Mathematical and Computational Applications

MDPI

Article

A 3D Memristive Hyperchaotic Map with Bounce-Escape Zigzag Scrambling for Color Image Encryption

Muhammad Hayat [1], M. G. Abbas Malik [2,*] and Zia Bashir [1]

[1] Department of Mathematics, Quaid-i-Azam University, Islamabad 45320, Pakistan; mhayat@math.qau.edu.pk (M.H.); zbashir@qau.edu.pk (Z.B.)
[2] Interdisciplinary Sustainable Systems (IS2) Research Group, College of Computer and Information Science, Prince Sultan University, Riyadh 11586, Saudi Arabia
* Correspondence: amaalik@psu.edu.sa

Abstract

As the need for secure image encryption grows, chaotic cryptography using hyperchaotic systems has attracted significant attention. This article proposes a novel three-dimensional (3D) memristive hyperchaotic map. We analyze the map using phase portraits, Lyapunov exponents, sensitivity analysis, the 0-1 test, Lyapunov dimension, bifurcation diagrams, and Kolmogorov–Sinai entropy. The map has two positive Lyapunov exponents, and the Kaplan–Yorke dimension equals three, confirming strong hyperchaotic dynamics across a wide parameter range. Based on this map, we introduce a new color image encryption scheme. It incorporates a key derived from both an external 256-bit secret key and the SHA-256 hash of the plaintext; a novel Bounce-Escape Zigzag (BEZ) scrambling algorithm with boundary-bouncing diagonal trajectories, variable-length diagonal traversals, and adaptive collision escape; dynamic DNA encoding; and chaotic XOR diffusion. Comprehensive security analysis on standard test images demonstrates near-ideal entropy values, NPCR and UACI close to the theoretical ideals, and correlation coefficients close to zero. The proposed scheme also shows strong robustness against differential, known-plaintext, and chosen-plaintext attacks, as well as cropping and noise corruption, with an effective key space of 2^{256}. By integrating hyperchaotic dynamics, BEZ scrambling, and multi-layer DNA-XOR diffusion, the proposed method achieves superior performance. It is well-suited for secure image communication and storage applications.

Keywords: memristive hyperchaotic map; DNA encoding; Bounce-Escape Zigzag scrambling; chaotic cryptography

Check for updates

Academic Editor: Guillermo Valencia-Palomo

Received: 22 September 2026
Revised: 5 October 2026
Accepted: 6 October 2026
Published: 9 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid development of information technology, digital images have become a central medium for communication, storage, and decision-making in modern society. They are widely used in sensitive domains, including medical diagnosis [1,2], satellite imaging [3], military surveillance, personal privacy, and online multimedia platforms. This ubiquity, however, introduces substantial security risks [4]. Digital images often contain confidential, commercial, or sensitive information, and unauthorized access during transmission or storage can lead to privacy breaches, data theft, or serious economic and safety consequences [5]. Effective image encryption methods are therefore needed to resist modern cryptographic attacks. Among such methods, chaos-based approaches have shown considerable potential [6–8].

While classical ciphers such as AES and DES provide strong cryptographic security for textual and binary data [9], their direct application to digital images can be inefficient. In basic modes such as Electronic Codebook (ECB), identical plaintext blocks map to identical ciphertext blocks, so the strong spatial correlation of images and their large data volume lead to structural leakage and perceptible patterns in the ciphertext, as illustrated by the classic ECB penguin image [10]. Secure modes such as CBC, CTR, or GCM eliminate these visual patterns. However, they treat images as linear byte streams and do not exploit the two-dimensional spatial correlation of natural images, often requiring additional permutation or scrambling stages. These limitations have motivated the adoption of chaotic image ciphers [11], whose properties such as initial-condition sensitivity, ergodicity, and pseudorandom behavior are naturally suited for simultaneous pixel scrambling and diffusion [12–15].

Given these considerations, the choice of chaotic map is critical. Discrete maps are generally preferred over continuous systems [16] because they are simpler to implement, computationally efficient, and compatible with digital hardware. Moreover, hyperchaos can emerge at lower dimensions in discrete systems than in continuous ones [17,18]. However, one-dimensional maps such as the logistic and sine maps have only one positive Lyapunov exponent and therefore cannot exhibit hyperchaotic behavior. Two-dimensional discrete maps can exhibit hyperchaos, but their chaotic parameter ranges are often limited, and their dynamics may be unstable or require additional bounding operations [19]. These limitations motivate the development of a higher-dimensional discrete map that remains hyperchaotic over a broad parameter range.

One promising route is memristor-based coupling, which has proven effective for improving the dynamical complexity of chaotic maps [20–22]. A memristor introduces state-dependent feedback and memory, which can strengthen a map's nonlinear behavior and widen its chaotic parameter range. Inspired by these properties, we propose a novel three-dimensional memristive hyperchaotic map that combines sine and logistic functions with a flux-controlled memristor term. The map has two positive Lyapunov exponents and remains hyperchaotic over a wide range of parameters; moreover, its bounded nonlinear terms and memristive feedback keep the dynamics intrinsically bounded. Building on this map, we design a color image encryption scheme that combines Bounce-Escape Zigzag (BEZ) scrambling, dynamic DNA coding, and chaotic XOR diffusion; we describe each stage in the following paragraphs.

In addition to the chaotic map, the scrambling stage is a key part of the image encryption scheme. Zigzag scanning is widely used for pixel shuffling because it is simple to implement. However, the classical zigzag pattern offers only limited confusion because its traversal order is regular and predictable [23]. Block-based dynamic zigzag methods boost randomness by using a chaotic starting point and randomly changing direction after every jump [24]. Nevertheless, they still rely on a fixed step length of one pixel, only two diagonal directions, and an arbitrary jump to any unvisited pixel when the walk is blocked. These design restrictions weaken the permutation's randomness and may introduce residual correlation in the scrambled image.

To address these limitations, this work proposes a novel Bounce-Escape Zigzag (BEZ) scrambling algorithm. Unlike conventional zigzag methods, BEZ performs a global traversal over the entire image using variable-length diagonal runs whose lengths are determined by chaotic values. The walk proceeds along one of four diagonal directions and reflects at matrix boundaries instead of stopping. When only one coordinate is out of bounds, a second chaotic value may flip the other component, producing a pinball-like reflection. If the next cell has already been visited, the algorithm performs an adaptive collision escape by first attempting to move to an unvisited cell in its 8-neighborhood and resorting to a global

jump only when none is available. After each escape, the traversal direction is updated according to the movement vector to preserve coherent diagonal motion. This procedure guarantees that every pixel is visited exactly once and produces a highly unpredictable traversal order that effectively destroys spatial correlation.

Beyond the scrambling stage, the diffusion stage must efficiently transform pixel values to defend against statistical and differential attacks. DNA is widely used for diffusion because it converts pixel values into DNA base representations and allows logical and arithmetic operations [25]. Although recent schemes have explored dynamic DNA encoding, several established methods [26] still apply a single fixed DNA rule to all pixels, limiting their dynamism. In the proposed scheme, each pixel value is encoded according to one of the eight DNA rules via the chaotic keystream, and four different DNA operations are applied to the four bases of every pixel using fixed operation tables. The encoding and decoding rules are selected dynamically per pixel, ensuring pixel-dependent diffusion. Moreover, the initial conditions and parameters of the memristive hyperchaotic map are generated from the SHA-256 hash of the plaintext image combined with an external 256-bit secret key. Therefore, even a one-bit change in the plaintext or the key produces a completely different ciphertext, which strengthens the scheme against known-plaintext and chosen-plaintext attacks.

The key contributions of this article are outlined as follows:

- We design a novel three-dimensional memristive hyperchaotic map integrating sine and logistic maps with a flux-controlled memristor. Dynamical analysis confirms robust hyperchaotic behavior with two positive Lyapunov exponents, and the map achieves larger Lyapunov exponents, Kolmogorov–Sinai entropy, and Lyapunov dimension than three recent 3D memristive maps.
- We introduce the Bounce-Escape Zigzag (BEZ) scrambling algorithm, which employs variable-length diagonal runs, boundary bouncing, and adaptive collision escape to overcome classical and block-based limitations. A formal proof guarantees termination and one-to-one coverage for arbitrary matrix dimensions.
- We construct a color image encryption scheme combining BEZ scrambling, dynamic DNA encoding with four operations per pixel, and a chaotic XOR mask. The key is derived from an external 256-bit secret key and the SHA-256 hash of the plaintext, and the receiver reconstructs it from the transmitted hash and key.
- We conduct a comprehensive evaluation of six images at three resolutions through statistical, differential, key-sensitivity, and key-space analyses, together with cryptanalytic arguments against chosen- and known-plaintext attacks. We compare entropy, MSE, NPCR, and UACI with recent schemes on standard test images. We quantify robustness to cropping and to salt-and-pepper and Gaussian noise using MSE, PSNR, and SSIM at several corruption levels. An ablation study isolates the BEZ, DNA, and XOR stages, and a complexity analysis reports each stage's cost along with encryption and decryption runtimes.

