Harnessing Complex-Valued Chaos in Discrete-Time Hopfield Neural Network for Secure Image Encryption

Quanli Deng, Chunhua Wang, Yichuang Sun *Senior Member, IEEE*, Gang Yang

***Abstract*—The secure transmission of images in critical applications like smart healthcare and autonomous driving demands encryption schemes that are both highly secure and efficient. While chaos-based systems are promising, their security is fundamentally limited by the complexity of the underlying chaotic generator. This paper introduces a novel complex-valued discrete-time Hopfield neural network (CVDHNN) to address this challenge. We demonstrate that the CVDHNN exhibits rich hyperchaotic dynamics through various numerical analyses. The network is successfully implemented on an FPGA, verifying its capability for chaotic sequence generation. Leveraging this complex chaos, we design a robust image encryption algorithm that integrates multi-stage confusion and diffusion. Security analysis confirms the cipher's excellence, achieving favorable statistical properties, high key sensitivity, and strong resistance to various attacks.**

***Index Terms*—Complex-valued neural network, Hyperchaos, Image encryption, FPGA.**

I. Introduction

IN the era of IoT and smart systems, applications from medical imaging to autonomous driving generate and transmit vast volumes of image data, making information security paramount. Conventional cryptographic algorithms are often challenged by the distinct characteristics of image data and the need for high-speed encryption [1]–[3]. Chaos theory offers a powerful foundation for designing lightweight yet secure image encryption schemes, due to its inherent ergodicity and sensitivity [4]–[6]. Therefore, developing chaotic systems that exhibit enhanced complexity and are hardware-friendly remains an important objective for the advanced chaos-based encryption.

Most recently, research focus has been shifting from artificially constructed chaotic maps towards harnessing the intrinsic nonlinear dynamics found in biological systems [7]–[9]. In particular, the chaotic phenomena observed in neural activities have attracted considerable attention for generating complex dynamics [10]–[12]. The Hopfield neural network (HNN), rooted in models of neural dynamics, has emerged as a prominent platform for emulating and studying neuro-inspired chaos. For instance, Zhang *et al.* developed a memristive HNN generating amplitude controlled multiscroll attractors, which enlarges the structure of attractor by using memristor [13]. Lin *et al* introduced two memristors into an HNN to construct multi-wing butterfly attractors [14]. While most research focuses on continuous-time HNNs, the discrete-time HNN (DHNN) offers distinct advantages for digital implementation and algorithmic analysis. The research on DHNN has therefore garnered attention, for example, Bao *et al.* integrated a memristor into a discrete HNN, which enhances the multistability behavior of the network [15]. However, existing DHNN models predominantly operate in the real-valued domain, inherently limiting their dynamic complexity.

Dynamical systems in the complex-valued domain are not merely mathematical abstractions, they are fundamental to describing physical phenomena, from the state vector of quantum systems to the propagation of electromagnetic waves. It is a natural progression to extend the study of chaos into the complex-valued domain, a field now gaining considerable research interest. Zhang *et al.* constructed a novel one-dimensional complex-valued discrete chaotic system using a complex multi-valued function [16]. Through theoretical analysis, they demonstrated that the multi-valued nature of complex functions can generate intricate chaotic phenomena. Hua *et al.* designed a coupled two-dimensional complex-valued discrete system and proved its chaotic robustness using Lyapunov exponent theory. Additionally, they validated the excellent random characteristics of this two-dimensional chaotic system through various numerical simulations [17]. Yao *et al.* constructed an n-dimensional complex-valued discrete system using complex parametric Pascal matrices. Experimental results confirmed that the n-dimensional discrete system in the complex-valued format exhibits stronger chaotic robustness compared to its real-valued domain counterparts [18]. However, a common thread among these works is their focus on artificially designed complex maps. In this work, we introduce the complex-valued discrete HNN (CVDHNN), moving the study of complex-valued chaos from artificially constructed maps to a framework grounded in the dynamics of neural network model.

The evolution in chaotic systems has directly influenced the domain of image encryption. Recent schemes have sought

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported in part by the National Natural Science Foundation of China under Grants 62571183 and 62271197.

Quanli Deng, Chunhua Wang, and Gang Yang are with the College of Information Science and Engineering, Hunan University, Changsha, 410082, China (Corresponding author: Chunhua Wang, e-mail: wch1227164@hnu.edu.cn).

Yichuang Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, U.K.

to leverage increasingly sophisticated dynamics to enhance security [19], [20]. For instance, Yang *et al.* enhanced the hidden chaotic dynamics by enlarging the number of stable node-focus equilibria and developed a hidden multi-wing attractor-based encryption scheme with enhanced security performance [21]. Guang *et al.* proposed a multi-scroll conservative chaotic system to enhance the dynamical behavior of chaotic system and developed a multi-scroll conservative chaotic attractor-based encryption [22]. Recent advances in circuits and systems for video techonlogy have witnessed significant progress in chaotic encryption. For instance, Ding *et al.* proposed a privacy-preserving selective-face encryption scheme that couples a tunable 3D-CIMBA chaotic map [23], Teng *et al.* proposed a multi-image encryption algorithm based on a spatiotemporal chaotic system [24], Gong *et al.* designed an image compression encryption scheme using a four-dimensional chaotic system [25], and Li *et al.* developed a video coefficient confusion encryption scheme based on chaotic maps [26]. These studies have made significant contributions to encryption system design. However, they primarily focused on real-valued chaotic systems, leaving two critical gaps: (1) the nonlinear characteristics of complex-valued chaotic systems remain underexplored, and (2) no specialized encryption scheme has been designed to leverage the random properties of complex-valued chaos in the complex domain. In constrast, we address these limitations through a specialized framework that fully exploits the unique dynamics of complex-domain chaos. Therefore, this work offers a fresh perspective on both encryption theory and the research target by introducing a dedicated encryption framework tailored for complex-valued chaotic systems.

The core contributions are summarized as follows. First, we propose the novel complex-valued discrete-time Hopfield neural network (CVDHNN), demonstrating its ability to generate rich chaotic dynamics. Second, we provide a rigorous analysis of its dynamical properties, including Lyapunov exponents and bifurcation analysis, confirming its complex dynamical behavior. Third, we successfully implement the CVDHNN on an FPGA platform, validating its feasibility for generating high-speed, resource-efficient chaotic sequences in hardware. Finally, we design and test a robust image encryption algorithm that leverages the complex chaotic states of the CVDHNN. Extensive experimental results and security analyses confirm that the proposed cipher achieves a high level of security, outperforming several recent chaos-based schemes in terms of resistance to statistical, differential, and brute-force attacks.

This paper is organized as follows. Section II introduces the proposed Complex-Valued Discrete-time Hopfield Neural Network (CVDHNN) and analyzes its stability. Section III investigates the rich hyperchaotic dynamics of the model through numerical simulations. Section IV details the successful FPGA implementation, verifying the system's hardware feasibility. Section V presents the novel image encryption algorithm built upon the CVDHNN and provides a comprehensive security evaluation. Finally, Section VI concludes the paper and discusses potential future research directions.

II. Complex-valued discrete HNN

A. Model description

The Hopfield neural network (HNN) is a prominent model for simulating complex brain dynamics. Its discrete-time form can be derived from the continuous model via Euler's difference method. This work introduces a discrete-time, complex-valued HNN with two neurons, formulated as:

$$\begin{cases} z_1(n+1) = \mu z_1(n) + w_{11} f(z_1(n)) + w_{12} f(z_2(n)) \\ z_2(n+1) = \mu z_2(n) + w_{21} f(z_1(n)) + w_{22} f(z_2(n)) \end{cases} \tag{1}$$

where $z_i, w_{ij} \in \mathbb{C}$ denote the complex-valued state variable of the i-th neuron and the complex connection strength from the neuron j to neuron i, respectively. The parameter $\mu \in (0,1)$ is a real-valued decay factor. The activation function $f(\cdot)$ governs the neuron's output. While real-valued HNNs typically use functions like the hyperbolic tangent or sigmoid, the transition to the complex domain provides a broader design space for activation functions due to the incorporation of amplitude and phase information. Consequently, we adopt a split complex sinusoidal function defined as:

$$f(z) = \sin(x) + i \cdot \sin(y) \tag{2}$$

where x and y are the real and imaginary parts of z, respectively.

To facilitate the study of complex dynamical behaviors, we fixed the connection weights in the second neuron and selected the weights in the first neuron as variables. The connection weight matrix is defined as

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} a & b \\ -1 & 0 \end{bmatrix} \tag{3}$$

where $a = a_r + ia_i$ and $b = b_r + ib_i$ are two complex-valued controlled parameters. The fixed values were obtained through repeated trial-and-error adjustments. Substituting (3) into (1), the resulting CVDHNN is governed by

$$\begin{cases} z_1(n+1) = \mu z_1(n) + a f(z_1(n)) + b f(z_2(n)) \\ z_2(n+1) = \mu z_2(n) - f(z_1(n)) \end{cases} \tag{4}$$

By decomposing each complex variable into its real and imaginary components via $z = x + iy$, this bi-neuron complex-valued system can be mapped into an equivalent four-dimensional real-valued system.

B. Fixed points calculation and stability analysis

A fixed point of a discrete system is defined as the point in the domain that remains unchanged under the system's mapping. For the CVDHNN model, the fixed points are determined by solving the system:

$$\begin{cases} Z_1 = \mu Z_1 + a f(Z_1) + b f(Z_2) \\ Z_2 = \mu Z_2 - f(Z_1) \end{cases} \tag{5}$$

From this system, the variable Z_2 can be expressed in terms of Z_1 as:

$$Z_2 = \frac{1}{\mu - 1} f(Z_1) \tag{6}$$

Substituting (6) into the first equation of (5) yields a single equation for the fixed point:

$$(\mu-1)Z_1 + af(Z_1) + bf(\frac{1}{\mu-1}f(Z_1)) = 0 \tag{7}$$

Obtaining an analytical solution for (7) is challenging due to its nonlinear form, which intricately couples the real and imaginary parts of the complex variable. To address this, we decompose the complex equation into its real and imaginary components using standard complex arithmetic. This yields the two-dimensional real-valued system:

$$\begin{cases} F_1(x,y) = (\mu-1)x + a_r\sin(x) - a_i\sin(y) \\ \qquad + b_r\sin(\frac{\sin(x)}{\mu-1}) - b_i\sin(\frac{\sin(y)}{\mu-1}) = 0 \\ F_2(x,y) = (\mu-1)y + a_i\sin(x) + a_r\sin(y) \\ \qquad + b_i\sin(\frac{\sin(x)}{\mu-1}) + b_r\sin(\frac{\sin(y)}{\mu-1}) = 0 \end{cases} \tag{8}$$

where x and y are the real and imaginary parts of Z_1, respectively.

