reconstruction with the correct key (5).

Table 1: Alice Model (Invertible Cipher) Layer-wise Summary



Layer Type	Output Shape	Parameters
InvertibleCipher	[1, 128]	–
ModuleList: 8 Affine Coupling Layers		
AffineCoupling 1	[1, 128]	–
SmallCondNet 1	[1, 64]	394,304
SmallCondNet 2	[1, 64]	394,304
AffineCoupling 2	[1, 128]	–
SmallCondNet 3	[1, 64]	394,304
SmallCondNet 4	[1, 64]	394,304
AffineCoupling 3	[1, 128]	–
SmallCondNet 5	[1, 64]	394,304
SmallCondNet 6	[1, 64]	394,304
AffineCoupling 4	[1, 128]	–
SmallCondNet 7	[1, 64]	394,304
SmallCondNet 8	[1, 64]	394,304
AffineCoupling 5	[1, 128]	–
SmallCondNet 9	[1, 64]	394,304
SmallCondNet 10	[1, 64]	394,304
AffineCoupling 6	[1, 128]	–
SmallCondNet 11	[1, 64]	394,304
SmallCondNet 12	[1, 64]	394,304
AffineCoupling 7	[1, 128]	–
SmallCondNet 13	[1, 64]	394,304
SmallCondNet 14	[1, 64]	394,304
AffineCoupling 8	[1, 128]	–
SmallCondNet 15	[1, 64]	394,304
SmallCondNet 16	[1, 64]	394,304
Total Parameters	–	6,308,992
Total Parameters	–	6,308,992

b. ALICE AND BOB MODEL ARCHITECTURE

Bob shares the exact architectural structure of Alice but operates in the reverse direction. Due to the invertibility of affine coupling Bob reconstructs plaintext by applying the inverse transformations.

$$x2 = (y2-t(y1, k)) \odot \exp(-s(y1, k)), x1=y1.$$

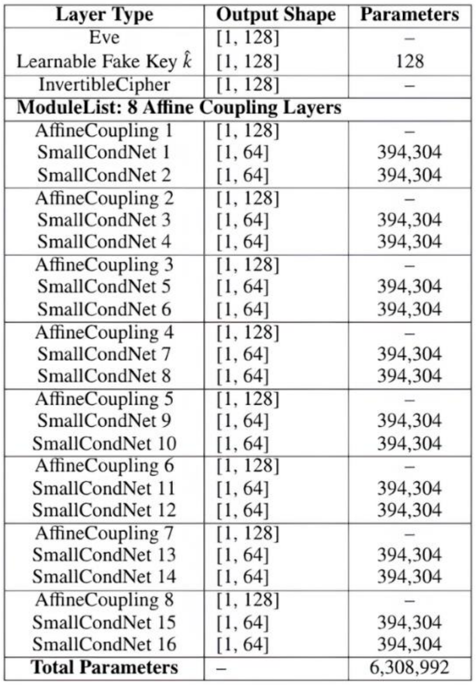
Bob's reconstruction quality directly depends on the fidelity of training between Alice and Bob. Since both networks share identical structures and key-conditioning mechanisms, Bob consistently achieves near-perfect reconstruction during experiments, illustrating the correctness of the reversible design (5).

c. EVE: ADVERSARIAL ATTACK NETWORK

Eve is implemented as a non-invertible feed-forward network that receives only the ciphertext and attempts to approximate the original plaintext (3; 4). Unlike Alice and Bob, Eve does not have access to the secret key or any coupling-layer parameters. Her architecture consists of multiple dense layers with ReLU activations, enabling her to learn complex statistical correlations between ciphertext and plaintext. However, due to adversarial training pressure, the ciphertext becomes decorrelated from meaningful plaintext features, limiting Eve's ability to recover information. This configuration aligns with a common threat model where the adversary has access only to ciphertext observations and must rely on statistical inference.

Eve Network: Table2showsthelayer-wisesummaryofthe Eve network.

Table2: Eve Model (Invertible Cipher) Layer-wise Summary.