The remainder of the article is organized as follows. Section 2 presents the three-dimensional memristive hyperchaotic map and its dynamical analysis. Section 3 describes the image encryption scheme. Section 4 reports the security and performance analysis. Section 5 concludes the article.

2. Memristor-Based System and Its Dynamical Analysis

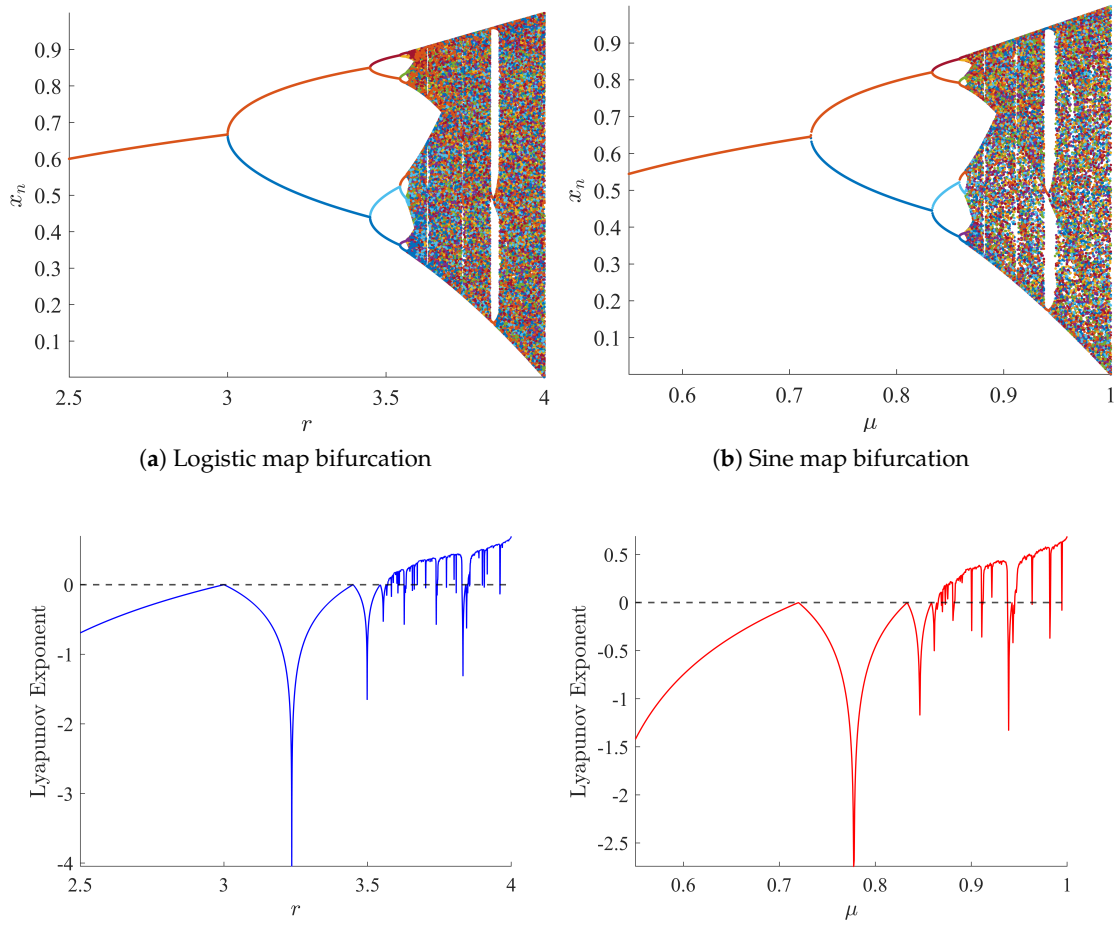
The logistic and sine maps are two classical one-dimensional (1D) maps. The logistic map is defined as follows:

$$x_{n+1} = r x_n (1 - x_n), \tag{1}$$

where r denotes the control parameter of the logistic map. This map exhibits chaotic behavior when the parameter value lies in the interval $r \in [3.57, 4]$. The sine map is written as follows:

$$x_{n+1} = \mu \sin(\pi x_n), \tag{2}$$

where μ is the control parameter of the sine map. The sine map shows chaotic dynamics for suitable values of μ. The bifurcation diagrams and Lyapunov exponents of both maps are shown in Figure 1, confirming their chaotic ranges.



(**a**) Logistic map bifurcation (**b**) Sine map bifurcation

(**c**) Logistic map Lyapunov exponent (**d**) Sine map Lyapunov exponent

Figure 1. Bifurcation diagrams and Lyapunov exponents of the classical logistic and sine maps, obtained with initial condition $x_0 = 0.2$.

Despite the wide use of 1D maps, they possess only one positive Lyapunov exponent and therefore cannot produce hyperchaotic behavior. Moreover, the chaotic ranges for these maps are narrow. These drawbacks motivate the development of higher-dimensional maps with memristor-like coupling.

In this work, we develop a 3D memristive hyperchaotic map by combining the nonlinear structures of the logistic and sine maps with a memristor-like coupling term. The proposed map is mathematically written as follows:

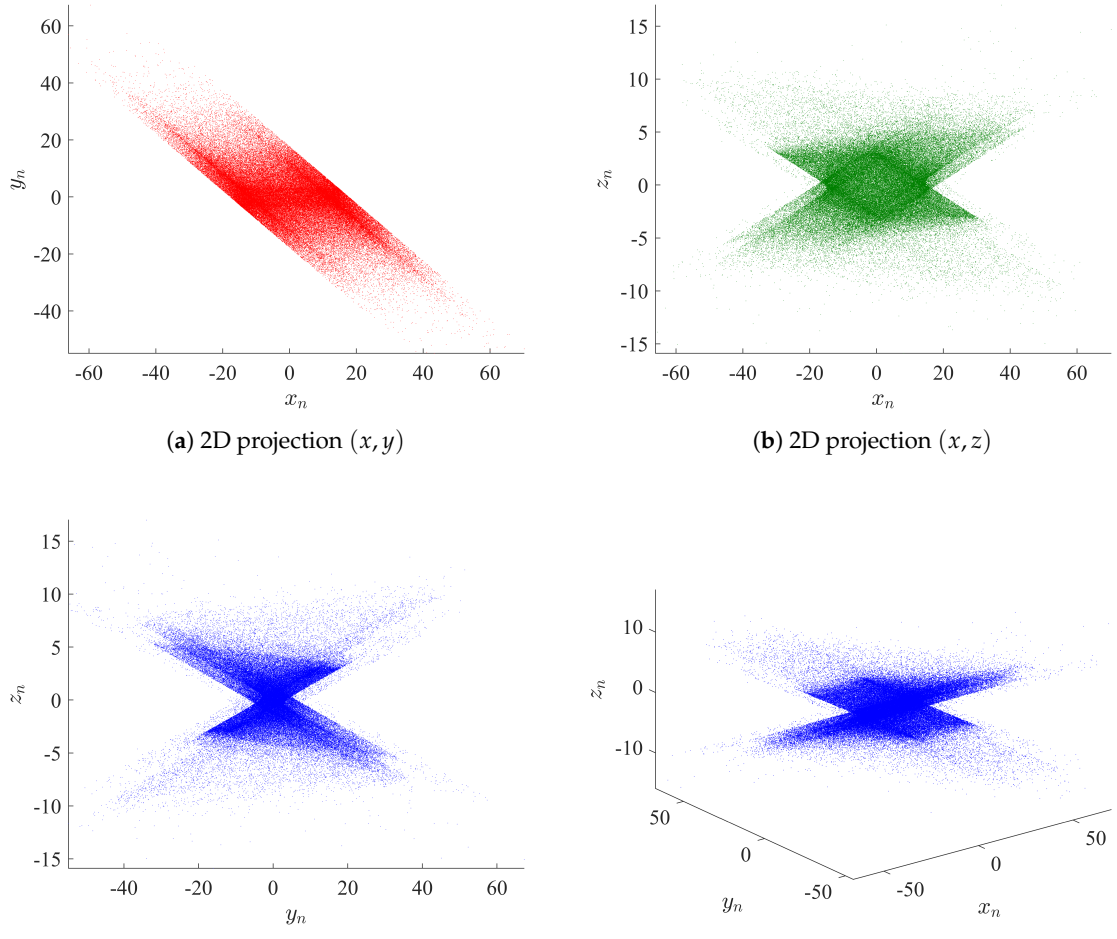
$$\begin{cases} x_{n+1} = a \sin\left(\pi b\, x_n(1 - y_n)\right) - x_n \cos(z_n) \\ y_{n+1} = c \sin\left(\pi d\, y_n(1 - x_n)\right) + x_n \cos(z_n) \\ z_{n+1} = 0.2\,(z_n + x_n) \end{cases} \tag{3}$$