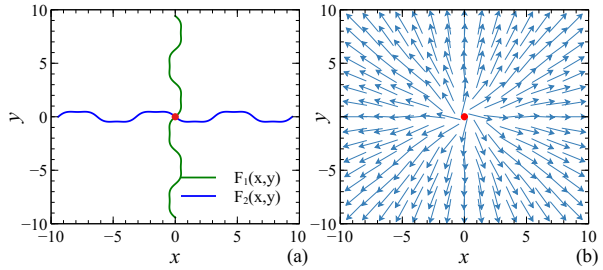

Fig. 1. Solution for the fixed point (a) zero-level contours (b) vector field of the solved point.

To determine the solution of (8), we employ a numerical approach with the parameter set to μ=0.1, a=-2+i and b=-2+1.5i. Fig. 1(a) displays the zero-level contours of the component equations $F_1(x,y)$=0 (green curve) and $F_2(x,y)$=0 (blue curve). The intersection point of these contours, marked by a red circle, corresponds to the system's fixed point [27]. Under this parameter set, the system possesses a single fixed point located at the origin. The local stability is analyzed through the vector field and the system's Jacobian. The vector field visualization in Fig. 1(b) shows all vectors pointing outward from the fixed point, suggesting its instability. For a formal analysis, the Jacobian matrix $\mathcal{J} \in \mathbb{R}^{4\times4}$ is derived by computing the partial derivatives of the four-dimensional real-valued system. The resulting matrix is

$$\mathcal{J} = \begin{bmatrix} \mu + a_r\cos(x_1) & -a_i\cos(y_1) & b_r\cos(x_2) & -b_i\cos(y_2) \\ a_i\cos(x_1) & \mu + a_r\cos(y) & b_i\cos(x_2) & b_r\cos(y_2) \\ -\cos(x_1) & 0 & \mu & 0 \\ 0 & -\cos(y_1) & 0 & \mu \end{bmatrix} \tag{9}$$

The eigenvalues of $\mathcal{J}$ at this fixed point are numerically computed as $\lambda_{1,2}$=0.898$\pm$0.195i and $\lambda_{3,4}$=-2.698$\pm$1.195i. According to the stability criterion for discrete-time systems, a fixed point is stable only if all eigenvalues lie within the unit circle in the complex plane. Since the magnitude of $\lambda_{3,4}$ exceed unity, we can conclude that this fixed point of the system (4) is unstable.

III. Numerical Simulation and Dynamical Analysis

A. Chaotic attractor under fixed parameters

The analysis in Section II-B confirms the existence of an unstable fixed point for the parameter set, (μ=0.1, a=-2+i, b=-2+1.5i), indicating a potential for complex dynamics. To verify this and investigate the resulting behaviors, we conduct numerical simulations using the same parameters. The initial condition is set to z_0=[0.1+0.1i,0.1+0.1i].

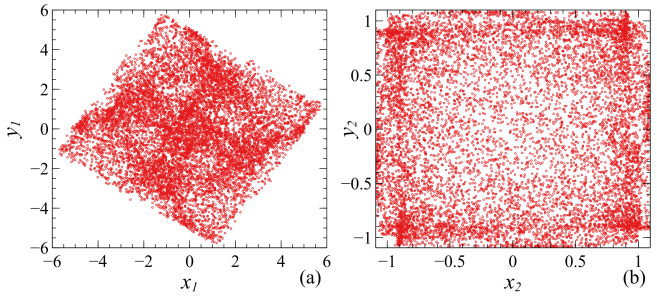

Fig. 2. Phase plot in (a) $x_1 - y_1$ plane, (b) $x_2 - y_2$ plane.

Fig. 2 shows the simulation results in phase space, where the real and imaginary parts of each neuron's state are denoted as x and y, respectively. The trajectories are bounded within a finite region and evolve aperiodically, suggesting a chaotic attractor. To substantiate its chaotic nature, we compute the Lyapunov exponents (LEs) using the standard QR decomposition method (Algorithm 1). This method iteratively evolves and re-orthonormalizes a set of orthogonal tangent vectors via the system's Jacobian. The computed LEs are λ_1=0.412, λ_2=0.236, λ_3=-0.131, and λ_4=-0.404. The presence of two positive Lyapunov exponents confirms that the observed attractor is hyperchaotic.

Algorithm 1 Computing Lyapunov Spectrum via QR Method

Require: Dynamics $\mathbf{f}$, Jacobian $D\mathbf{f}$, initial state $\mathbf{x}_0$, steps T, dimension n

Ensure: Lyapunov exponents $\lambda_1, \lambda_2, \ldots, \lambda_n$

$\mathbf{x} \leftarrow \mathbf{x}_0$, $\mathbf{Q} \leftarrow I_n$, $\mathbf{S} \leftarrow \mathbf{0}_n$
for $t = 1$ to T **do**
  $\mathbf{J} \leftarrow D\mathbf{f}(\mathbf{x})$
  $\mathbf{V} \leftarrow \mathbf{JQ}$
  $[\mathbf{Q}, \mathbf{R}] \leftarrow \mathrm{QR}(\mathbf{V})$
  $\mathbf{x} \leftarrow \mathbf{f}(\mathbf{x})$
  $\mathbf{S} \leftarrow \mathbf{S} + \ln(\mathrm{diag}(\mathbf{R}))$
end for
$\lambda_i \leftarrow S_i/T$ for $i = 1, \ldots, n$
return $\lambda_1, \lambda_2, \ldots, \lambda_n$

B. Dynamical variation with parameters

The analysis in Section III-A confirms the existence of a hyperchaotic attractor for a specific parameter set. To comprehensively identify transitions between different dynamical regimes, we investigate the system's evolution with its parameters. This exploration employs several numerical methods,

including Lyapunov exponents (LE) spectra, bifurcation diagrams, and largest LE (LLE)-based two-parameter dynamical maps. The LEs are computed using the QR decomposition method outlined previously, which offers both rapid and accurate numerical evaluation. The bifurcation diagram is constructed by plotting the local maxima of x_1, the real part of state z_1.

The initial condition is held constant at z_0, consistent with previous analysis. Fig. 3 depicts the transition in dynamical behavior with respect to the complex-valued parameter a, while $b = -2 + 1.5i$ is held constant. For varying parameter $a_r \in [-20, 20]$ with $a_i = 1$, the first two LEs in Fig. 3(a2) are approximately symmetric about the vertical axis. Notably, within the interval [-0.5,0.5], the LLE is less than zero, indicating non-chaotic dynamics. Outside this interval, both of the first two LEs become positive (excluding several periodic windows), signifying the emergence of hyperchaos. Furthermore, the values of these LEs continuously increase with the growing absolute of a_r, suggesting a strengthening of chaotic characteristics. The bifurcation diagram in Fig. 3(a1) consistently corresponds to these features, clearly revealing the periodic windows and non-chaotic parameter region. Similarly, with the a_r fixed at -2, an alteration in dynamical behavior is observed when varying a_i across the interval [-20,20], as illustrated in Fig. 3(b). The corresponding LEs in Fig. 3(b2) reveal that the LLE remains non-positive within the approximate range of [-1.2,0.6], indicating the absence of chaos. Beyond this interval, the system exhibits two positive LEs whose values increase with the growing absolute value of a_i, confirming a transition to hyperchaos. A distinct periodic window is identified within [-9.9,-9.6] in the left half-plane, whereas no such window exists in the right half-plane. The bifurcation diagram in Fig. 3(b1) consistently reflects the dynamics suggested by the LE changes. Specifically, the diagram displays characteristic bifurcation pattern of periodic behavior within the [-1.2,0.6], while the chaotic region in the left half-plane is interrupted by the aforementioned periodic window.

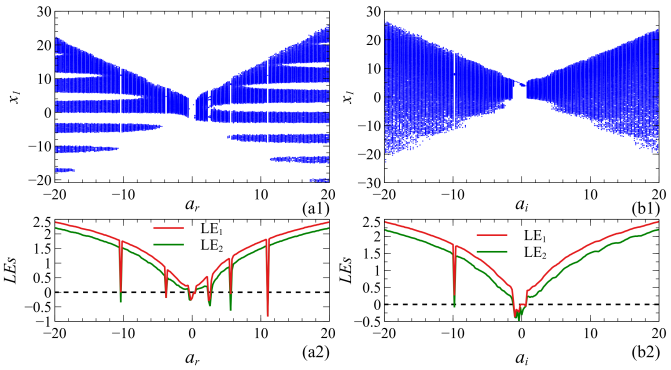

Fig. 3. Dynamics variation with parameter a for fixed $b = -2 + 1.5i$ (a) $a_r \in [-20, 20]$, $a_i = 1$, (b) $a_i \in [-20, 20]$, $a_r = -2$

The influence of parameter b on the system's dynamics is illustrated in Fig. 4. It can be observed that the effect of b shares both similarities and differences with that of parameter a. The influence of b_r is shown in Fig. 4(a2), within the relatively small interval of [-4,3.5], the positive LEs exhibit fluctuations, indicating transitions between hyperchaos and periodic behavior in this range. Outside this interval, as the absolute value of the parameter increases, the values of the two positive LEs continuously grow. Additionally, a relatively wide periodic window is present in the right half-plane. The bifurcation diagram in Fig. 4(a1) clearly reflects this characteristic, showing alternating transitions between chaotic and periodic states. Similarly, Fig. 4(b) displays the changes in system dynamics as parameter b_i varies. Unlike the case for b_r, only two periodic windows are observed, located in [-2,-0.95] and [0.15,0.5], while two positive LEs exit across the remaining intervals.

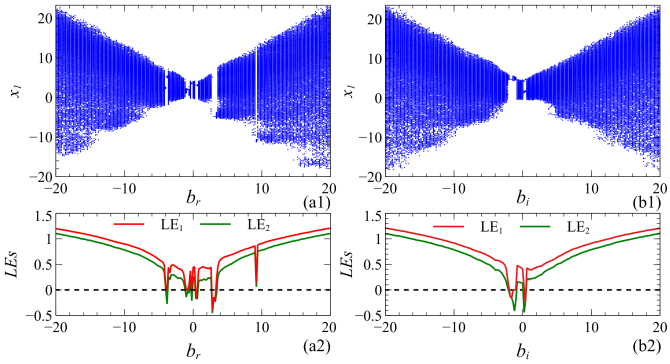

Fig. 4. Dynamics variation with parameter b for fixed $a = -2 + 1i$ (a) $b_r \in [-20, 20]$, $b_i = 1.5$, (b) $b_i \in [-20, 20]$, $b_r = -2$

In summary, the systematic investigation of parameter variations in Figs.3 and 4 reveals the following key characteristics of the system's dynamical transitions. First, for both parameters a and b, increasing the absolute value of the varied real and imaginary components generally lead to enhanced hyperchaotic behavior, as evidenced by the growth in the two LLE. Second, regular dynamics tend to occur within limited parameter intervals, often located near the origin or symmetric regions, while hyperchaos dominates outside these ranges. Third, periodic windows are observed in certain parameter intervals, interrupting the otherwise continuous chaotic regimes, with their presence and distribution differing between parameters a and b. These transitions underscore the sensitivity of the system's dynamics to parameter changes.