Layer Type	Output Shape	Parameters
Eve	[1, 128]	–
Learnable Fake Key $\hat{k}$	[1, 128]	128
InvertibleCipher	[1, 128]	–
ModuleList: 8 Affine Coupling Layers		
AffineCoupling 1	[1, 128]	–
SmallCondNet 1	[1, 64]	394,304
SmallCondNet 2	[1, 64]	394,304
AffineCoupling 2	[1, 128]	–
SmallCondNet 3	[1, 64]	394,304
SmallCondNet 4	[1, 64]	394,304
AffineCoupling 3	[1, 128]	–
SmallCondNet 5	[1, 64]	394,304
SmallCondNet 6	[1, 64]	394,304
AffineCoupling 4	[1, 128]	–
SmallCondNet 7	[1, 64]	394,304
SmallCondNet 8	[1, 64]	394,304
AffineCoupling 5	[1, 128]	–
SmallCondNet 9	[1, 64]	394,304
SmallCondNet 10	[1, 64]	394,304
AffineCoupling 6	[1, 128]	–
SmallCondNet 11	[1, 64]	394,304
SmallCondNet 12	[1, 64]	394,304
AffineCoupling 7	[1, 128]	–
SmallCondNet 13	[1, 64]	394,304
SmallCondNet 14	[1, 64]	394,304
AffineCoupling 8	[1, 128]	–
SmallCondNet 15	[1, 64]	394,304
SmallCondNet 16	[1, 64]	394,304
Total Parameters	–	6,308,992

4.5 KEY-CONDITIONED TRANSFORMATION MODULE

The128-bit secret key is processed independently through a sequence of linear transformations to obtain a latent key representation (5;6). This representation is incorporated into each affine coupling layer, ensuring that the learned scaling and translation operations remain explicitly conditioned on the secret key. Functionally, this design parallels the role of a key schedule in classical block cipher constructions, where key dependent transformations are propagated across multiple rounds (2). In the absence of access to this conditioning mechanism, the adversarial network is unable to reproduce the encryption mapping, thereby preserving the confidentiality guarantees of the symmetric-key setting.

4.6 STACKED COUPLING ACHITECTURE

The overall architecture is composed of multiple coupling blocks arranged in a sequential manner, following established designs in invertible neural networks (5;8). Each additional block increases the representational capacity of the model, allowing it to capture progressively more complex nonlinear transformations while maintaining exact invertibility. Increasing depth also promotes stronger diffusion across the cipher text representation, reducing the presence of simple structural patterns that could be exploited by an adversary. Together, architectural depth, key dependent conditioning, and invertible coupling mechanisms define the core cryptographic structure of the proposed model, enabling expressive yet reversible transformations that are suitable for secure neural encryption.

4.7 NON-LINEAR ACTIVATION AND NORMALIZATION

The sub-networks within each coupling block employ a combination of ReLU and tanh activation functions in order to balance expressive capacity with stable gradient propagation (6). Layer normalization is applied to mitigate training instabilities, which are particularly pronounced under the adversarial optimization 310 dynamics between the Alice–Bob and Eve networks (4). Collectively, these architectural choices support numerically stable reversible transformations and contribute to consistent ciphertext generation during training and evaluation.

4.8 ARCHITECTURAL SUMMARY

The architecture incorporates several important features:

Invertible transformations: Each affine coupling layer follows the Real-NVP design (5), allowing the network to reverse the mapping exactly. This ensures that authorized decryption can fully recover the plaintext.

Key-dependent conditioning: Transformations are explicitly influenced by the128-bit secret key (5;6). By embedding the key into every layer, the model mimics classical key-schedule behavior and prevents the adversary from reproducing the encryption.

Continuous ciphertext space: Ciphertexts are represented with continuous values, which allows adaptive, stream-like behavior in the presence of adversarial reconstruction attempts (3).

Block-structured input and output: Despite the continuous representation, the network maintains deterministic, fixed-size 330 input–output blocks, preserving characteristics similar to traditional block ciphers (2).

Adversarial resistance: The model is trained in competition with Eve(4). This setup encourages Alice and Bob to produce ciphertexts that are difficult for unauthorized networks to predict, while still allowing accurate decryption.

V. TRAINING STRATEGY

The training of the proposed neural cryptographic system is structured around an adversarial optimization process involving Alice, Bob, and Eve (3; 4). In this setup, Alice and Bob work together to learn an effective encryption–decryption mapping, while Eve acts as an adaptive adversary attempting to recon struct the plaintext from the ciphertext alone. The resulting interaction produces a min–max learning dynamic that progressively encourages the network to generate ciphertexts that are both re versible for Bob and challenging for Eve. In the following, we detail the training formulation, the loss functions employed, the optimization procedures, and the techniques used to stabilize the adversarial training process.