where a, b, c, and d are the control parameters of the memristive map. In this system, the memristor is flux-controlled. The state variable z_n plays the role of the flux, while x_n acts as the voltage input that drives the flux update. The term $\cos(z_n)$ is the flux-controlled memductance. It introduces state-dependent feedback, strengthening the map's nonlinearity.

The map exhibits hyperchaotic behavior over a wide parameter range. The dynamical analysis in the following subsections confirms this behavior. Unless otherwise stated, the parameters are set to $a = 14.8$, $b = 3.59$, $c = 3.0$, and $d = 4.86$, with initial conditions $x_0 = 0.2$, $y_0 = 0.3$, and $z_0 = 0.4$.

2.1. Phase Space Analysis

Phase portrait analysis is an important tool for visualizing the dynamics of a dynamical system. Through this analysis, we can observe the system's periodic, quasi-periodic, and chaotic motion. Here, we used the analysis to observe the 3D memristive map. We chose the default parameter and initial condition values and iterated the map for 100,000 iterations. In Figure 2, the two-dimensional projections (x–y, x–z, and y–z) and a full three-dimensional view (x, y, z) are plotted. All these plots are bounded, dense, and scattered, and no discernible pattern is observed—a clear indication of chaos. The Lyapunov exponents of the map are $\lambda_1 = 6.368214$, $\lambda_2 = 3.193574$, and $\lambda_3 = -1.446380$. Two of these exponents are positive, a necessary condition for hyperchaos. Thus, the phase portraits combined with the two positive Lyapunov exponents confirm that our system exhibits complex hyperchaotic dynamics.



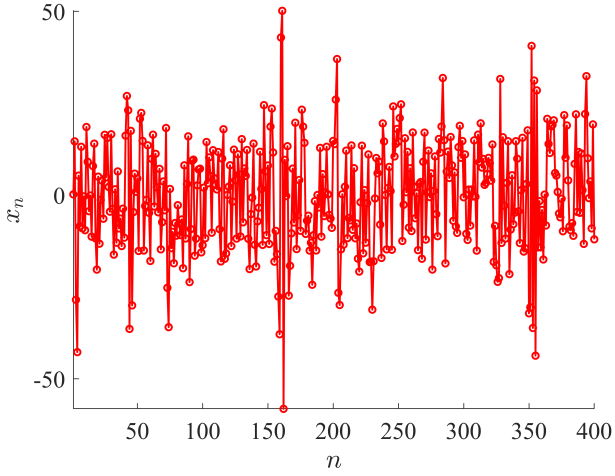
(**a**) 2D projection (x, y) (**b**) 2D projection (x, z)

(**c**) 2D projection (y, z) (**d**) 3D attractor (x, y, z)

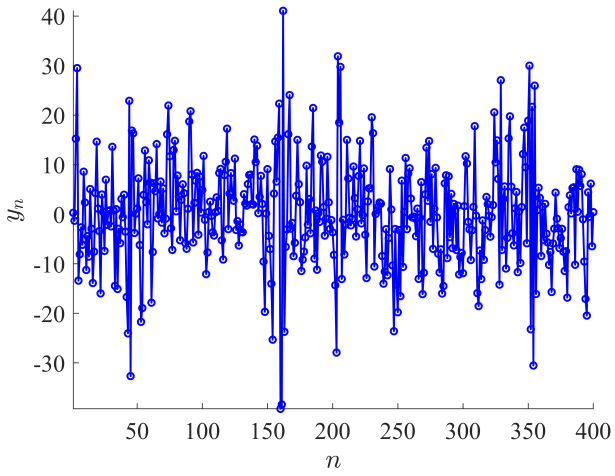
Figure 2. Phase portraits of the 3D memristive map.

2.2. Time Series Analysis

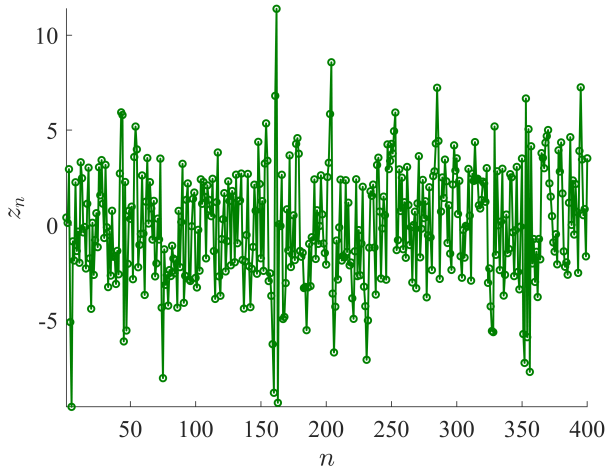
In the time series analysis of the discrete map, we plot each state variable versus the iteration step n. Here we show the first 400 iterated values of each state variable (giving 401 points when including the initial value at $n = 0$). Through time series analysis, we can also verify whether the dynamics are periodic or chaotic. Figure 3 shows the time series plots for each state variable. In all plots, we observe no periodic behavior; the trajectories exhibit irregular, aperiodic fluctuations with no discernible pattern, demonstrating the map's chaotic nature.



(**a**) Time series of x_n



(**b**) Time series of y_n

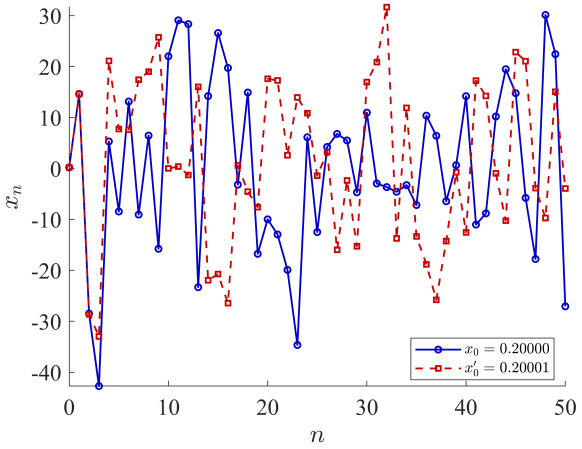


(**c**) Time series of z_n

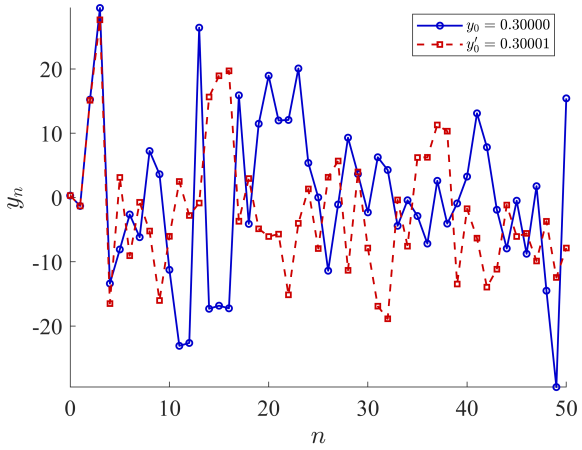
Figure 3. Time series of the 3D memristive map.