IV. FPGA-BASED IMPLEMENTATION

To validate the physical realizability and performance of our proposed system under real-world constraints, a hardware implementation is needed. Among various hardware platforms, FPGAs offer a unique balance of parallel processing capability, design flexibility, and rapid prototyping, making them suitable for realizing the complex, concurrent computations inherent in chaotic systems [28]–[32]. In this work, the experimental setup is built on a Xilinx Zynq-7000 FPGA (device xc7z020clg400-1), which serves as the core computational unit. The system is clocked at 50 MHz to ensure stable and synchronous operations. To visualize the chaotic attractors, the digital outputs from the FPGA are converted into analog signals using a AD9767 dual channel 14-bit DAC. This high-resolution DAC

accurately renders the fine, continuous structure of the chaotic phase portraits on an oscilloscope.

To facilitate a practical hardware implementation on a digital platform, this complex-valued model is explicitly decomposed into its real and imaginary components, resulting in a set of four coupled real-valued equations that govern the dynamics of the state variables x_1, y_1, x_2 and y_2. The hardware architecture is designed to implement these decomposed equations in a practical and efficient manner, aiming to meet the constraints of high throughput and low resource utilization on the target FPGA. This is achieved through the extensive use of fixed-point arithmetic, where a Q5.27 (5 integer bits including sign, and 27 fractional bits) is carefully selected to provide an optimal balance between dynamic range, numerical precision, and hardware resource consumption.

To efficiently compute the sine function without resorting to resource-intensive floating-point units, a hardware-optimized CORDIC (Coordinate Rotation Digital Computer) algorithm is employed. The CORDIC algorithm iteratively rotates a vector to approximate the sine and cosine values, utilizing only shift-and-add operations, which aligns perfectly with the resource-efficient design philosophy of this work. The algorithm operates in rotation mode. Given an input angle θ, it initializes a vector $(x_0, y_0) = (K, 0)$, where $K \approx 0.60725$ is a constant scaling factor. Over a series of N rotations, the vector is rotated by a predefined set of decreasing angles $\alpha_i = \arctan(2^{-i})$. At each iteration, the rotation direction d_i is determined by the sign of the remaining angle z_i, and the vector is updated according to the following equations

$$\begin{aligned} x_{i+1} &= x_i - d_i \cdot (y_i \gg i) \\ y_{i+1} &= y_i + d_i \cdot (x_i \gg i) \\ z_{i+1} &= z_i - d_i \alpha_i \end{aligned} \tag{10}$$

where $\gg i$ denotes an arithmetic right shift by i bits. After N iterations, the output $\sin(\theta)$ is approximated by the final y_N component.

The top level architecture, depicted in Fig. 5, is designed as a synchronous digital circuit operating on the fixed-point system. The state transition diagram of the top level controller is presented in Fig. 5(a), while the corresponding register-transfer level (RTL) schematic of the architecture is shown in Fig. 5(b). The core of the design consists of four main state registers that hold the current system state. The computation of the next state follows a highly parallel data path, enabled by the modular structure of the design. Upon each active clock edge when the enable signal is asserted, the current state from the registers is fed concurrently into two primary computational blocks. First, a set of four parallel sine calculation modules compute the nonlinear terms. These terms are then routed, along with the current state values, to the second set of blocks, the next state computation logic. This logic comprises four dedicated units that implement the decomposed iteration equations. The results of these computations are fed into a bank of multiplexers at the input of the state registers. A simple control logic block governs this data selection. It routes the newly computed values to the registers when the system is enabled and not in reset. This streamlined architecture ensures that a single, full

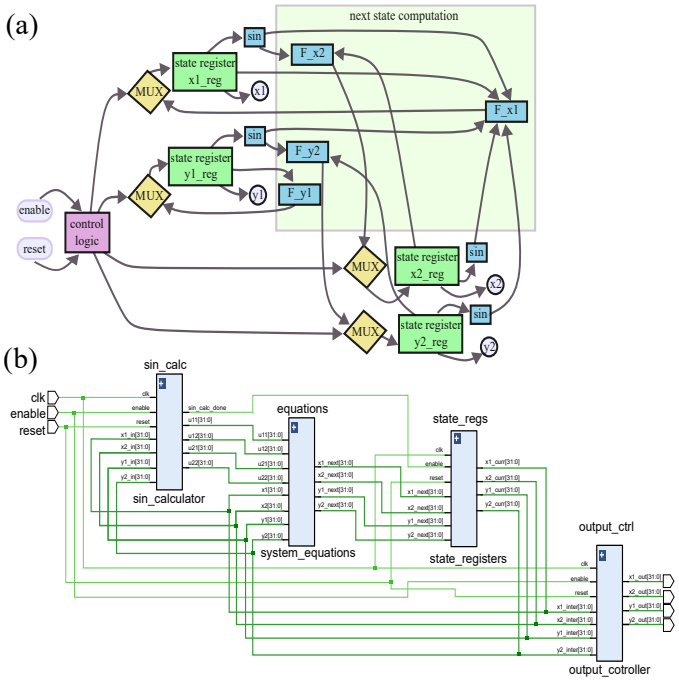

Fig. 5. Scheme of FPGA-based implementation for the CVDHNN: (a) state transition diagram, (b) RTL diagram.

iteration of the complex chaotic neural system is completed within one clock cycle.

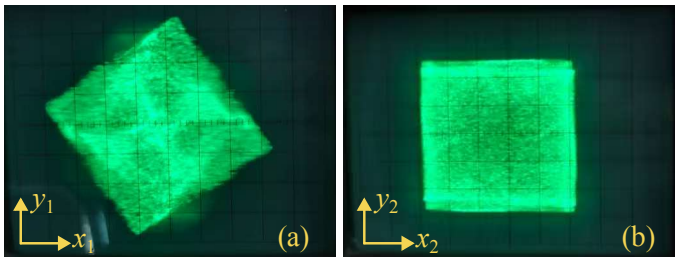

Fig. 6. FPGA-based implementation of chaotic attractor in (a) $x_1 - y_1$ plane, (b) $x_2 - y_2$ plane.

The experimental results presented in Fig. 6 validate the successful hardware implementation of the proposed chaotic system. The phase portraits, captured directly from the oscilloscope, exhibit the complex and non-repeating trajectories characteristic of chaotic dynamics. A visual comparison with the numerical simulations in Fig. 2 shows a remarkable qualitative agreement between the hardware output and the software model; the characteristic structure and geometry of the attractors are accurately preserved. This close correspondence demonstrates that the employed fixed-point arithmetic and the associated hardware-optimized approximations have maintained the essential nonlinear behavior of the system without introducing significant distortion.

The trade-off between numerical accuracy and hardware resource utilization is illustrated in Fig. 7, where four precision configurations are evaluated with respect to numerical accuracy, as measured by root mean square error (RMSE), and hardware efficiency. Tables I and II summarize the per-

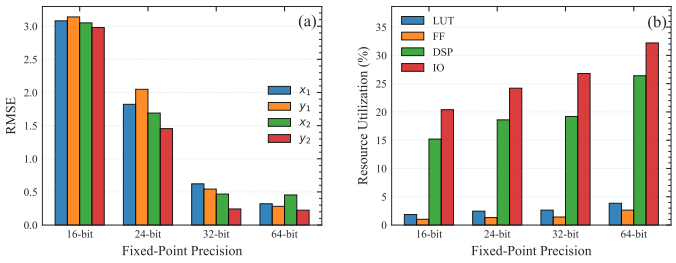

Fig. 7. Comparative analysis of FPGA implementation metrics: (a) numerical accuracy measured by RMSE, (b) hardware resource utilization for each precision configuration.

TABLE I
RMSE COMPARISON FOR DIFFERENT FIXED-POINT PRECISIONS

Precision	x_1	y_1	x_2	y_2
16-bit	3.08156	3.14058	3.05086	2.98061
24-bit	1.82186	2.04831	1.68990	1.45524
32-bit	0.62181	0.54305	0.46756	0.24291
64-bit	0.32115	0.28249	0.45404	0.22426

formance of each setting. As can be observed, the 32-bit design achieves a favorable balance, providing high numerical accuracy with acceptable RMSE values for all state variables while maintaining moderate hardware resource utilization. Specifically, the 32-bit configuration achieves RMSE below 0.62 for all variables, representing a 65% improvement over 24-bit precision. Notably, transitioning to 64-bit precision yields only marginal accuracy gains. Furthermore, the 32-bit implementation maintains resource usage below 20% for critical DSP blocks, whereas the 64-bit precision demands more DSP resources for diminishing accuracy returns. The analysis confirms that the selected bit-width precision represents the optimal trade-off point between numerical fidelity and hardware efficiency.

The impact of finite-precision arithmetic on chaotic dynamics was investigated through systematic cycle detection tests. Seven fixed-point formats with varying data width (ranging from 12 to 64 bits) were evaluated using Brent's algorithm to detect periodic behavior. Each format was tested with 1000 random initial states across data volumes from 1GB to 2TB (corresponding to approximately 134 million iterations). Table III summarizes the key findings. The percentages in the table indicate the proportion of sequences that exhibited periodic behavior out of 1000 random tests; the accompanying values denote the average cycle length (with undetected cycles marked as "NAN"). It is evident that lower-precision implementations (12-24 bits) consistently show periodic behavior with relatively short cycle lengths, confirming the degradation of chaos under insufficient numerical precision. In contrast, implementations with 32-bit or 64-bit data widths maintained chaotic characteristics throughout all tests, with no cycles detected even at the maximum test volume. These results provide quantitative evidence that, with a 32-bit data width, generating up to 2 TB of pseudorandom numbers does not lead to periodic behavior, which is sufficient to ensure its applicability in cryptographic systems.

TABLE II
FPGA RESOURCE UTILIZATION (%) FOR DIFFERENT FIXED-POINT PRECISIONS

Precision	**LUT**	**FF**	**DSP**	**IO**
16-bit	1.86	1.02	15.2	20.4
24-bit	2.46	1.32	18.6	24.2
32-bit	2.64	1.42	19.2	26.8
64-bit	3.85	2.64	26.4	32.2

To further validate the cryptographic suitability of the hardware-generated chaotic sequence, we performed the NIST SP 800-22 statistical randomness test on the output bitstream. The test employed a sample length of 10^6 bits, with the significance level set at α=0.01. As shown in Table IV, the sequence passed all 15 test items, with P-values significantly exceeding the 0.01 threshold. We also employed the more stringent open-source test suit TestU01. Our test uses five batteries, including *Rabbit*, *Alphabit*, *BlockAplhabit*, *SmallCrush* and *Crush*. Each battery applies multiple statistical tests to the sequence and produces a P-value. A sequence is considered to pass a battery if all resulting P-values are greater than 0.001. The results, summarized in Table V, confirm that the generated pseudorandom numbers pass all batteries. These randomness tests indicate that, despite being implemented using fixed-point format, the generated chaotic sequence exhibits no statistically significant deviation from a true random sequence, fully satisfying the requirements for pseudorandom number generators in cryptographic applications.