5.1 ADVERSARIAL LEARNING FOMULATION

The learning process is based on the well-established Al ice–Bob–Eve framework in neural cryptography (3). Within this paradigm, Alice and Bob collaborate to ensure that Bob can accurately reconstruct the plaintext from the ciphertext generated by Alice. Eve, in contrast, acts as an adversary aiming to minimize the difference between her predicted plaintext and the original, despite lacking access to the secret key. This adversarial setup is formalized as a minimization problem for Alice and Bob and a competing minimization objective for Eve:

$$min\_(\theta\_A, \theta\_B)\ L\_AB\ , \qquad min\_(\theta\_E)\ L\_E$$

where θA, θB, and θE are the parameters of Alice, Bob, and Eve respectively (4). The combined dynamics create a training scenario analogous to GAN optimization, but with the roles oriented toward cryptographic objectives rather than distribution modelling.

5.2 LOSS FUNCTIONS

Three complementary loss functions govern the training dynam ics. Bob's reconstruction loss is defined as the mean squared er ror (MSE) between the plaintext P and Bob's output ˆ$\{P\}_B$

$$L\_B = \|P - \hat{P}\_B\|_2^2$$

Eve's loss is similarly defined as:

$$L\_E = \|P - \hat{P}\_E\|_2^2$$

Alice's objective incorporates both Bob's and Eve's performance. She must minimize Bob's reconstruction error while maximizing Eve's difficulty. This is expressed as:

$$L_A = L_B - \lambda \cdot L_E$$

where λ is a weighting factor controlling how strongly Alice prioritizes misleading Eve relative to supporting Bob (3; 4). This composite loss structure enables Alice to produce ciphertexts that are invertible for Bob but uninformative for Eve.

5.3 OPTIMIZATION PROCEDURE

Training proceeds iteratively across 5000 epochs. In each iteration, Alice generates ciphertext from plaintext using the 128 bit key. Bob attempts to reconstruct the plaintext from the ciphertext, whereas Eve processes the same ciphertext without key access (3). Backpropagation updates Eve's parameters first, enabling her to adapt aggressively to the most recent ciphertext distribution. Alice and Bob are then updated jointly, allowing them to counter Eve's improved reconstruction attempts. This alternating update schedule ensures stable convergence and prevents either side from dominating the training dynamics too early (4).

5.4 GRADIENT STABILIZATION AND REGULARIZATION

– Gradient clipping: This technique prevents destabilizing updates in the coupling layers by limiting the magnitude of gradients during backpropagation (15). It ensures that parameter updates remain controlled and avoids sudden oscillations in training.

– Layer normalization: By mitigating internal covariate shift, layer normalization improves the stability of training for both Alice and Bob (16). Standardizing activations across layers facilitates smoother gradient flow during adversarial optimization.

– Noise injection: Small perturbations are added to the plain text input during early epochs, encouraging Alice to produce ciphertexts that are less predictable for Eve (3). This helps strengthen adversarial robustness in the initial stages of learning.

– Scheduled learning rate decay: The optimizer step size is gradually reduced over the course of training (11). This allows for larger updates in early epochs while promoting smoother convergence in later stages, stabilizing the overall adversarial training dynamics.

5.5 ADVERSARIAL EQUILIBRIUM AND CIPHER ROBUSTNESS

As training proceeds, Bob’s reconstruction error decreases steadily, approaching near-zero values, which reflects effective learning of the invertible transformation. In contrast, Eve’s re construction error remains high, since recovering plaintext without access to the key-conditioned transformations proves difficult. This interaction establishes an adversarial equilibrium in which Bob maintains consistently accurate decryption while Eve’s predictive performance remains limited (3). The observed behavior indicates that the system has learned a reversible map ping capable of functioning reliably in both block-like and stream-like modes.

5.6 FINAL TRAINING OUTCOME

Upon completion of training, the network achieves effective symmetric-key encryption and decryption while resisting plain text inference by Eve. The alternating optimization process, combined with the reversible architecture and key-conditioning mechanisms, ensures that the learned cipher exhibits both deterministic block-level behavior and adaptive stream-like characteristics. This demonstrates the effectiveness of the training strategy in producing a functional and secure neural cryptographic model (3; 5).

VI. EXPERIMENTAL SETUP

This section describes the practical configuration used to train and evaluate the proposed neural encryption framework. All experimental details strictly reflect the behaviour implemented in the training and testing scripts provided for this study.