2.3. Sensitivity Analysis

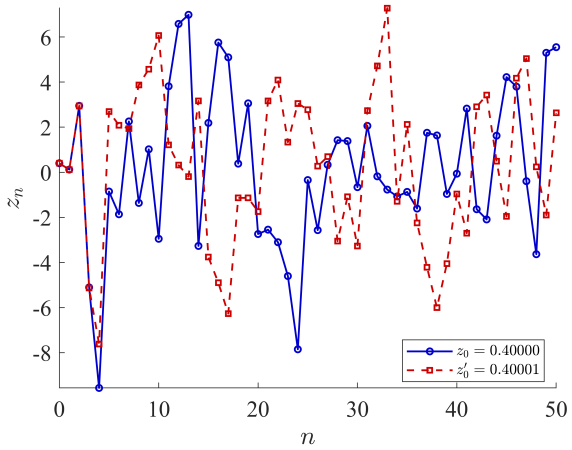
Sensitivity to initial conditions is the most important defining characteristic of a chaotic system. To properly verify this property in the 3D memristive map, we introduced a perturbation of $\delta = 10^{-5}$ to each state variable. Figure 4 plots the original and perturbed trajectories for the first 50 iterations of each state variable. In each subfigure, the trajectories start with only a tiny variation in the initial state, yet they rapidly diverge from the original trajectories and follow completely different paths. This extreme sensitivity, combined with the two positive Lyapunov exponents, confirms the map's complex hyperchaotic dynamics.



(**a**) Sensitivity of x_n



(**b**) Sensitivity of y_n



(**c**) Sensitivity of z_n

Figure 4. Sensitivity analysis of the 3D memristive map ($\delta = 10^{-5}$).

2.4. 0-1 Test Analysis

The 0-1 test is another metric that distinguishes between regular and chaotic dynamics. For chaotic behavior, the median K value approaches 1, whereas for regular motion it stays near 0. The median K values obtained for the state variables are $K_x = 0.998177$, $K_y = 0.998397$ and $K_z = 0.998175$, all very close to the ideal chaotic limit of 1, confirming chaotic dynamics in every variable. Figure 5 shows the p-q trajectories for each state variable. All three plots exhibit Brownian-like motion, further confirming the map's chaotic nature.

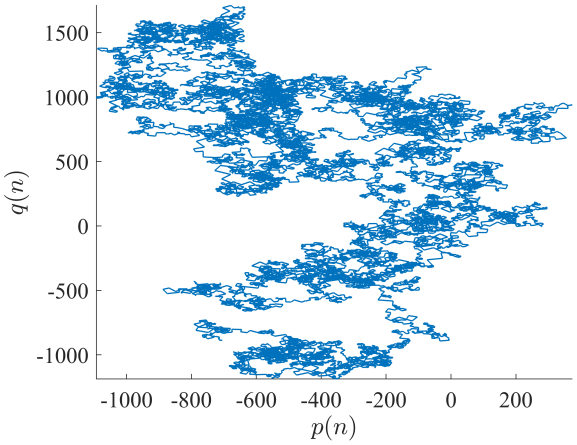
2.5. Lyapunov Dimension Analysis

The Lyapunov dimension quantifies the structural complexity of a chaotic attractor. The Kaplan–Yorke formula defines it as:

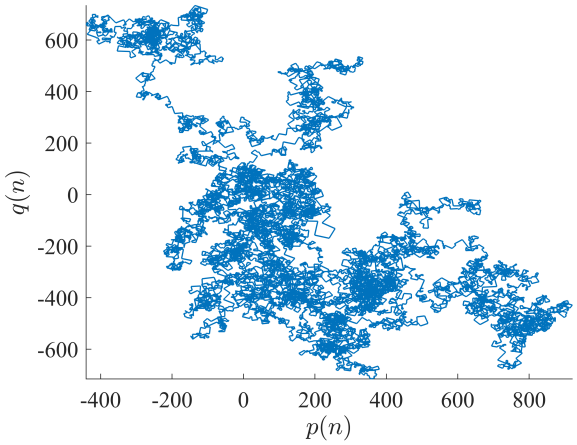
$$D_L = j + \frac{1}{|\lambda_{j+1}|} \sum_{i=1}^{j} \lambda_i,$$

where the Lyapunov exponents λ_i are sorted in descending order and j is the largest integer such that the cumulative sum $\sum_{i=1}^{j} \lambda_i$ remains positive. For the proposed 3D memristive map, we obtained $\lambda_1 = 6.368214$, $\lambda_2 = 3.193574$, and $\lambda_3 = -1.446380$. The cumulative sum of all three exponents is positive, so the Kaplan–Yorke dimension equals

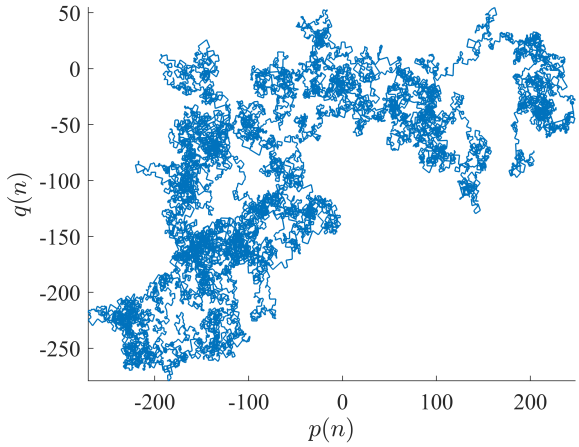
the state-space dimension, $D_L = 3$. This indicates that the attractor's fractal complexity numerically saturates the state-space dimension. While $D_L = 3$ does not imply literal space-filling behavior, it confirms both the presence of two positive Lyapunov exponents and a dimension equal to the state-space dimension, a strong indicator of hyperchaotic dynamics. Such a dimension-saturated attractor is well-suited for cryptographic applications.



(**a**) p-q plot, x_n



(**b**) p-q plot, y_n



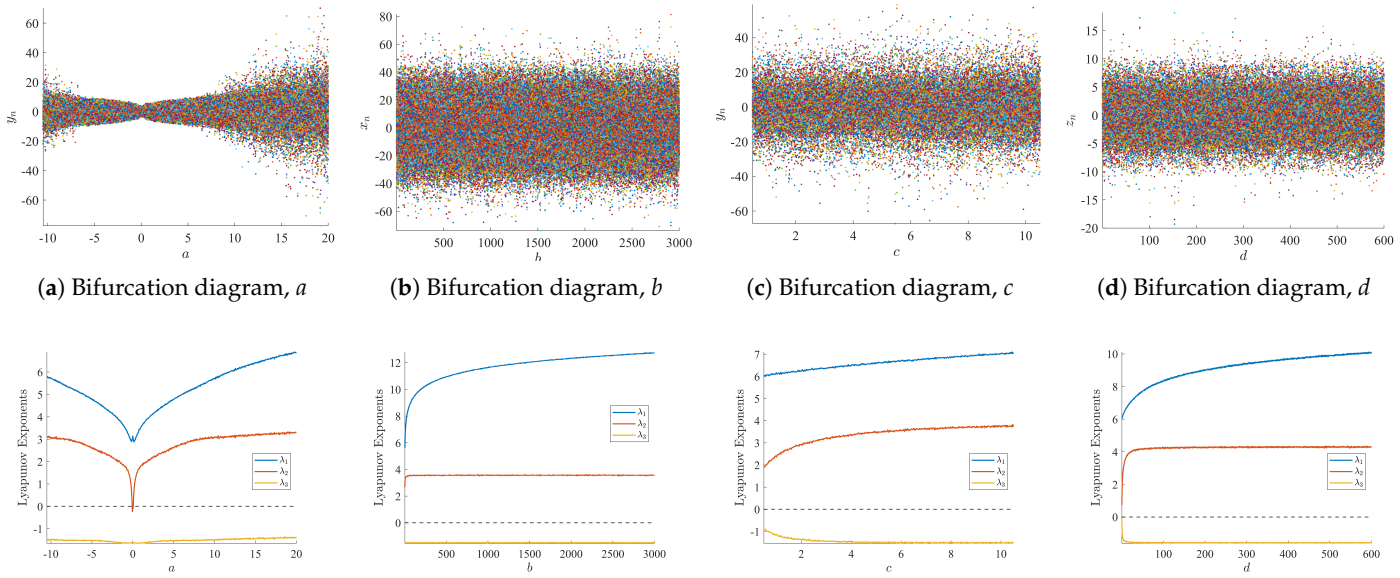
(**c**) p-q plot, z_n

Figure 5. p-q plots of the state variables x_n, y_n, and z_n.