V. IMAGE ENCRYPTION BASED ON THE CVDHNN

The dynamics analysis and successful FPGA implementation presented in the above have established the CVDHNN as a robust generator of highly random sequences. To translate these advantages into tangible information security application, we propose a novel image encryption scheme that fully leverages the inherent strengths of the CVDHNN. The core of the proposed algorithm lies in utilizing the high-dimensional, pseudo-random sequence generated by the complex-valued chaotic system to drive a multi-stage encryption process, which integrates confusion, diffusion and a unique geometric transformation in the complex plane.

A. Proposed image encryption scheme

The overall procedure, depicted in Fig. 8, follows a multi-stage structure to thoroughly scramble both the spatial correlation and statistical distribution of the plaintext image. The process begins with generating a sufficient long complex-valued chaotic sequence $\mathbf{S_c}$ from a user-provided personal key K_p. This sequence serves as the foundational source of randomness for the subsequent stages.

First, a pixel-level confusion is performed, where the real part of $\mathbf{S_c}$ guides a non-sequential pairing and rearrangement of image pixels, effectively breaking the spatial correlations. Subsequently, a value-level diffusion is applied via a bitwise XOR operation between the paired pixels and quantized version of the chaotic sequence, altering the pixel value globally. To further enhance security, the intermediate image is transformed into the polar coordinate system. The magnitude (R)

TABLE III
CYCLE DETECTION RESULTS FOR DIFFERENT FIXED-POINT FORMATS

Data width	Data length							
	1GB	8GB	32GB	64GB	256GB	512GB	1TB	2TB
12-bit	100%/6491	100%/6491	100%/1176	100%/6491	100%/6491	100%/1176	100%/6491	100%/6491
14-bit	100%/9686	100%/9657	100%/9762	100%/9622	100%/9686	100%/9657	100%/9628	100%/9785
16-bit	100%/162459	100%/153302	100%/134841	100%/143999	100%/116381	100%/143999	100%/143999	100%/143999
24-bit	0%/NAN	0%/NAN	87.5%/845714490	100%/845714490	100%/845714490	100%/845714490	100%/845714490	100%/845714490
32-bit	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN
64-bit	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN	0%/NAN

TABLE IV
THE NIST TEST ANALYSIS REPORT OF CHAOTIC SEQUENCE

sub-test	p-value				
	x1	y2	x2	y2	z
Frequency	0.6579	0.5192	0.6948	0.4685	0.3242
Block frequency	0.4019	0.4871	0.5676	0.6545	0.5226
Cum. sums	0.8676	0.2662	0.4372	0.4896	0.2496
Runs	0.7791	0.2896	0.2785	0.6579	0.4946
Longest run	0.1435	0.7399	0.9204	0.9114	0.6993
Rank	0.2026	0.6579	0.3669	0.5341	0.7399
FFT	0.5544	0.2022	0.4741	0.3504	0.2268
Non-Ovla. temp.	0.1372	0.3109	0.6375	0.7192	0.9558
Universal	0.8513	0.7594	0.3669	0.6163	0.5544
Appr. entropy	0.9357	0.4559	0.7981	0.6578	0.1421
Ran. Exc.	0.2757	0.6579	0.6371	0.2022	0.5680
Ran. Exc. Var.	0.8343	0.1815	0.4732	0.3669	0.6024
Serial(1st)	0.5544	0.9988	0.1626	0.8168	0.7339
Serial(2nd)	0.9114	0.9240	0.4749	0.4559	0.4559
Linear complexity	0.4786	0.3184	0.5955	0.2492	0.8813

TABLE V
TESTU01 RESULTS FOR DIFFERENT LENGTHS OF PRNS

TestU01		state variable				
Sub-tests	Data size	x1	y1	x2	y2	z
Rabbit	32GB	40/40	40/40	40/40	40/40	40/40
Alphabit	32GB	17/17	17/17	17/17	17/17	17/17
BlockAlphabit	32GB	102/102	102/102	102/102	102/102	102/102
SmallCrush	6GB	15/15	15/15	15/15	15/15	15/15
Crush	1TB	144/144	144/144	144/144	144/144	144/144

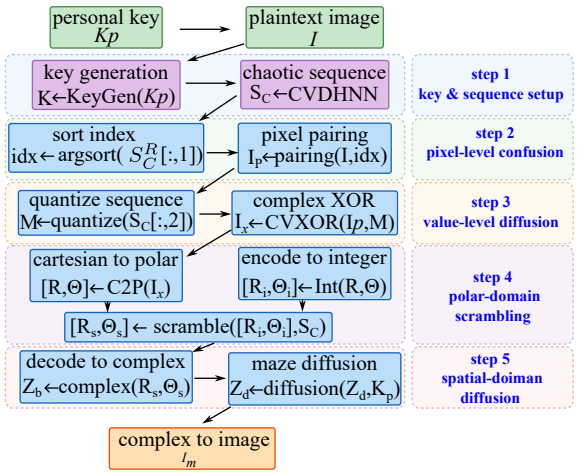

Fig. 8. Overall Image Encryption Process

and phase (Θ) components are then independently scrambled using a digit-decomposition-based permutation, which is also chaotically controlled. The scrambled components are reconstructed into a complex array, upon which a unique spatial diffusion, termed Complex Maze Diffusion, is applied. This stage propagates local change across the entire image through a chain reaction of complex rotations along a chaotic path.

To enhance the key's effectiveness and ensure it is properly formatted for the chaotic system, a cryptographic hash function and a compression step are employed to the user-defined 256-bit personal key K_p. The personal key is firstly processed using a SHA-256 hash function to obtain a 256-bit value H. Then, this hash value H is divided into 8 segments of 32 bits. The segments are compressed into 8 initial key values $K_1, K_2, \ldots K_8$ through a bitwise XOR operation with each segment

$$K_j = H_{4(j-1)+1} \oplus H_{4(j-1)+2} \oplus H_{4(j-1)+3} \oplus H_{4(j-1)+4}. \quad (11)$$

The eight K_j values are subsequently mapped to the system parameters and initial conditions through a linear scaling transformation:

$$P_j = L_j + K_j \cdot (U_j - L_j), \quad j = 1, 2, \ldots, 8 \quad (12)$$

where P_j represents the j-th parameter, and $[L_j, U_j]$ specifies its allowable range. Based on the chaotic region analysis presented in Fig. 3 and Fig. 4, the following ranges are selected to maximize chaotic behavior, avoid periodic windows, and maintain robustness under finite-precision implementation: $a_r, b_r \in [-20, -10]$, $a_i, b_i \in [5, 20]$ and $z_0 \in [-20, 20]$. As illustrated in Fig. 9(a), the effective key space is evaluated through three critical stages. Beginning with 256-bit user input, the mapping to CVDHNN parameters preserves the full entropy in double-precision floating-point representation. Statistical verification of over one million randomly generated keys confirms that 100% of the mapped parameters reside within the chaotic region, yielding an effective key space of 2^{256} bits. When using Q5.27 fixed-point hardware implementation format, each parameter is quantized to 2^{27} discrete levels, establishing a theoretical upper bound of 2^{216} distinct parameter combinations. Crucially, our analysis demonstrates that quantization preserves chaotic dynamics in all tested cases, resulting in a retained key space of 2^{216} bits. Although reduced from the floating-point ideal, this remains cryptographically sufficient, exceeding the 2^{100}-bit security threshold to resist brute-force attacks. The parameter sensitivity, defined as $S = 1 - |\rho|$ where ρ is the correlation coefficient between original and 1-LSB perturbed sequences, is presented in Fig. 9(b). All eight parameters

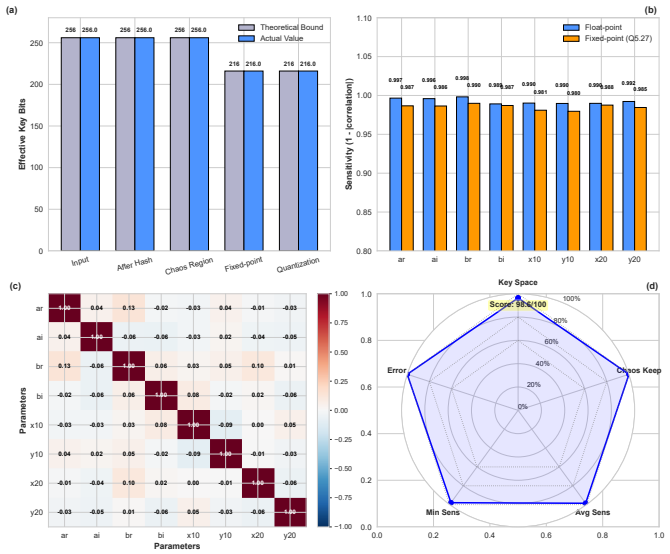

Fig. 9. The comprehensive security analysis for (a) effective key space reduction, (b) parameter sensitivity, (c) parameter correlations, (d) fixed-point implementation performance.

exhibit excellent sensitivity ($S > 0.98$), confirming avalanche properties. Fig. 9(c) shows low inter-parameter correlation coefficients, indicating efficient entropy distribution across the 8-dimensional parameter space. Fig. 9(d) illustrates that the fixed-point implementation maintains over 98% performance across all evaluated metrics, demonstrating practical feasibility without security degradation.

With the system initialized, a sufficiently long complex chaotic sequence $\mathbf{S_c}$ is generated. To eliminate transient effects, the first N_{cut} iterations are discarded.

Following the generation of the chaotic sequence, the first confusion operation is applied to disrupt the strong spatial correlations inherent in nature images. The process operates on the flattened, one-dimensional vector representation of the plaintext image I of size $m \times n$. Let $\vec{P} = [p_1, p_2, \ldots, p_{m \cdot n}]$ be this vector. The goal is to create a new vector $\vec{P}_{pair}$ where pixels are rearranged in pairs according to a chaotic sorting order. The real part of the first channel of the complex chaotic sequence $\mathbf{S_c}[:, 1]$ is utilized to generate a permutation index vector as

$$idx = \text{argsort}(\Re(\mathbf{S_c}[:, 1])) \tag{13}$$

where argsort$(\cdot)$ returns the indices that sort the input sequence in ascending order. Pixels are then paired sequentially according to this order.

Subsequent to the spatial confusion, the pixel-paired image I_{pair} undergoes its first value-altering diffusion. To achieve globally modifying the pixel values, a bitwise complex XOR operation is employed, which leverages the randomness of the complex chaotic sequence to perform a mixing at the bit level. The second channel of the complex chaotic sequences, $\mathbf{S_c}[:, 2]$ is utilized to generate a diffusion sequence $\vec{M}$. This is accomplished by quantizing the chaotic values into the integer range [0,255], suitable for pixel-level operations. The intermediate image I_{pair} is treated as a vector of complex numbers, $\vec{V}$, where the real and imaginary parts are the pixels. For each element $v_k^r + jv_k^i$ in $\vec{V}$ and $m_k^r + jm_k^i$ in $\vec{M}$, the operation is defined as

$$v_k^{xor} = (v_k^r \oplus m_k^r) + j(v_k^i \oplus m_k^i) \tag{14}$$

where $\oplus$ denotes the bitwise XOR operation. The result is a complex vector $\vec{V}_{xor}$ where both the real and imaginary components have been altered.