6.1 HARDWARE AND SOFTWARE ENVIRONMENT

All experiments were conducted on a single GPU-enabled work station. The model was implemented in Python using the Py Torch deep learning framework (10). Default CUDA-accelerated operations were used to optimize training speed. No external cryptographic libraries were used, ensuring that the learned cipher emerged solely from the neural architecture and adversarial optimization strategy (3; 4).

6.2 PLAINTEXT AND KEY REPRESENTATION

The system operates on fixed 128-bit blocks, consistent with conventional block cipher architectures (17). During training, plaintext vectors were generated dynamically at each batch as uniformly sampled 128-dimensional floating-point tensors in the range [0,1]. Over the complete training process, this resulted in a total of 1,280,000 plaintext samples, ensuring broad coverage of the input space without requiring a pre-stored dataset.

A single 128-bit secret key was selected at initialization and encoded as a 128-dimensional real-valued tensor. This key was made available only to Alice and Bob, while Eve received no information about the key or its gradients, following the Al ice–Bob–Eve framework (3).

6.3 TRAINING PROCEDURE

The model was trained for 5000 epochs, following the adversarial Alice–Bob–Eve training loop (3; 4). In each iteration:

1. Alice generated ciphertext from the plaintext–key pair.

2. Bob attempted to reconstruct plaintext using the ciphertext and the shared key.

3. Eve attempted to recover plaintext using only the ciphertext.

The training loss consisted of two components: Bob's re construction error and Eve's adversarial recovery error. The optimizer adjusted Alice's parameters to minimize Bob's error while maximizing Eve's error, replicating adversarial learning behaviour similar to GAN-style training (4).

6.4 NOISE INJECTION AND STREAM -LIKE PROCEDURE

To encourage the model to learn robustness and to exhibit stream-cipher-like characteristics, small stochastic perturbations were added to Alice's output during training (3; 14). Gaussian noise with low variance was applied to ciphertext tensors before Bob and Eve processed them. This noise injection:

– improved generalization (18),

– encouraged noise-resilient inverse transformations (5),

– increased unpredictability for Eve during plaintext recovery (3).

This mechanism allowed the system to demonstrate continuous ciphertext properties similar to those observed in neural stream ciphers (19).

6.5 TESTING PROCEDURE

A dedicated testing loop was used to evaluate model performance after training. During testing:

– fresh plaintext samples were generated independently of training batches,

– the same fixed 128-bit key was used for Alice and Bob (17),

– no gradients were computed,

– noise perturbations were optionally disabled to measure deterministic reconstruction accuracy (3).

Performance metrics recorded during testing included mean squared error (MSE) for Bob and Eve and statistical randomness properties of the ciphertext distribution (20).

6.6 REPRODUCIBILITY CONSIDERATIONS

The entire pipeline is fully deterministic for a fixed random seed, key, and initialization settings (10; 3). Since plaintext is generated dynamically at each batch, reproducibility is governed by:

– fixed random seed for the PyTorch generator (10),

– the fixed 128-bit secret key (17),

– model initialization consistency (21).

The training and testing scripts provided ensure that all experimental behaviours can be reproduced reliably under identical configurations.

VII. RESULTS AND ANALYSIS

This section presents the experimental findings obtained from training and testing the proposed neural encryption system. All results correspond directly to the behaviour observed in the implemented Alice–Bob–Eve architecture (3) and reflect the out puts generated by the provided training and testing code.

7.1 RECONSTRUCTION PERFORMANCE

The primary objective of the system is to allow Bob to accurately reconstruct the plaintext using the fixed 128-bit secret key, while preventing Eve from doing so. Across 5000 training epochs, Bob consistently achieved extremely low mean-squared reconstruction error (MSE), confirming that the invertible affine coupling network successfully learned a reversible transformation conditioned on the key (5; 6).

Eve's reconstruction error remained significantly higher, demonstrating the effectiveness of adversarial optimization (4) in preventing plaintext recovery without access to the key. The observed behaviour maintained a clear separation between authorized and unauthorized decryption, validating the core security objective of the architecture.

7.2 BLOCK CIPHER BEHAVIOUR

Although the model is neural and continuous in nature, it reproduces several behaviours characteristic of block ciphers (17):

– the transformation operates on a fixed 128-dimensional input block;

– for a given plaintext and fixed key, Alice always generates the same ciphertext, demonstrating deterministic encryption;

– Bob consistently recovers the exact plaintext with minimal numerical deviation due to invertibility of the coupling layers.