2.6. Bifurcation and Lyapunov Exponent Analysis

A bifurcation diagram captures qualitative changes in a system's behavior as a parameter varies, while the Lyapunov exponent provides a quantitative measure that distinguishes regular, chaotic, and hyperchaotic dynamics. Here, we vary the control parameters individually: $a \in [-10.5, 20]$, $b \in [1, 3000]$, $c \in [0.5, 10.5]$, and $d \in [0.1, 600.1]$, while keeping the remaining parameters and the initial condition at their default values. Figure 6 displays the bifurcation diagrams and corresponding Lyapunov exponent spectra for these varying parameters. For a (Figure 6a,e), the bifurcation diagram is densely filled, and the second Lyapunov exponent is negative near zero for some parameter values, indicating ordinary chaos. For the rest of the range, the second exponent becomes positive, giving hyperchaos. For b, c, and d (Figure 6b–d,f–h), the bifurcation diagrams are dense with no visible periodic windows, and two Lyapunov exponents remain positive throughout the tested ranges, confirming hyperchaotic dynamics. Overall, the system remains strongly hyperchaotic across parameter variations, except for isolated intervals near zero when a is

varied, where it is merely chaotic. Hence, the 3D memristive map possesses rich and robust hyperchaotic dynamics.



(**a**) Bifurcation diagram, a (**b**) Bifurcation diagram, b (**c**) Bifurcation diagram, c (**d**) Bifurcation diagram, d

(**e**) Lyapunov exponents, a (**f**) Lyapunov exponents, b (**g**) Lyapunov exponents, c (**h**) Lyapunov exponents, d

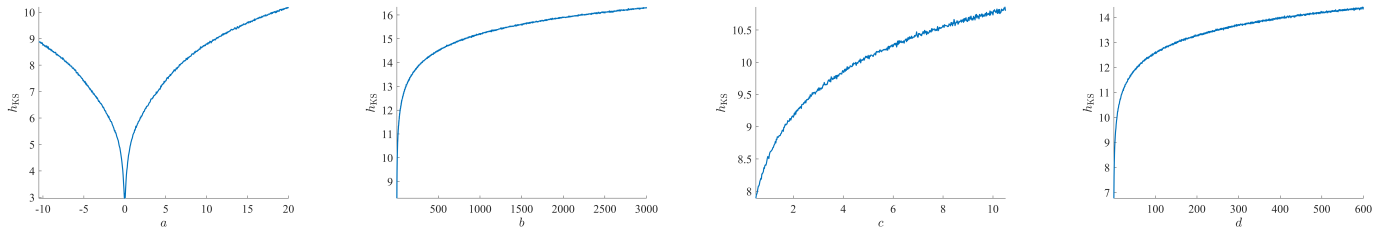
Figure 6. Bifurcation diagrams and Lyapunov exponent spectra for varying control parameters a, b, c, and d.

2.7. Kolmogorov–Sinai Entropy Analysis

Kolmogorov–Sinai (KS) entropy is an important quantitative measure of the rate of information production in a dynamical system. A positive KS entropy indicates chaotic behavior, whereas zero KS entropy corresponds to regular dynamics. For the proposed map, we compute the KS entropy using the Pesin identity, which states that KS entropy equals the sum of all positive Lyapunov exponents. Mathematically, this is expressed as:

$$h_{\mathrm{KS}} = \sum_{\lambda_i > 0} \lambda_i.$$

In Figure 7, we compute the KS entropy over the same parameter ranges used in Section 2.6. The KS entropy remains positive across all parameter values, confirming chaos throughout the parameter range. In most cases, the KS entropy is large, corresponding to hyperchaotic behavior (two positive Lyapunov exponents), while in narrow intervals near $a = 0$, its value is smaller, indicating ordinary chaos. These results reinforce the earlier Lyapunov exponent findings and demonstrate the robustness of the chaotic dynamics.



(**a**) KS entropy vs. a (**b**) KS entropy vs. b (**c**) KS entropy vs. c (**d**) KS entropy vs. d

Figure 7. Kolmogorov–Sinai entropy as each control parameter a, b, c, and d is varied.

Table 1 compares the proposed map with three recent 3D discrete memristive hyperchaotic maps that report complete Lyapunov triples. Each reference map has two positive Lyapunov exponents, with $\lambda_1 \in [0.2437, 0.2515]$ and $\lambda_2 \in [0.0245, 0.0952]$. For the reference maps, the Kolmogorov–Sinai entropy h_{KS}, computed from the published Lyapunov exponents via the Pesin identity, lies in $[0.2682, 0.3467]$, and the Lyapunov dimension D_L remains below 2.71. The proposed map reaches $\lambda_1 = 6.3682$, $\lambda_2 = 3.1936$, $h_{\mathrm{KS}} = 9.5618$, and $D_L = 3.0000$, indicating that the attractor's fractal complexity saturates the state-space dimension. The proposed map attains larger Lyapunov exponents and Lyapunov dimension, and higher Kolmogorov–Sinai entropy, than all three compared maps. The proposed map also remains hyperchaotic across $b \in [1, 3000]$, $c \in [0.5, 10.5]$, and $d \in [0.1, 600.1]$, which supports its robustness against parameter drift in practical encryption.

Table 1. Comparison with three recent 3D discrete memristive hyperchaotic maps reporting complete Lyapunov triples, Kolmogorov–Sinai entropy, and Lyapunov dimension.

Map	Lyapunov Exponents	h_{KS}	D_L
[27]	(0.2437, 0.0245, −0.5405)	0.2682	2.4962
[28]	(0.2454, 0.0661, −0.4393)	0.3115	2.7092
[29]	(0.2515, 0.0952, −2.2751)	0.3467	2.1524
Proposed	(6.3682, 3.1936, −1.4464)	9.5618	3.0000

The compared maps and the proposed map differ in their memristive structure. Gao et al. [27] and Nabil and Tayeb [28] use a charge-controlled sine-based memristance of the form $M(q_n) = c_1 - c_2 \sin(q_n)$ coupled to pure sine-function nonlinearities. Wang et al. [29] use a current-controlled cosine memristance $M(q_n) = \cos(q_n)$, coupled with a quadratic term ay_n^2 and added with a positive sign. The proposed map instead uses a flux-controlled cosine memductance $W(z_n) = \cos(z_n)$, with the term $x_n \cos(z_n)$ appearing with opposite signs in the x and y equations, and it couples the state variables through two cross-coupled sine-logistic terms, $\sin(\pi b\, x_n(1 - y_n))$ in the x equation and $\sin(\pi d\, y_n(1 - x_n))$ in the y equation. This gives a structurally different feedback path from the compared maps.

3. Application of the Proposed Memristive Hyperchaotic System to Color Image Encryption

In this section, we introduce a novel image encryption scheme that combines the proposed memristive hyperchaotic map with DNA encoding. The input is a color image of size $M \times N \times 3$, and the following subsections describe the scheme in detail.

3.1. Parameter and Initial Condition Generation

In this subsection, we describe the method for generating the initial conditions and control parameters of the 3D memristive hyperchaotic map.

We first divide the 256-bit external secret key into 32 equal parts, each consisting of 8 bits, and convert each part into decimal form as follows:

$$S = \{s_1, s_2, s_3, \ldots, s_{32}\} \tag{4}$$

Let P be the plaintext image. By applying the SHA-256 hash algorithm to P, we obtain a 256-bit output, which is divided into 32 equal segments, each with 8 bits, and converted into decimal form:

$$V = \{v_1, v_2, v_3, \ldots, v_{32}\} \tag{5}$$

Next, by applying the bitwise XOR operation between the corresponding entries of (4) and (5), we obtain the combined sequence U, expressed as:

$$U = \{u_1, u_2, u_3, \ldots, u_{32}\}, \qquad u_i = s_i \oplus v_i \tag{6}$$

Next, using the first 12 elements of U, the initial conditions of the 3D memristive hyperchaotic map are generated as follows:

$$\begin{cases} x_0 = \dfrac{u_1 \cdot 2^{24} + u_2 \cdot 2^{16} + u_3 \cdot 2^8 + u_4}{2^{32}} \\ y_0 = \dfrac{u_5 \cdot 2^{24} + u_6 \cdot 2^{16} + u_7 \cdot 2^8 + u_8}{2^{32}} \\ z_0 = \dfrac{u_9 \cdot 2^{24} + u_{10} \cdot 2^{16} + u_{11} \cdot 2^8 + u_{12}}{2^{32}} \end{cases} \tag{7}$$