Algorithm 2 Polar Domain Scrambling via Digit Decomposition

Require: Complex vector V, Chaotic matrix $S \in \mathbb{C}^{L \times 4}$
Ensure: Scrambled R, Θ

$L \leftarrow |V|$, $R_{max} \leftarrow 255\sqrt{2}$
for $k \leftarrow 1$ **to** L **do**
  $r \leftarrow |V[k]|$, $\theta \leftarrow \text{angle}(V[k])$
  $R_{int}[k] \leftarrow \lfloor \frac{r}{R_{max}} \cdot 65535 \rfloor$
  $\Theta_{int}[k] \leftarrow \lfloor \frac{\theta + \pi}{2\pi} \cdot 65535 \rfloor$
end for
$R_{dig} \leftarrow \text{Split16to4}(R_{int})$, $\Theta_{dig} \leftarrow \text{Split16to4}(\Theta_{int})$
$R_{perm} \leftarrow \text{Permute}(R_{dig}, \text{argsort}(\Re(S)))$
$\Theta_{dig}[:, 2:4] \leftarrow \text{Permute}(\Theta_{dig}[:, 2:4], \text{argsort}(\Im(S)))$
$\Theta_{perm} \leftarrow \text{Combine4to16}(\Theta_{dig})$

return R_{perm}, Θ_{perm}

After the initial diffusion, the intermediate complex-valued image $\vec{V}_{xor}$ possesses altered pixel values but may still retain some statistical patterns in its complex components. To introduce a higher level of nonlinearity, the data is transformed from the Cartesian coordinate system to the Polar coordinate system. This transformation, followed by a digit-level scrambling, constitutes a confusion-diffusion hybrid stage as depicted in Algorithm 2. The process begins by converting each complex number to its polar form, denoted as R_k and Θ_k. To facilitate digital manipulation, the continuous values of R_k and Θ_k are encoded into 16-bit unsigned integers. This encoding maps the phase and magnitude into an integer space, making them amenable to bit-level and digit-level operations. In the digit decomposition scrambling step, each 16-bit integer $R_{int}[k]$ or $\Theta_{int}[k]$ can be considered a 4-digit number in a base-16 numeral system. Separate chaotic sequences, derived from the remaining parts of $\mathbf{S_c}$, are used to generate permutation vector P_R and P_Θ for the magnitude and phase, respectively. The scrambling is then applied digit by digit. For the magnitude R_{int}, all four digits are permuted. For the phase Θ_{int}, only the three less significant digits are permuted, leaving the most significant digit intact to prevent the phase from undergoing a full-circle wrap-around which would render the scrambling ineffective. The scrambled digits are then recombined to form the final scrambled integers R_{perm} and Θ_{perm}.

Following the scrambling in the polar domain, the data is reconstructed into a complex-valued image. To ensure that a modification in a single pixel value propagates, a Complex Maze Diffusion mechanism is employed as formalized in Algorithm 3. This stage performs a chain reaction of complex rotations in a local neighborhood, guided by a chaotic path.

Algorithm 3 Complex Maze Diffusion Algorithm

Require: Scrambled complex vector Z, Personal key K_p, Image dimensions $m \times n$
Ensure: Diffused complex vector Z_{diff}

$L \leftarrow m \times n$
$[K_d, \mathcal{O}] \leftarrow \text{GenerateDynamicKeys}(Z, K_p)$
$\mathcal{P} \leftarrow \text{argsort}(\Re(K_d[:, 2]))$
$\mathcal{N} \leftarrow \text{PrecomputeNeighbors}(m, n)$
for $t = 1$ **to** L **do**
  $i \leftarrow \mathcal{P}[t]$
  $\Phi \leftarrow \text{RotationField}(Z[i], K_d[t], t)$
  for $nb = 1$ **to** 8 **do**
    $j \leftarrow \mathcal{N}[i, nb]$
    if $j \neq -1$ **then**
      $Z[j] \leftarrow Z[j] \times \Phi[nb]$
    end if
  end for
end for

return Z

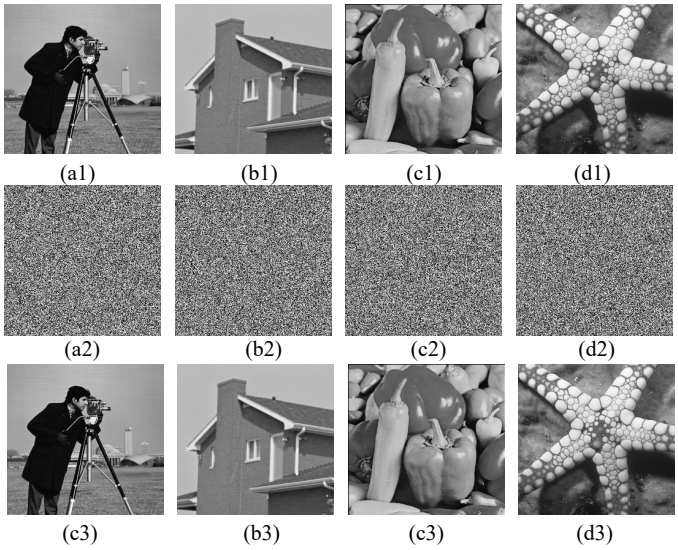

Fig. 10. The plaintext image, encrypted image and decrypted image for (a) Cameraman, (b) House, (c) Peppers, (d) Fishstar.

This mimics a wave of change spreading through a maze. The process begins by decoding the scrambled polar coordinates R_{perm} and Θ_{perm} back into a complex-valued array Z_{before}, which is then prepared for diffusion.

To ensure that the diffusion process is sensitive to the plaintext, a dynamic key derivation mechanism is employed. First, the scrambled complex vector Z is serialized into a binary stream, from which a 256-bit cryptographic hash digest $\mathcal{H}_Z = \text{SHA-256}(Z)$ is computed. Then, the dynamic keys are irreversibly bound to both the specific image content and the secret key K_p by computing a derivation seed $S_d = \text{SHA-256}(\mathcal{H}_Z \| K_p)$, where $\|$ denotes concatenation. The seed S_d is subsequently mapped, using the identical transformation rules defined in (12), to initialize the CVDHNN. Finally, the dynamical key sequences K_d are obtained by iterating the chaotic system using the newly derived parameters. This design inherently mitigates the risks of structural leakage or state recovery. The use of a one-way hash function ensures that Z cannot be inferred from K_d, breaking any direct analytical link. Moreover, since the entire chaotic system is re-initialized with a seed unique to each plaintext iamge, the process is stateless and prevents any inter-image synchronization when encrypting multiple images with the same K_p. The chaotic path $\mathcal{P}$ is generated by sorting a specific chaotic sequence as

$$\mathcal{P} = \text{argsort}(\Re(K_{dynamic}[:, 2])). \tag{15}$$

This path $\mathcal{P}$ defines the order in which pixels are activated in the maze. When a cell Z_i at spatial coordinate (x_i,y_i) is activated, it influences its 8-connected neighbors. For each neighbor Z_j, a complex rotation is applied

$$Z_j^{new} = Z_j^{old} \cdot e^{j\phi_j} \tag{16}$$

The rotation angle θ_j for the j-th neighbor is generated by a local function $\mathcal{F}$ that depends on the current central pixel Z_i, the corresponding dynamic key K_i, and the step index t as

$$\phi_j = \mathcal{F}(\angle Z_i, \angle K_i, t, j) = \angle Z_i + \angle K_i + \alpha \cdot \sin(\beta \cdot t) + \gamma_j \tag{17}$$

where α, β are constant, and γ_j is a base angle offset of the j-th neighbor. The local rotation field ensures that each neighbor is perturbed in a unique and unpredictable direction. As the algorithm follows the chaotic path $\mathcal{P}$, the changes cascade from one activated cell to its neighbors, creating a complex web of dependencies. Following this diffusion, the processed complex-valued vector $Z_{diffuse}$ is converted back into the spatial domain to form the final ciphertext image I_{enc}.

The decryption procedure is the symmetric inverse of the encryption process, executed in reverse order. It begins by applying the inverse of the Complex Maze Diffusion, which involves tracing the same chaotic path $\mathcal{P}$ but applying complex rotations with conjugate angle $-\phi_j$ to neutralize the diffusion effect. Following this, the scrambled polar coordinates are restored by applying the inverse digit decomposition scrambling, after which the data is transformed back from the polar to the Cartesian coordinate system. The inverse bitwise XOR operation is then performed to cancel the initial diffusion. Finally, the inverse chaotic-guided pixel pairing rearranges the pixel back to their original spatial positions. As the chaotic system is deterministic, the same initial key K_p will generate an identical sequence $\mathbf{S_c}$, ensuring that all inverse operations reconstruct the original data. As shown in Fig. 10, four representative images are processed to validate the encryption scheme. The corresponding ciphertexts exhibit a noise-like, visually unrecognizable appearance, whereas the decrypted images perfectly match the originals. These results demonstrate that the encryption effectively conceals visual information while enabling exact reconstruction through the inverse process with the correct key.

B. Formal security analysis

Beyond statistical metrics, a cryptographic evaluation should address resistance to chosen-plaintext attacks (PCA) and chosen-ciphertext attacks (CCA). The proposed scheme

achieves this through a plaintext-dependent randomized diffusion mechanism that breaks deterministic predictability while maintaining invertibility for legitimate users.

Let the encryption function be denoted as $\mathcal{E}_{K_p}(\mathcal{I}) = \mathcal{C}$, where K_p is the 256-bit personal key and $\mathcal{I}$ is the plaintext image. The process can be formally decomposed into two cryptographically coupled stages:

$$\mathcal{E}_{K_p}(\mathcal{I}) = \mathcal{C}_2(K_d(\mathcal{Z}), \mathcal{C}_1(K_p, \mathcal{I})), \tag{18}$$

where $\mathcal{C}_1$ represents the first four states (pixel pairing, XOR diffusion, polar scrambling) driven solely by the key-derived chaotic sequence $\mathcal{S}_c$, and $\mathcal{C}_2$ denotes the complex Maze diffusion stage. The crucial security enhancement resides in the generation of dynamic diffusion keys K_d:

$$K_d = \Psi(\text{SHA-256}(\mathcal{Z} \| K_p)), \quad \mathcal{Z} = \mathcal{C}_1(K_p, \mathcal{I}) \tag{19}$$

This construction ensures that K_d is a cryptographic hash of both the secret key and the intermediate cipher state $\mathcal{Z}$, making the final diffusion stage $\mathcal{C}_2$ plaintext-aware and non-deterministic for a fixed K_p.