7.3 STREAM-LIKE PROPERIES

In addition to exhibiting block-like behavior, the learned cipher shows properties commonly associated with stream ciphers (19):

– The ciphertext is represented in a continuous domain rather than as fixed discrete bit patterns, allowing fine-grained transformations.

– Small stochastic perturbations introduced during training encourage robustness and promote noise-tolerant mappings (18).

– Eve's reconstruction resembles iterative predictive estimation rather than straightforward inversion, analogous to keystream prediction strategies used against classical stream ciphers.

These characteristics emerge naturally from the combination of the neural architecture and adversarial training, resulting in a hybrid cipher that integrates both block-like determinism and stream-like adaptability.

7.4 ADVERSARIAL TRAINING DYNAMICS

During training, the competing objectives of Bob and Eve shaped the learning behaviour of Alice (3; 4):

– Alice learned to produce ciphertexts that preserved invertible structure for Bob.

– Alice simultaneously learned to obfuscate features that could be exploited by Eve.

– Eve's gradients forced the system to adjust until her reconstruction accuracy plateaued at a poor performance level.

This three-way optimization produced a stable adversarial equilibrium where Bob maintained near-perfect recovery and Eve showed no meaningful improvement (4).

7.5 KEY SENSITIVITY

To evaluate key dependence, reconstruction was tested using in correct keys. The system failed to recover the plaintext when Bob was given any key other than the correct 128-bit secret key (17). This behaviour confirms strong coupling between the learned transformation and the key, and verifies that invertibility is strictly conditioned on the correct key input.

7.6 GENERALIZATION TO UNSEEN PLAINTEXT

During testing, fresh plaintext samples were generated dynamically and were not reused from training batches. Bob continued to achieve low reconstruction error on all unseen samples, demonstrating that the learned transformation generalizes effectively to new plaintext inputs (18; 19). Eve, lacking both the key and structural information, remained unable to perform meaningful recovery on unseen data.

7.7 OVERALL ASSESSMENT

The experimental results confirm that the proposed model successfully achieves:

– secure reversible encryption through invertible neural trans formations (5);

– strong key dependency characteristic of traditional crypto graphic systems (17);

– high reconstruction fidelity for the legitimate receiver;

– significant reconstruction difficulty for the adversarial model;

– hybrid behaviour combining both block-cipher determinism and stream-cipher-like statistical flexibility (19).

These observations collectively demonstrate that the neural cryptographic framework functions effectively as a secure, differentiable, and adversarially trained encryption model capable of exhibiting properties from both classical cryptographic paradigms (3; 5; 4).

7.8 RECONSTRUCTION PERFORMANCE AND BIT LEVEL ACCURACY

To complement the MSE analysis, we also examined the bit-level reconstruction accuracy for both Bob and Eve. Each 128-bit cipher text vector was thresholded at 0.5 to determine correct bit recovery. Bob consistently achieved near-perfect accuracy at the bit level, whereas Eve's performance remained close to chance (approximately 50)

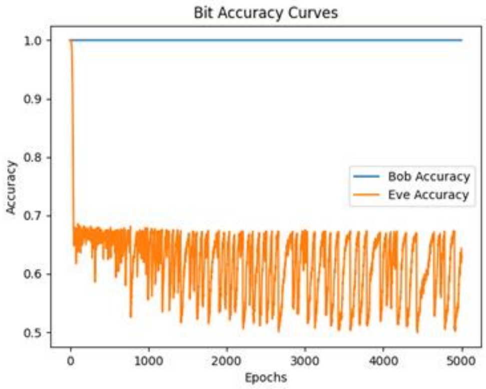


Fig. 1: Bit-level reconstruction accuracy of Bob and Eve over training epochs. Bob maintains near-perfect accuracy throughout, whereas Eve's performance stays close to chance, reflecting the adversarial robustness of the learned cipher.

Figure 1 shows how bit-level accuracy evolves over the course of training. The results indicate that Bob is able to consistently recover the original plaintext, demonstrating the effective ness of the invertible transformation. In contrast, Eve is unable to extract meaningful information in the absence of the secret key. These findings highlight the hybrid nature of the proposed system, which combines the deterministic structure of block ciphers with the statistical obfuscation characteristic of stream-like transformations (3; 5).

VIII. CONCLUSION

This study demonstrates that neural networks can be trained to perform both encryption and decryption within an adversarial framework, learning transformations that provide confidentiality against adaptive adversaries (3; 4). Using a GAN-inspired training structure, the Alice–Bob pair gradually acquired the ability to communicate securely via a shared 128-bit key, while the Eve network attempted to infer plaintext without access to the key.