Similarly, utilizing the remaining elements of U, the control parameters of the 3D memristive hyperchaotic map are generated by the following procedure:

$$\begin{cases} a = 10 + 10 \cdot \dfrac{u_{13} \cdot 2^{32} + u_{14} \cdot 2^{24} + u_{15} \cdot 2^{16} + u_{16} \cdot 2^8 + u_{17}}{2^{40}} \\ b = 2 + 3 \cdot \dfrac{u_{18} \cdot 2^{32} + u_{19} \cdot 2^{24} + u_{20} \cdot 2^{16} + u_{21} \cdot 2^8 + u_{22}}{2^{40}} \\ c = 1 + 3 \cdot \dfrac{u_{23} \cdot 2^{32} + u_{24} \cdot 2^{24} + u_{25} \cdot 2^{16} + u_{26} \cdot 2^8 + u_{27}}{2^{40}} \\ d = 3 + 3 \cdot \dfrac{u_{28} \cdot 2^{32} + u_{29} \cdot 2^{24} + u_{30} \cdot 2^{16} + u_{31} \cdot 2^8 + u_{32}}{2^{40}} \end{cases} \tag{8}$$

The parameters generated by Equation (8) lie in the ranges $a \in [10, 20]$, $b \in [2, 5]$, $c \in [1, 4]$, and $d \in [3, 6]$. These ranges are strict subsets of the intervals examined in the bifurcation and Lyapunov analyses of Section 2.6, so every parameter set produced during encryption operates inside a verified hyperchaotic region.

Because the scheme uses SHA-256 together with the external secret key, even a small change in the plaintext image or the external key leads to a significant change in both the initial conditions and the control parameters of the 3D memristive hyperchaotic map. This substantially improves the robustness of the scheme against various cryptographic attacks. The hash V is transmitted to the receiver together with the external secret key S through the same secure channel used for key distribution. Since V depends on the plaintext, each image produces different initial conditions and control parameters, resulting in a unique keystream per image.

3.2. Keystream Generation

The encryption keystream is generated using the 3D memristive hyperchaotic map. The procedure is as follows.

- Using the initial conditions and parameters from Section 3.1, we iterate the map $7MN$ times and store the resulting values as the sequences x, y, and z, respectively.
- Each of the sequences x, y, and z is further divided into four subsequences: the first $4MN$ elements are used for the Bounce-Escape Zigzag scrambling, the next MN elements are used for the dynamic DNA rules, the following MN elements are used for the DNA keystream, and the remaining MN elements are used for the chaotic XOR operation.

- The first $4MN$ elements of each sequence are normalized to the interval $[0, 1]$ using their minimum and maximum values, and the resulting sequences are denoted as X_1, Y_1, and Z_1, respectively:

$$X_1 = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad Y_1 = \frac{y - \min(y)}{\max(y) - \min(y)}, \quad Z_1 = \frac{z - \min(z)}{\max(z) - \min(z)}.$$

  Here X_1, Y_1, and Z_1 are one-dimensional sequences of length $4MN$.
- The next MN elements of each sequence are used to determine the dynamic DNA coding rules. For each element, the corresponding rule index R_k is computed as

$$R_k = \left(\left\lfloor |c_k| \cdot 10^{15} \right\rfloor \bmod 8 \right) + 1.$$

  The resulting values are then reshaped into $M \times N$ matrices denoted X_2, Y_2, and Z_2, respectively.
- The following MN elements of each sequence generate the DNA keystream. For each element, the corresponding key value K_k is computed as

$$K_k = \left\lfloor |c_k| \cdot 10^{15} \right\rfloor \bmod 256.$$

  Reshape the resulting values into $M \times N$ matrices and denote them as X_3, Y_3, and Z_3, respectively.
- Use the remaining MN elements to generate the final XOR mask in the same way. The mask value W_k is obtained using the same operation as above, and the resulting values are reshaped into $M \times N$ matrices, denoted X_4, Y_4, and Z_4, respectively.

The sequences X_1, Y_1, Z_1 (of length $4MN$) and the matrices X_2, Y_2, Z_2, ..., X_4, Y_4, Z_4 (of size $M \times N$) are then used in the subsequent scrambling and diffusion stages of the proposed encryption algorithm.

3.3. Bounce-Escape Zigzag (BEZ) Algorithm

Existing image encryption schemes typically rely on either a classical zigzag pattern, whose traversal order is regular and predictable [23], or a block-based dynamic zigzag [24]. The block-based dynamic zigzag improves randomness by using a chaotic starting point and randomly changing its direction after each diagonal move; however, it still uses a constant step size of one pixel, only two diagonal directions, and jumps to an arbitrary unvisited pixel whenever the walk is blocked.

To overcome these drawbacks, the proposed Bounce-Escape Zigzag (BEZ) algorithm performs a global, variable-length diagonal traversal over the entire $M \times N$ matrix. Starting from a key-dependent initial position, the walk proceeds along one of four diagonal directions. Before each sweep, a normalized chaotic value determines how many successive diagonal moves to take, making the run length unpredictable. When the walk reaches a matrix boundary, it bounces by reflecting the corresponding direction component instead of stopping. If both coordinates are simultaneously out of bounds, both are reversed; if only one coordinate is out of bounds, a second chaotic value may also flip the other component, producing a pinball-like reflection. If the next element has already been visited, the algorithm performs a collision escape. It first attempts to move to an unvisited cell in its 8-neighborhood, and only if none is available does it resort to a global jump to any remaining unvisited position. After each escape, the traversal direction is updated from the movement vector so that the walk continues along a diagonal; when the escape is purely horizontal or vertical, the other component is set from a chaotic value to restore a diagonal direction. This procedure continues until all elements of the matrix are visited exactly

once. The resulting traversal order is then used to rearrange the matrix entries, thereby destroying spatial correlation. The complete steps are given in Algorithm 1.

Algorithm 1 Bounce-Escape Zigzag (BEZ) Scrambling for an $M \times N$ Matrix.

Require: Input matrix A of size $M \times N$ (total elements $L = MN$); normalized chaotic sequence C of length $4L$
Ensure: Scrambled matrix B

$V \leftarrow \text{false}(M, N)$; $B_{\text{vec}} \leftarrow \text{zeros}(L, 1)$; $count \leftarrow 1$
$\hat{i}_0 \leftarrow 1 + \lfloor C(1)M \rfloor$; $\hat{j}_0 \leftarrow 1 + \lfloor C(2)N \rfloor$
$i_0 \leftarrow \min(\max(\hat{i}_0, 1), M)$; $j_0 \leftarrow \min(\max(\hat{j}_0, 1), N)$
$V(i_0, j_0) \leftarrow$ true; $B_{\text{vec}}(1) \leftarrow A(i_0, j_0)$; $p \leftarrow 3$
Initial direction:
if $C(3) < 0.25$ **then**
    $(d_i, d_j) \leftarrow (-1, 1)$
else if $C(3) < 0.5$ **then**
    $(d_i, d_j) \leftarrow (-1, -1)$
else if $C(3) < 0.75$ **then**
    $(d_i, d_j) \leftarrow (1, -1)$
else
    $(d_i, d_j) \leftarrow (1, 1)$
end if
$p \leftarrow p + 1$
while $count < L$ **do**
    $L_r \leftarrow 1 + \lfloor C(p) \min(M, N) \rfloor$ ▷ Variable run-length
    $p \leftarrow p + 1$
    for $s = 1$ **to** L_r **do**
        $(i_n, j_n) \leftarrow (i_0 + d_i,\ j_0 + d_j)$; $out_i \leftarrow (i_n \notin [1, M])$; $out_j \leftarrow (j_n \notin [1, N])$
        while out_i **or** out_j **do** ▷ Boundary bounce
            if out_i **then** $d_i \leftarrow -d_i$
            if out_j **then** $d_j \leftarrow -d_j$
            if $out_i \neq out_j$ **then** ▷ Exactly one component out
                $b \leftarrow C(p)$; $p \leftarrow p + 1$
                if $b \geq 0.5$ **then if** out_i **then** $d_j \leftarrow -d_j$ **else** $d_i \leftarrow -d_i$
            end if
            $(i_n, j_n) \leftarrow (i_0 + d_i,\ j_0 + d_j)$; $out_i \leftarrow (i_n \notin [1, M])$; $out_j \leftarrow (j_n \notin [1, N])$
        end while
        if $\neg V(i_n, j_n)$ **then** ▷ Successful move
            $(i_0, j_0) \leftarrow (i_n, j_n)$; $V(i_0, j_0) \leftarrow$ true; $count \leftarrow count + 1$; $B_{\text{vec}}(count) \leftarrow A(i_0, j_0)$
        else ▷ Collision escape
            $(i_{\text{prev}}, j_{\text{prev}}) \leftarrow (i_0, j_0)$; $\mathcal{U} \leftarrow$ unvisited 8-neighbors of (i_0, j_0) ▷ enumerated in row-major order
            if $\mathcal{U} \neq \emptyset$ **then**
                $idx \leftarrow \min\big(1 + \lfloor C(p)|\mathcal{U}| \rfloor,\ |\mathcal{U}|\big)$; $(i_0, j_0) \leftarrow \mathcal{U}(idx)$
            else
                $\mathcal{G} \leftarrow$ all unvisited indices ▷ enumerated in column-major order
                $idx \leftarrow \min\big(1 + \lfloor C(p)|\mathcal{G}| \rfloor,\ |\mathcal{G}|\big)$; $(i_0, j_0) \leftarrow \mathcal{G}(idx)$
            end if
            $p \leftarrow p + 1$; $V(i_0, j_0) \leftarrow$ true; $count \leftarrow count + 1$; $B_{\text{vec}}(count) \leftarrow A(i_0, j_0)$
            Update direction: $\mathbf{v} \leftarrow (i_0 - i_{\text{prev}}, j_0 - j_{\text{prev}})$
            if $\mathbf{v}(1) \neq 0$ **and** $\mathbf{v}(2) \neq 0$ **then**
                $(d_i, d_j) \leftarrow (\text{sgn}(\mathbf{v}(1)), \text{sgn}(\mathbf{v}(2)))$
            else if $\mathbf{v}(1) = 0$ **then** ▷ Horizontal escape
                $d_j \leftarrow \text{sgn}(\mathbf{v}(2))$; $r \leftarrow C(p)$; $p \leftarrow p + 1$; $d_i \leftarrow 2(r \geq 0.5) - 1$
            else ▷ Vertical escape
                $d_i \leftarrow \text{sgn}(\mathbf{v}(1))$; $r \leftarrow C(p)$; $p \leftarrow p + 1$; $d_j \leftarrow 2(r \geq 0.5) - 1$
            end if
            break
        end if
    end for
end while
$B \leftarrow \text{reshape}(B_{\text{vec}}, M, N)$
return B