1) Resistance to chosen-plaintext attacks: Under the CPA model, adversary $\mathcal{A}$ can query an encryption oracle $\mathcal{O}_{\text{enc}}$ with adaptively chosen plaintexts $\{\mathcal{I}_1, \mathcal{I}_2, \dots\}$ and receives corresponding ciphertexts $\{\mathcal{C}_1, \mathcal{C}_2, \dots\}$. The security relies on the indistinguishability of the tuple $K_d, \mathcal{C}_2$ from a random function ensemble. Since $\Psi(\text{SHA-256}(\mathcal{Z}_i \| K_p))$ and $\mathcal{Z}_i = \mathcal{C}_1(K_p, \mathcal{I}_i)$, each query yields a fresh, statistically independent diffusion key $K_d^{(i)}$, even for identical K_p. Consequently, the adversary cannot establish useful relationship between $(\mathcal{I}_i, \mathcal{C}_i)$ pairs for the following reasons. The intermediate state $\mathcal{Z}_i$ is concealed by the key-dependent permutation $\mathbf{S}_c$ in $\mathcal{C}_1$. And the diffusion operator $\mathcal{C}_2$ is randomized per query through $K_d^{(i)}$. Any differential analysis on $\{\mathcal{I}, \mathcal{I}'\}$ fails due to the difference $\Delta\mathcal{Z} = \mathcal{Z} \oplus \mathcal{Z}'$ propagating through a uniquely randomized rotation field $\Phi^{(i)}$ defined by $K_d^{(i)}$ and the chaotic path $\mathcal{P}^{(i)}$. Formally, for any non-zero difference $\Delta\mathcal{I}$, the probability that an adversary can predict the corresponding ciphertext difference $\Delta\mathcal{C}$ is negligible as

$$\mathcal{P}[A(\Delta\mathcal{I}, \Delta\mathcal{C}) = 1] \leq \frac{1}{2^{\Delta\mathcal{Z}}} + \epsilon(\lambda) \tag{20}$$

where $\epsilon(\lambda)$ is a negligible function in the security parameter λ (determined by the 256-bit key and avalanche effect of SHA-256). Hence, the encryption scheme behaves as a pseudorandom permutation under chosen-plaintext queries.

2) Resistance to chosen-ciphertext attacks: In the CCA model, $\mathcal{A}$ additionally possesses a decryption orcale $\mathcal{O}_{\text{dec}}$, except for the challenge ciphertext. The security stems from the non-malleability of the diffusion process and the state-dependent invertibility of $\mathcal{C}_2$. Decryption requires the exact dynamic key K_d used during encryption as

$$\mathcal{D}_{K_p}(\mathcal{C}) = \mathcal{C}^{-1}(K_p, \mathcal{C}_2^{-1}(K_d, \mathcal{C})) \tag{21}$$

where $\mathcal{C}_2^{-1}$ is the inverse diffusion using the same K_d. Since K_d is derived from $\mathcal{Z}$ which is not directly accessible from $\mathcal{C}$ without K_p, any adversary-submitted ciphertext $\tilde{\mathcal{C}} \notin \mathcal{C}_i$ will be decrypted to a random plaintext $\tilde{\mathcal{I}}$) that bears no relation to any valid plaintext. Formally, for any adversarial crafted $\tilde{\mathcal{C}}$

$$\mathcal{P}[\mathcal{D}_{K_p}(\tilde{\mathcal{C}} \in \mathcal{M}_{\text{valid}})] \leq \frac{1}{2^{128}} \tag{22}$$

where $\mathcal{M}_{\text{valid}}$ denotes the set of structured image plaintexts. This is because the inversion of $\mathcal{C}_2$ without the correct K_d results in random phase perturbations that avalanche through $\mathcal{C}_1^{-1}$. The decryption oracle therefore provides no advantage over the CPA settings, as it returns effectively random responses for malicious queries.

C. Examination of the encryption scheme

Following the delineation of the encryption algorithm, the security of the proposed CVDHNN-based scheme is evaluated through a series of tests. All simulations are conducted within the Matlab 2024a environment, running on a Zorin OS 17.3 platform with an AMD Ryzen 7 5800 CPU. The test images, a selection of 256×256 grayscale images from the Set12 dataset, are encrypted and subsequently analyzed using metrics including pixel histogram distribution, adjacent pixel correlation, key sensitivity, and information entropy. The security and robustness of the proposed encryption system are demonstrated through analysis and discussion of the obtained results.

1) MSE and SSIM analysis: To evaluate the perceptual dissimilarity between the original plaintext image and the encrypted ciphertext image, the mean square error (MSE) and the structural similarity index measure (SSIM) are employed as objective metrics. These measurements quantify the deviation of the encrypted image from the original in terms of pixel-level intensity differences and structural information preservation, respectively. These metrics are defined as:

$$\begin{aligned} \text{MSE} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - I'(i,j)]^2 \\ \text{SSIM} &= \frac{(2\mu_I \mu_{I'} + c_1)(2\sigma_{II'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)} \end{aligned} \tag{23}$$

where M and N denote the image dimensions, I and I' represent the original and encrypted images, respectively, μ, σ^2, and $\sigma_{II'}$ are the mean, variance and covariance with c_1 and c_2 being small constants for numerical stability. For a robust encryption scheme, the MSE should be significantly high and the SSIM should approach zero, indicating that the encryption process has destroyed the structural information of the original image. The experimental results are summarized in Table VI, demonstrating that the proposed scheme achieves high MSE values and SSIM values close to 0, confirming the effective elimination of perceptual similarity between plaintext and ciphertext.

2) Histogram Analysis: A primary objective of secure image cipher is to render the statistical properties of the cipher-image independent of the plain-image. The pixel value histogram serves as a fundamental tool for this assessment, as it reveals the distribution of intensity levels. An effective encryption scheme should produce a cipher-image with a flat, nearly uniform histogram, irrespective of the distribution present in

TABLE VI
RESULTS OF MSE AND SSIM

Image	MSE	SSIM	Image	MSE	SSIM
Cameraman	10075.5	0.0094	Airplane	10002.8	0.0038
House	9436.6	0.0065	Parrot	8969.7	0.0019
Pepper	8796.5	0.0087	Barbara	9215.4	0.0046
Fishstar	9984.2	0.0042	Ship	9131.5	0.0073
Baboon	9876.8	0.0009	Man	9879.3	0.0085

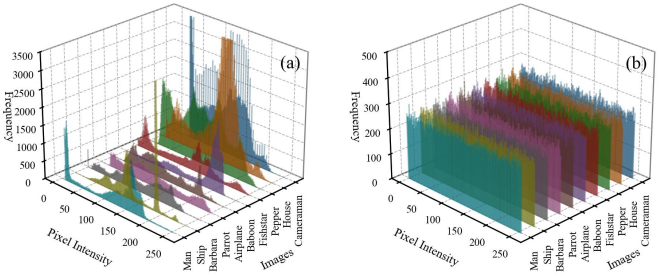

Fig. 11. Pixel value histogram distribution for (a) plain-images, (b) cipher-images.

the original image. This characteristic signifies that the cipher provides no statistical leverage for frequency-based attacks. As illustrated in Fig. 11, the histograms of the original test images, Fig. 11(a), exhibit distinct and varied distributions, with peaks and valleys corresponding to dominant intensity levels. In contrast, the histograms of their corresponding cipher-images, Fig. 11(b), are all consistently flat and uniformly distributed across all intensity bins. This visual evidence indicates that the encryption process successfully obscures the statistical characteristics of the plaintext.

3) Correlation Coefficient: Beyond histogram analysis, resisting statistical attacks necessitates the disruption of strong correlations between adjacent pixels, which is an inherent characteristic of natural images. A robust encryption scheme should ideally reduce this correlation to zero, making the cipher-image resemble random noise. The correlation distribution of adjacent pixels for the Cameraman is visually presented in Fig. 12 (a), which comprises scatter plots for images along the horizontal (H), vertical (V), and diagonal (D) directions. The plain-image scatter plots exhibit a linear concentration, revealing the high correlation. In contrast, the corresponding cipher-image scatter plots display a random distribution, indicating the successful disruption of spatial relationships by the proposed algorithm.

For a quantitative assessment across the test, the correlation coefficients are illustrated in Fig. 12(b). The results show that the correlation coefficients of the plain-images, represented by the solid line, fluctuate within a high range of 0.8 to 1.0 across all three directions. Conversely, the correlation coefficients of the cipher-images, denoted by the dashed line, are reduced and oscillate closely around zero.

4) Information Entropy: Information entropy serves as a pivotal metric to quantify the uncertainty and randomness inherent in an information source. In the context of image encryption, it measures the average amount of information contained in each pixel. For a random grayscale image with 256

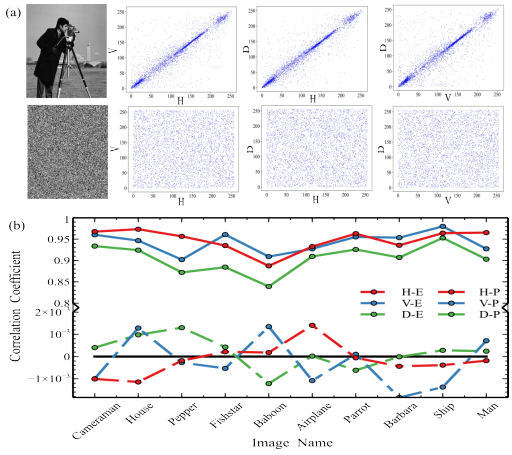

Fig. 12. (a) Adjacent pixels correlation distribution for Cameraman in different directions, (b) correlation coefficients of adjacent pixels for all test images.

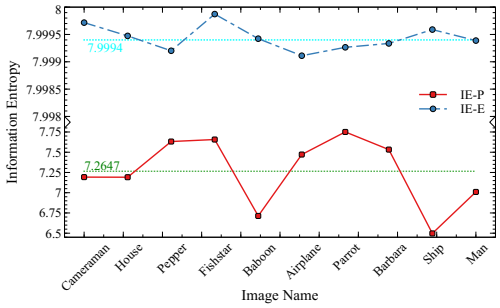

Fig. 13. Information entropy for plain and cipher images.

possible intensity levels, the theoretical maximum information entropy is 8. Consequently, an effective encryption algorithm needs to produce a cipher image whose information entropy is close to this ideal value, indicating that the pixel distribution is nearly random and unpredictable. The information entropy results for the test images are summarized in Fig. 13. The graph delineates the entropy values of the plain images, represented by the solid red line, and those of the cipher images, denoted by the dashed blue line. The entropy curve for the plain images fluctuates around a mean value of 7.2647, reflecting the inherent redundancy and non-uniformity of natural images. In contrast, the entropy curve for the cipher images is tightly clustered around a mean value of 7.9994. This value is close to the theoretical maximum of 8, demonstrating that the output of the proposed encryption algorithm possesses a high degree of randomness. To provide a performance comparison, Table VII compares the information entropy values of classical cipher images with the existing studies. The achieved better results manifest that the superior capability of the CVDHNN-based encryption scheme in generating cipher images with a high degree of randomness and uncertainty.