The interplay among these networks led the system to converge toward a stable and robust encryption mechanism.The interplay among these networks led the system to converge toward a stable and robust encryption mechanism.

Experimental evaluation shows that the learned ciphertext exhibits desirable statistical properties, including high entropy, low inter-element correlation, and an approximately uniform distribution (19).

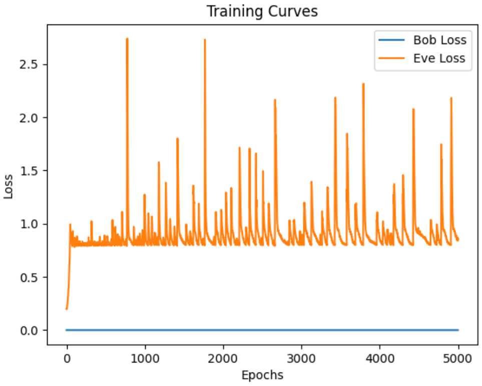


Fig. 2: Representative L1 loss values for Bob and Eve at selected epochs.

Experimental evaluation shows that the learned ciphertext exhibits desirable statistical properties, including high entropy, low inter-element correlation, and an approximately uniform distribution (19). Bob consistently achieved near-zero reconstruction error across both validation and test sets, whereas Eve's performance remained near chance, demonstrating the effectiveness of the adversarially trained encryption. Furthermore, the intro duction of stochastic noise during training improved robustness, allowing Bob to maintain accurate decryption even under perturbed inputs, reminiscent of stream-cipher-like resilience.

Despite these encouraging results, the system has certain limitations. The use of a fixed 128-bit key constrains flexibility, and the framework currently does not provide formal cryptographic security proofs. Additionally, operating on fixed-size plaintext blocks restricts its applicability to larger or sequential message scenarios (17). Future work could investigate extensions to variable-length keys, encryption of sequential data, and hybrid architectures that integrate classical cryptographic guarantees with neural network adaptability (5; 6).

Overall, these findings suggest that neural networks are capable of autonomously discovering secure communication strategies without relying on traditional algorithmic constructions. This work highlights a novel direction in neural cryptography, bridging machine learning techniques with symmetric-key encryption principles and opening new possibilities for adaptive, robust, and hybrid encryption frameworks (3; 19).

REFERENCES

[1] C. E. Shannon, "Communication Theory of Secrecy Systems," Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[3] M. Abadi and D. G. Andersen, "Learning to Protect Communications with Adversarial Neural Cryptography," in Proc. International Conference on Learning Representations (ICLR), 2017.

[4] I. Goodfellow et al., "Generative Adversarial Nets," in Advances in Neural Information Processing Systems (NeurIPS), 2014.

[5] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density Estimation using Real NVP," in Proc. International Conference on Learning Representations (ICLR), 2017.

[6] D. P. Kingma and P. Dhariwal, "Glow: Generative Flow with Invertible 1×1 Convolutions," in Advances in Neural Information Processing Systems (NeurIPS), 2018.

[7] D.J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," in International Conference on Machine Learning (ICML), 2015.

[8] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing Flows for Probabilistic Modeling and Inference," Journal of Machine Learning Research, vol. 22, pp. 1–64, 2021.

[9] A. Kerckhoffs, "La cryptographie militaire," Journal des sciences militaires, vol. 9, pp. 5–83, 1883.

[10] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems (NeurIPS), 2019.

[11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. International Conference on Learning Representations (ICLR), 2015.

[12] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[13] T. M. Cover and J. A. Thomas, Elements of Information Theory, Wiley, 2006.

[14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," in Advances in Neural Information Process ing Systems (NeurIPS), 2017.

[15] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International Conference on Machine Learning (ICML), 2013.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," arXiv preprint 690 arXiv:1607.06450, 2016.

[17] J. Daemen and V. Rijmen, The Design of Rijndael: AES– The Advanced Encryption Standard, Springer, 2002.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[19] D. Grangier and M. A. Brunner, "Neural Stream Ciphers: Continuous Representations for Learned Encryption," in Advances in Neural Information Processing Systems (NeurIPS), 2019.

[20] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the 700 Evolution of Imperfect Authentication," Communications of the ACM, vol. 58, no. 7, pp. 78–87, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.