The proposed BEZ traversal terminates after visiting every element of the $M \times N$ matrix exactly once. At initialization, exactly one cell is marked visited and $count = 1$. Every subsequent cell selection either moves to an unvisited neighboring cell or performs a collision escape, and in both cases the selected cell is unvisited, so it is marked visited and $count$ increases by one. Since no cell is ever selected twice, $count$ reaches $L = MN$ after exactly L cell selections, which proves termination and one-to-one coverage for arbitrary M and N. The boundary-bounce loop terminates in practice after a small number of iterations,

since each iteration flips a direction component toward the matrix interior. The unvisited 8-neighbors $\mathcal{U}$ and the global unvisited set $\mathcal{G}$ are enumerated in fixed orders (row-major and column-major, respectively), so the traversal is fully deterministic.

3.4. DNA Encoding and Operations

Deoxyribonucleic acid (DNA) sequences consist of four nucleic acid bases: adenine (A), cytosine (C), guanine (G), and thymine (T). In the DNA-based image encryption algorithm, each 8-bit pixel value is expressed as four 2-bit pairs, and every 2-bit pair is transformed into one DNA base according to the chosen DNA rule. Table 2 presents the 8 Watson-Crick encoding rules. In the proposed scheme, the chaotic keystream adaptively selects the rule index for each pixel.

Table 2. The 8 DNA encoding rules.

Pair\Rule	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
00	A	A	C	C	G	G	T	T
01	C	G	A	T	A	T	C	G
10	G	C	T	A	T	A	G	C
11	T	T	G	G	C	C	A	A

We perform DNA operations on the DNA bases after mapping the 2-bit binary values to their corresponding bases. In this work, we apply four operations using fixed operation tables: addition, subtraction, XNOR, and XOR. For Rule 1, Table 3 defines these operations; the same tables apply to all other rules. In the proposed scheme, different operations are applied to the four DNA bases of each pixel. Since the encoding and decoding rules are selected dynamically per pixel, the overall DNA process is pixel-dependent.

Table 3. DNA operation tables based on Rule 1.

(a) Addition				
$+$	A	C	G	T
A	A	C	G	T
C	C	G	T	A
G	G	T	A	C
T	T	A	C	G
(b) Subtraction				
$-$	A	C	G	T
A	A	T	G	C
C	C	A	T	G
G	G	C	A	T
T	T	G	C	A
(c) XNOR ($\odot$)				
$\odot$	A	C	G	T
A	T	G	C	A
C	G	T	A	C
G	C	A	T	G
T	A	C	G	T
(d) XOR ($\oplus$)				
$\oplus$	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

3.5. Proposed Image Encryption Scheme Process

This subsection explains the proposed image encryption scheme in detail. The main steps are stated as follows:

Step 1: Read the plaintext color image of size $M \times N \times 3$ and divide it into three separate channels P_r, P_g, and P_b.

Step 2: Using the 256-bit external secret key and the SHA-256 hash of the plaintext image, follow the procedure described in Section 3.1 to generate the initial conditions and control parameters of the 3D memristive hyperchaotic map.

Step 3: Using the generated initial conditions and parameters, produce the keystreams $X_1, Y_1, Z_1, \ldots, X_4, Y_4, Z_4$ according to the procedure described in Section 3.2.

Step 4: Using the sequences X_1, Y_1, and Z_1, scramble each channel P_r, P_g, and P_b by applying the BEZ algorithm described in Section 3.3 (Algorithm 1). This produces the scrambled channels P_{1r}, P_{1g}, and P_{1b}, respectively:

$$\begin{cases} P_{1r} = \mathrm{BEZ}(P_r,\ X_1), \\ P_{1g} = \mathrm{BEZ}(P_g,\ Y_1), \\ P_{1b} = \mathrm{BEZ}(P_b,\ Z_1). \end{cases} \tag{9}$$

Step 5: Convert each scrambled channel P_{1r}, P_{1g}, and P_{1b} into its DNA-encoded form using the dynamic DNA rule matrices X_2, Y_2, and Z_2, respectively. Similarly, convert the DNA key matrices X_3, Y_3, and Z_3 using the same rule matrices X_2, Y_2, and Z_2, respectively. This encoding can be expressed as

$$\begin{cases} P_{1r}^{DNA} = \mathrm{DNA_encode}\big(P_{1r},\ X_2\big), \\ P_{1g}^{DNA} = \mathrm{DNA_encode}\big(P_{1g},\ Y_2\big), \\ P_{1b}^{DNA} = \mathrm{DNA_encode}\big(P_{1b},\ Z_2\big), \\ X_3^{DNA} = \mathrm{DNA_encode}\big(X_3,\ X_2\big), \\ Y_3^{DNA} = \mathrm{DNA_encode}\big(Y_3,\ Y_2\big), \\ Z_3^{DNA} = \mathrm{DNA_encode}\big(Z_3,\ Z_2\big). \end{cases} \tag{10}$$

Here P_{1r}^{DNA}, P_{1g}^{DNA}, P_{1b}^{DNA} denote the DNA-encoded scrambled channels, and X_3^{DNA}, Y_3^{DNA}, and Z_3^{DNA} denote the DNA-encoded key matrices.

Step 6: For each DNA-encoded scrambled channel P_{1r}^{DNA}, P_{1g}^{DNA}, and P_{1b}^{DNA} and the corresponding DNA-encoded key matrices X_3^{DNA}, Y_3^{DNA}, and Z_3^{DNA}, the four DNA bases of every pixel are processed using four different DNA operations as defined in Table 3. For a given pixel, let L_1, L_2, L_3, L_4 denote the four DNA bases of the scrambled pixel, and let K_1, K_2, K_3, K_4 denote the four DNA bases of the corresponding key pixel. The operations are performed using fixed DNA operation tables:

$$\begin{cases} F_1 = \mathrm{DNA}_{\mathrm{add}}\big(L_1, K_1\big), \\ F_2 = \mathrm{DNA}_{\mathrm{sub}}\big(L_2, K_2\big), \\ F_3 = \mathrm{DNA}_{\mathrm{xnor}}\big(L_3, K_3\big), \\ F_4 = \mathrm{DNA}_{\mathrm{xor}}\big(L_4, K_4\big), \end{cases} \tag{11}$$

where F_1, F_2, F_3, F_4 are the resulting DNA bases. Since the DNA encoding rule changes from pixel to pixel, the encoded bases vary, and therefore the operation

results also vary, making the overall DNA process dynamic. Repeat the above operations independently for every pixel in every color channel.