5) Key Sensitivity: A crucial attribute of a secure image encryption scheme is its key sensitivity, which demands that the ciphertext be extremely sensitive to minute alterations in the secret key. This characteristic is quantitatively assessed using the number of bit change rate (NBCR). The NBCR

TABLE VII
COMPARISON OF INFORMATION ENTROPY

Image	[33]	[34]	[35]	[36]	Proposed
Cameraman	7.9997	–	7.9986	7.9993	7.9997
Pepper	7.9991	7.9993	7.9987	7.9994	7.9992
Baboon	7.9966	7.9993	7.9987	7.9993	7.9994
Barbara	7.9997	7.9993	7.9987	7.9992	7.9993

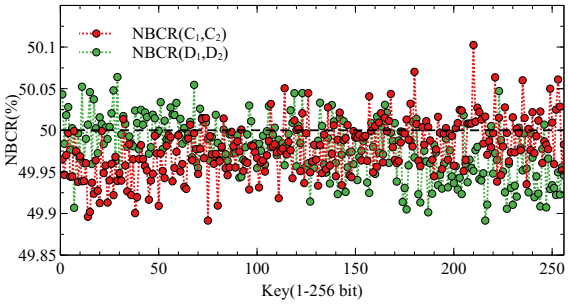

Fig. 14. Key sensitivity analysis measured by the NBCR.

between two images I_1 and I_2 represented by their bit-level matrices B_1 and B_2 of dimensions $M \times N \times 8$, is defined as

$$\text{NBCR}(I_1, I_2) = \frac{100\%}{MN} \cdot \frac{\|B_1 \oplus B_2\|_1}{8} \tag{24}$$

where $\oplus$ denotes the bit-level XOR operation, and $\|\cdot\|_1$ is the L1 norm. This metric is applied in two critical scenarios to assess the sensitivity during encryption and decryption, where $\text{NBCR}(C_1, C_2)$ denotes the cipher images encrypted with the original key and a one-bit changed key, and $\text{NBCR}(D_1, D_2)$ represents the images decrypted with the correct key and a faulty key. The sensitivity of the proposed scheme is tested on its 256-bit user-defined private key. Fig. 14 plots the NBCR values resulting from sequentially flipping each bit of the key. The results demonstrate that both metrics fluctuate within a narrow range around the ideal value of 50%. The observed values exhibit a deviation of 0.2167% from the ideal. This indicates that an average of nearly half of all pixel bits are altered by a single-bit key change, which can effectively resist brute-force and differential attacks.

6) NPCR and UACI analysis: Generally, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are utilized to evaluate the sensitivity of the encryption algorithm to plaintext variations, specifically measuring the differential impact when a single pixel in the plaintext image is modified. These metrics are critical indicators of the algorithm's resistance to differential cryptanalysis. Ideally, a secure image encryption scheme should yield an NPCR value approaching 99.609% and a UACI value approaching 33.4635% for 8-bit grayscale images, signifying that a minute change in the plaintext propagates to approximately half of the ciphertext pixels with uniform intensity distribution. Mathematically, NPCR and UACI are defined as:

$$\begin{aligned} \text{NPCR} &= \frac{\sum_{ij} D(i,j)}{M \times N} \times 100\% \\ \text{UACI} &= \frac{1}{M \times N} \sum_{ij} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \end{aligned} \tag{25}$$

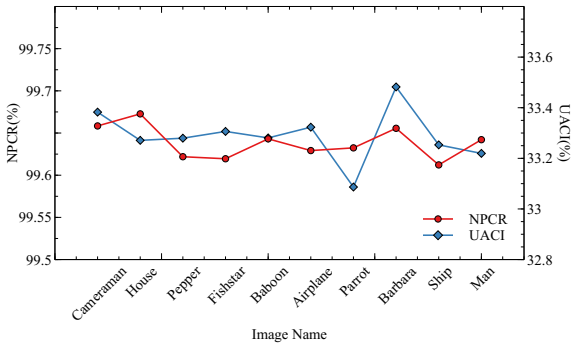

Fig. 15. NPCR and UACI results for different images

where M and N denote the height and width of the image, respectively; C_1 and C_2 represent the cipher images encrypted from the original plaintext and the modified plaintext with a 1-bit least significant bit alteration at a randomly selected pixel position. The experimental results are illustrated in Fig. 15, presenting the distributions of NPCR and UACI values across all tested images. The average NPCR and UACI values are 99.618% and 33.367%, respectively. These results demonstrate that the proposed scheme achieves NPCR and UACI values that are close to theoretical expectations. This confirms that the encryption algorithm possesses strong resistance against differential attacks, as even a single-bit modification in the plaintext results in substantial and unpredictable changes in the ciphertext.

The analyses of pixel histogram uniformity, adjacent pixel correlation, information entropy, and key sensitivity collectively demonstrate that the cipher effectively resists statistical attacks and exhibits the characteristics required for a secure cryptosystem. The results confirm that the algorithm successfully conceals visual information and is highly sensitive to the secret key.

7) Comparison analysis: To investigate whether the effectiveness of the encryption system is affected by complex chaotic systems, we designed the following ablation experiments, which are essential to validate the contribution of this component [37], [38]. Under the fixed encryption scheme, the only varying component is the chaotic generator: real-valued DHNN [39], versus the proposed CVDHNN. To mitigate initialization bias and assess statistical consistency, we performed 100 independent runs on the Set12 dataset with randomly initialized personal keys for each configuration and reported the mean value with standard deviation of security metrics. The comparative results are summarized in Table VIII. It can be observed that, under this specific encryption framework, CVHNN exhibits better performance than real-valued DHNN across all evaluated metrics. Specifically, CVHNN reduces the horizontal and vertical coefficients and achieves higher information entropy and the NCPR/UACI values are closer to theoretical benchmarks. A possible explanation for this observation advantage lies in the native compatibility between the complex-valued chaos dynamics and several operations within our encryption process. Certain components, such as polar-domain scrambling and complex-domain representation

TABLE VIII
COMPARISON OF ENCRYPTION PERFORMANCE BY USING CVDHNN AND DHNN

Chaos source	Correlation			Entropy	Differential attack	
	H	V	D		NCPR	UACI
DHNN	0.0115±0.0034	-0.0075±0.0008	0.0065±0.0002	7.9781±0.0008	99.56±0.04	33.21±0.08
CVDHNN	-0.0015±0.0018	0.0029±0.0006	0.0032±0.0001	7.9976±0.0005	99.61±0.002	33.35±0.002

TABLE IX
ENCRYPTION PERFORMANCE FOR DIFFERENT DATA TYPES

Data type		Correlation(H)	Entropy	Time(s)
Color	San Diego	0.0075	7.9986	0.587
	Oakland	-0.0013	7.9979	0.433
	Foster City	0.0086	7.9983	0.516
High-resolution	4K	0.0021	7.9938	4.52
Video frames	Foreman	0.0016	7.9979	6.83
	News	0.0038	7.9991	6.57
	Coastguard	0.0016	7.9984	6.65

of pixel states are specifically designed to operate on complex-valued quantities. When driven by real-valued DHNN, however, the real-valued outputs must be artificially paired or mapped to form complex numbers. This impedance mismatch may introduce structural redundancy or weaken the nonlinear mixing effect.

To further evaluate the generalizability and scalability of the proposed scheme, we extend our experiments to three additional scenarios: color images, high-resolution grayscale images, and video frames. We test the scheme on standard 512×512 RGB test images (San Diego, Oakland and Foster City from the USC-SIPI dataset). Each color channel is treated as an independent grayscale image and encrypted separately. The average correlation coefficients, entropy, NCPR and time across the three channels are reported in Table IX. The results demonstrate that the proposed scheme maintains security performance comparable to that of grayscale cases. We evaluate the scheme on a 4K image from SRITM-4K image dataset. The encryption completes within 4.52 seconds on our testing platform, indicating that the algorithm remains practically feasible for large-scale images. We select three CIF-format (352×288) test sequences: Foreman, News, and Coastguard. The first 50 frames of each sequence are encrypted independently as individual images. As shown in Table IX, both metrics remain close to their theoretical ideal values. These extended experiments demonstrate that the proposed scheme is not limited to grayscale images, but can be extended to color images, high-resolution images, and video applications.

VI. CONCLUSION AND OUTLOOK

This paper has established a comprehensive framework for neuro-inspired complex chaos and its application in secure image encryption through the introduction of the CVDHNN. We have demonstrated that the CVDHNN serves as a potent hyperchaotic source, whose rich dynamics, rigorously validated through numerical analysis, offer a superior alternative to conventional real-valued systems. The successful FPGA implementation further confirmed its physical realizability and efficiency for high-speed sequence generation. By leveraging these complex chaotic sequences, we developed a novel encryption scheme that integrates multi-stage confusion and diffusion. Extensive security analyses affirmed the cipher's robustness, showcasing its ability to resist statistical, differential, and brute-force attacks by achieving near-ideal encryption metrics.

While the proposed encryption algorithm achieves strong confusion and diffusion properties through its ciphertext-feedback design, we acknowledge several limitations that warrant future investigation. First, the current design prioritizes cryptographic strength over robustness against transmission errors, which may render the scheme unsuitable for noisy channels without additional error-correcting codes. Future work will explore the hybrid designs that maintain cryptographic binding while enabling localized error containment, as well as the integration of fountain codes as an outer layer for noisy channel applications. Second, the SHA-256 computations introduce non-negligible overhead compared to static-key chaotic ciphers. In resource-constrained environments, this hashing cost may limit throughput. Future research may focus on hardware acceleration strategies and the design of lightweight hash alternatives. Finally, the current implementation does not explicitly address timing attacks or power analysis on the chaotic iteration loops. Future hardware-oriented implementations will incorporate constant-time chaotic iteration using bit-masking techniques and randomized iteration counts to obscure Hamming weight leakage.

Building on the findings of this study, our future research will primarily focus on the following directions. First, we will address the identified implementation limitations of the current encryption scheme, including robustness against transmission errors, computational efficiency in resource-constrained environment, and resistance to side-channel attacks. Second, we will explore the integration of this neuro-inspired chaotic framework with deep learning-based perceptual models to develop semantically aware encryption schemes that jointly optimize cryptographic strength and visual information preservation. Third, we intend to extend the CVDHNN framework to model spatiotemporal chaos and investigate architectures involving coupled CVDHNN networks.

REFERENCES

[1] Q. Sheng, C. Fu, M. Tie, X. Wang, J. Chen, and C.-W. Sham, "A chaos-based tunable selective encryption algorithm for H.265/HEVC with semantic understanding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 11, pp. 11 040–11 055, 2024.

[2] F. Yu, S. He, W. Yao, S. Cai, and Q. Xu, "Bursting firings in memristive Hopfield neural network with image encryption and hardware implementation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. Early Access, DOI 10.1109/TCAD.2025.3567878, pp. 1–1, 2025.

[3] G. Yang, C. Wang, Y. Sun, and Q. Deng, "A class of discrete memristive hyperchaotic maps with multi-cavity multi-structure attractors and its application in secure communication," *IEEE Transactions on Industrial Informatics*, vol. Early Access, DOI /10.1109/TII.2026.3663427, 2026.

[4] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, C. Wang, and X. Tang, "Asynchronous updating boolean network encryption algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4388–4400, 2023.

[5] S. Zhang, X. Peng, X. Wang, C. Chen, and Z. Zeng, "A novel memristive multiscroll multistable neural network with application to secure medical image communication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 2, pp. 1774–1786, 2025.

[6] Q. Deng, C. Wang, and G. Yang, "Discrete memristor-based complex-valued chaotic system dynamics and application in dual-image encryption," *Acta Physica Sincia*, vol. Early Access, 2025.

[7] H. Li and F. Min, "Attractor dynamics of 2-lobe discrete corsage memristor-coupled neuron map," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 72, no. 9, pp. 4820–4829, 2025.

[8] G. Yang, C. Wang, Y. Sun, and Q. Deng, "Delayed discrete memristive ring neural network and application in pseudorandom number generator," *IEEE Internet of Things Journal*, vol. Early Access, DOI 10.1109/JIOT.2025.3646638, 2025.

[9] Q. Deng, C. Wang, Y. Sun, C. Xu, H. Lin, and Z. Deng, "Memristor-based brain emotional learning neural network with attention mechanism and its application," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. Early Access, DOI 10.1109/TCAD.2025.3567534, 2025.

[10] D. Luo, C. Wang, J. Liang, and Q. Deng, "Memristor coupled fractional-order Hopfield neural network composed by heterogeneous neurons and its FPGA implementation," *Nonlinear Dynamics*, vol. 113, pp. 29 983–29 998, 2025.

[11] Q. Lai and L. Ji, "A bidirectional cross-scrambling medical image encryption scheme incorporates compressed sensing and its application in iomt," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 8, pp. 7697–7705, 2025.

[12] G. Yang, C. Wang, Y. Sun, and Q. Deng, "A discrete memristive hopfield neural network with grid multi-structure/scroll-like attractors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. Early Access, DOI 10.1109/TCAD.2026.3661893, 2026.

[13] S. Zhang, D. He, Y. Li, D. Lu, and C. Li, "Dual memristor-coupled Hopfield neural network with any multi-scroll amplitude control and its application for medical image classification," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 17 828–17 840, 2025.

[14] H. Lin, X. Deng, F. Yu, and Y. Sun, "Diversified butterfly attractors of memristive HNN with two memristive systems and application in IoMT for privacy protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 1, pp. 304–316, 2025.

[15] B. Bao, H. Tang, H. Bao, Z. Hua, Q. Xu, and M. Chen, "Simplified discrete two neuron Hofpield neural network and FPGA implementation," *IEEE Transactions on Industrial Electronics*, vol. 72, no. 4, pp. 4105–4115, 2025.

[16] Y. Zhang, Z. Hua, H. Bao, and H. Huang, "Multi-valued model for generating complex chaos and fractals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 6, pp. 2783–2796, 2024.

[17] Z. Hua, J. Yao, Y. Zhang, H. Bao, and S. Yi, "Two-dimensional coupled complex chaotic map," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 85–95, 2025.

[18] J. Yao, Y. Zhang, H. Bao, and Z. Hua, "Generation of n-dimensional complex chaotic system via parameter matrix configuration," *Chaos, Solitons & Fractals*, vol. 197, p. 116453, 2025.

[19] S. Gao, Z. Zhang, Q. Li, S. Ding, H. H.-C. Iu, Y. Cao, X. Xu, C. Wang, and J. Mou, "Encrypt a story: A video segment encryption method based on the discrete sinusoidal memristive rulkov neuron," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 6, pp. 8011–8024, 2025.

[20] S. Gao, R. Wu, H. H.-C. Iu, U. Erkan, Y. Cao, Q. Li, A. Toktas, and J. Mou, "Chaos-based video encryption techniques: A review," *Computer Science Review*, vol. 58, p. 100816, 2025.

[21] Y. Yang, L. Huang, N. V. Kuznetsov, B. Chai, and Q. Guo, "Generating multiwing hidden chaotic attractors with only stable node-foci: Analysis, implementation, and application," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 4, pp. 3986–3995, 2024.

[22] Y. Guang, Q. Ding, and D. Liu, "FPGA-oriented design and efficient implementation of a geometrically tunable multiscroll conservative chaotic system without equilibrium points," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 33, no. 9, pp. 2528–2541, 2025.

[23] D. Ding, D. Xie, H. Zhang, Z. Yang, and C. Liu, "Deepface-based chaotic image encryption using key optimization and semi-tensor product theory," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 7, pp. 6522–6534, 2025.

[24] L. Teng, P. Cao, and Y. Liu, "Multi-image encryption algorithm based on novel spatiotemporal chaotic system and dynamical chaotic trajectories," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 2, pp. 1562–1575, 2025.

[25] M. Gong, X. Chai, Y. Lu, and Y. Zhang, "Exploiting four-dimensional chaotic systems with dissipation and optimized logical operations for secure image compression and encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 8, pp. 7628–7642, 2024.

[26] L.-W. Li, C.-N. Lee, K. Gupta, H.-F. Yang, and A. K. Singh, "Syntax element encryption for H.265/HEVC using chaotic map-based coefficient scrambling scheme," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. Early Access, DOI 10.1109/TCSVT.2025.3625077, pp. 1–1, 2025.

[27] F. Yu, G. Yumba, Musoya, R. Guo, Z. Ying, J. Xu, W. Yao, J. Jin, and H. Lin, "Dynamic analysis and application of 6D multistable memristive chaotic system with wide range of hyperchaotic states," *Axioms*, vol. 14, no. 8, p. 638, 2025.

[28] D. Luo, C. Wang, Q. Deng, and G. Yang, "Discrete memristive hyperchaotic maps with high Lyapunov exponents," *Nonlinear Dynamics*, vol. 113, pp. 28 381–28 395, 2025.

[29] S. He, F. Yu, R. Guo, M. Zheng, T. Tang, J. Jin, and C. Wang, "Dynamic analysis and FPGA implementation of a fractional-order memristive hopfield neural network with hidden chaotic dual-wing attractors," *Fractal and Fractional*, vol. 9, no. 9, p. 561, 2025.

[30] A. Wassim and M. Youstina, "A new fast high dimensional and memristive hyperchaotic multiple image encryption method and its FPGA implementation," *Discover Electronics*, vol. 2, p. 75, 2025.

[31] Q. Deng, C. Wang, Y. Sun, and G. Yang, "Delay difference feedback memristive map: dynamics, hardware implementation and application in path planning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. Early Access, DOI 10.1109/TCSI.2025.3571961, 2025.

[32] F. Yu, X. Kong, W. Yao, J. Zhang, S. Cai, H. Lin, and J. Jin, "Dynamics analysis, synchronization and FPGA implementation of multiscroll Hopfield neural networks with non-polynomial memristor," *Chaos, Solitons, & Fractals*, vol. 179, p. 114440, 2024.

[33] H. Zhang, H. Hu, and W. Ding, "Image encryption algorithm based on Hilbert sorting vector and new spatiotemporal chaotic system," *Optics & Laser Technology*, vol. 167, p. 109655, 2023.

[34] Q. Lai, L. Yang, G. Hu, Z.-H. Guan, and H. H.-C. Iu, "Constructing multiscroll memristive neural network with local activity memristor and application in image encryption," *IEEE Transactions on Cybernetics*, vol. 54, no. 7, pp. 4039–4048, 2024.

[35] S. Kumar and D. Sharma, "A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm," *Artificial Intelligence Review*, vol. 57, no. 4, p. 87, 2024.

[36] U. Erkan, A. Toktas, S. Memis, F. Toktas, Q. Lai, H. Wen, and S. Gao, "OSMRD–IE: Octal-based shuffling and multilayer rotational diffusing image encryption using 2-D hybrid Michalewicz–Ackley map," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 35 113–35 123, 2024.

[37] S. Deb, A. Das, and N. Kar, "An applied image cryptosystem on Moore's automaton operating on $\delta(q^k)/\mathbb{F}_2$, enhancing image security via block cyclic construction and DNA-based LFSR," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, DOI 10.1145/3614433, no. 2, pp. 1–20, 2023.

[38] S. Deb and P. K. Behera, "Design of key-dependent bijective s-boxes for color image cryptosystem," *OPTIK*, vol. 253, p. 168548, 2022.

[39] G. Yang, C. Wang, Y. Sun, and Q. Deng, "A discrete memristive heterogeneous Hopfield neural network with multi-penguin-like/silkworm-like attractors and its application in secure communication," *Nonlinear Dynamics*, vol. Early Access, DOI 10.1007/s11071-025-12127-7, 2026.

Quanli Deng received the B.S. degree in microelectronics from the School of Physics and Optoelectronics, Xiangtan University, Xiangtan, China, in 2016, the M.S. in information and communication engineering from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2020 and the Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2024.

He is currently a Postdoctoral Research Fellow with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include modeling and analysis of neural systems, fundamental theory of nonlinear systems and circuits, and analog implementation of neuromorphic systems.

Chunhua Wang received the M.S. degree in microphysics from Zhengzhou University, Zhengzhou, China, in 1994, and the Ph.D. degree in microelectronics and solid-state electronics from the Beijing University of Technology, Beijing, China, in 2003.

He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where he is a Doctor Tutor and Director of the Advanced Communication Technology Key Laboratory. He has presided more than eight national and provincial projects and authored or coauthored more than 200 papers retrieved by SCI, among which 20 papers were highly cited. His research interests include chaotic circuit, memristor circuit, chaotic encryption, neural networks based on memristor, complex networks, and current-mode circuit.

Dr. Wang is the Director of the Chaos and Nonlinear Circuit Professional Committee of Circuit and System Branch of the China Electronic Society.

Yichuang Sun (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

Dr. Sun is currently Professor of Communications and Electronics, Head of Communications and Intelligent Systems Research Group, and Head of Electronic, Communication and Electrical Engineering Division in the School of Engineering and Computer Science of the University of Hertfordshire, UK. He has published over 420 papers and contributed 10 chapters in edited books. He has also published four text and research books: Continuous-Time Active Filter Design (CRC Press, USA, 1999), Design of High Frequency Integrated Analogue Filters (IEE Press, UK, 2002), Wireless Communication Circuits and Systems (IET Press, 2004), and Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits - the Systems on Chip Approach (IET Press, 2008). His research interests are in the areas of wireless and mobile communications, RF and analogue circuits, memristor circuits and systems, and machine learning and neuromorphic computing.

Gang Yang received the B.S. degree in communications engineering and the M.S. degree in artificial intelligence form the Jiangxi University of science and technology, Ganzhou, China, in 2020 and 2023, respectively. He is currently working toward the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

His current research interests include chaotic systems and circuits, memristor neural networks, and memristor systems and circuits.