Step 7: Decode the four resulting DNA bases F_1, F_2, F_3, F_4 back to an 8-bit binary value using the same dynamic DNA rule matrix that was used for encoding. This produces the DNA-diffused channels P_{2r}, P_{2g}, P_{2b}.

Step 8: In the final step, apply the chaotic XOR mask to each DNA-diffused channel. Explicitly, the channels P_{2r}, P_{2g}, P_{2b} are XORed with the mask matrices X_4, Y_4, Z_4, respectively:

$$\begin{cases} P_{3r} = P_{2r} \oplus X_4, \\ P_{3g} = P_{2g} \oplus Y_4, \\ P_{3b} = P_{2b} \oplus Z_4, \end{cases} \tag{12}$$

where $\oplus$ denotes the bitwise XOR operation. The generated matrices P_{3r}, P_{3g}, P_{3b} are the final encrypted channels, which are then merged to form the cipher image P_4 of size $M \times N \times 3$.

Since all operations in the proposed encryption scheme are reversible, decryption is the inverse of encryption. The receiver receives the ciphertext P_4, the external secret key S, and the hash V from the secure channel, computes $U = S \oplus V$ via Equation (6), and uses the initial conditions and parameters from Equations (7) and (8) to regenerate the keystreams. Using the same secret key and the same generated keystreams, we first apply the inverse XOR mask. The DNA decoding then uses the same dynamic rule matrices X_2, Y_2, and Z_2 that were used during encryption, and the inverse DNA operations are performed using the same fixed operation tables (addition reversed by subtraction, subtraction reversed by addition, XOR and XNOR reversed by themselves). Finally, the inverse BEZ permutation is performed to recover the original plaintext channels P_r, P_g, P_b.

4. Scheme Performance and Security Analysis

In this section, we analyze the performance and security of the proposed scheme through various tests. We use six test images of different resolutions: Peppers, House, Baboon, and Sailboat ($512 \times 512 \times 3$), Military ($256 \times 256 \times 3$), and Medical ($1024 \times 1024 \times 3$). The experiments are conducted in MATLAB 2024 on a personal computer with an Intel Core i5-10310U CPU @ 1.70 GHz (up to 2.21 GHz), 16 GB of RAM, and a 512 GB SSD, running Windows 11 (64-bit). Throughout the article, we use the external secret key $S =$ d00eea a94216484373e5719ff011ac565ea37b9e18cfe30db601d2bbefbb19f9. The initial conditions and parameters of the 3D memristive hyperchaotic map are generated according to the procedure described in Section 3.1.

4.1. Encryption and Decryption Experiments

This subsection presents the encryption and decryption results for six test images: Peppers, House, Baboon, Sailboat, Military, and Medical. Figure 8 shows the encryption and decryption outcomes. The first row shows the original plaintext images; the second row shows the corresponding encrypted images; and the third row shows the decrypted images. The encrypted images in the second row are noise-like and show no visual relationship to the original images, demonstrating the strong confusion and diffusion capabilities of the proposed scheme. The images in the last row match the corresponding original images, confirming that the proposed encryption scheme is fully reversible.

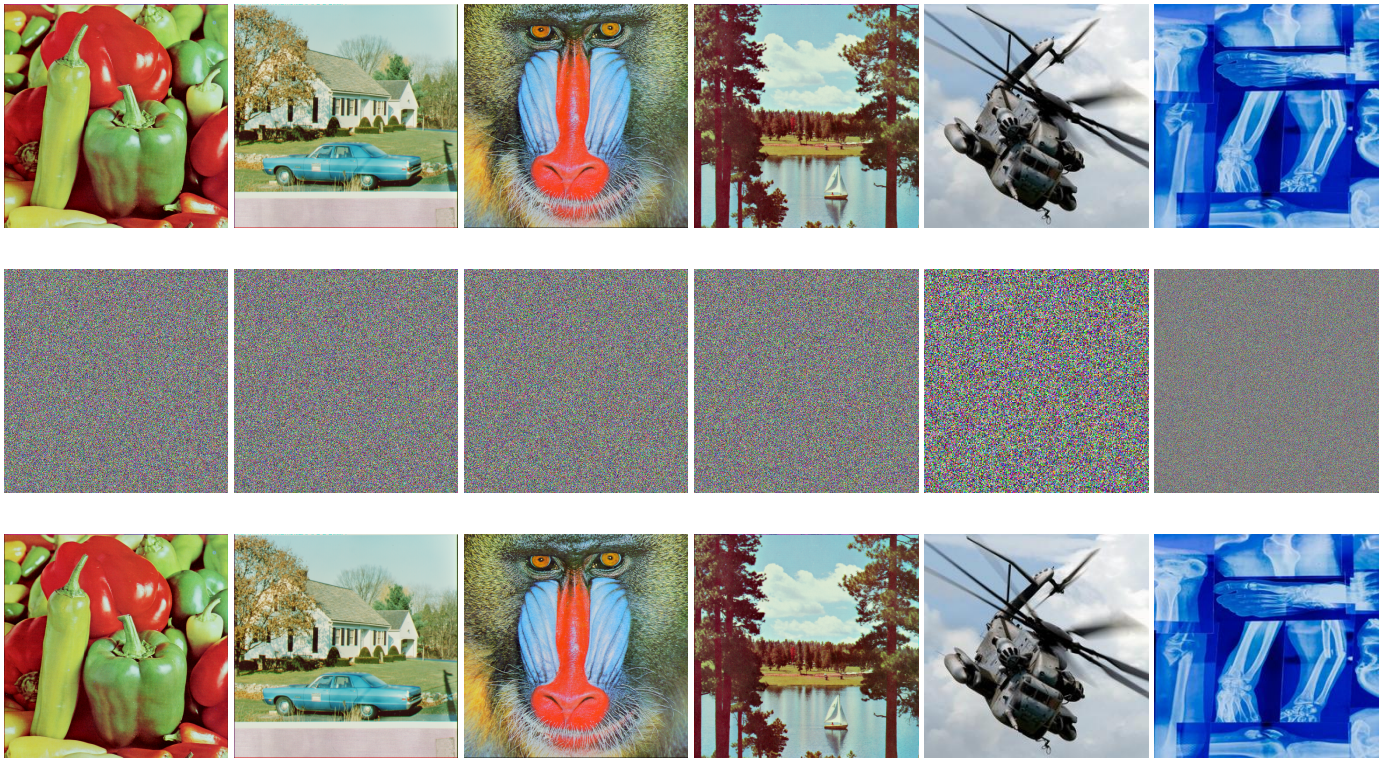


Figure 8. Encryption and decryption results for six test images. Top row: original images; middle row: encrypted images; bottom row: decrypted images. Columns from left to right: Peppers, House, Baboon, Sailboat, Military, Medical.

4.2. Histogram Analysis

Histogram analysis visually evaluates the intensity distribution of each channel's pixels. For a robust encryption scheme, the pixel distribution in the cipher image should be uniform. Figure 9 presents the histogram analysis of the proposed scheme. The first row shows the histograms of the original images, which have sharp peaks indicating patterns and redundancy in the plain data. The second row includes the corresponding encrypted images, where the data are uniformly dispersed, hinting that an attacker cannot obtain any useful statistical information to break the encryption. The last row shows histograms of the corresponding decrypted images, which match the original histograms, confirming that the scheme is reversible and decryption is lossless.

4.3. Chi-Square Test

The chi-square (χ^2) test evaluates the uniformity of pixel values in an encrypted image. For an 8-bit image, there are 256 possible intensity levels, and therefore the histogram has 255 degrees of freedom. At a significance level of $\alpha = 0.05$, the corresponding critical value is approximately 293.2478. If the computed χ^2 value for each color channel is less than the critical value, the histogram is considered uniformly distributed, and the test is passed; otherwise, the test is failed.

Table 4 lists the chi-square values for the three color channels of all tested images. From the table, it is clear that the chi-square values for every channel are strictly below the critical threshold, and therefore all tests are passed. These results indicate that the proposed scheme successfully removes statistical redundancy from the plaintext and is robust against histogram-based attacks.

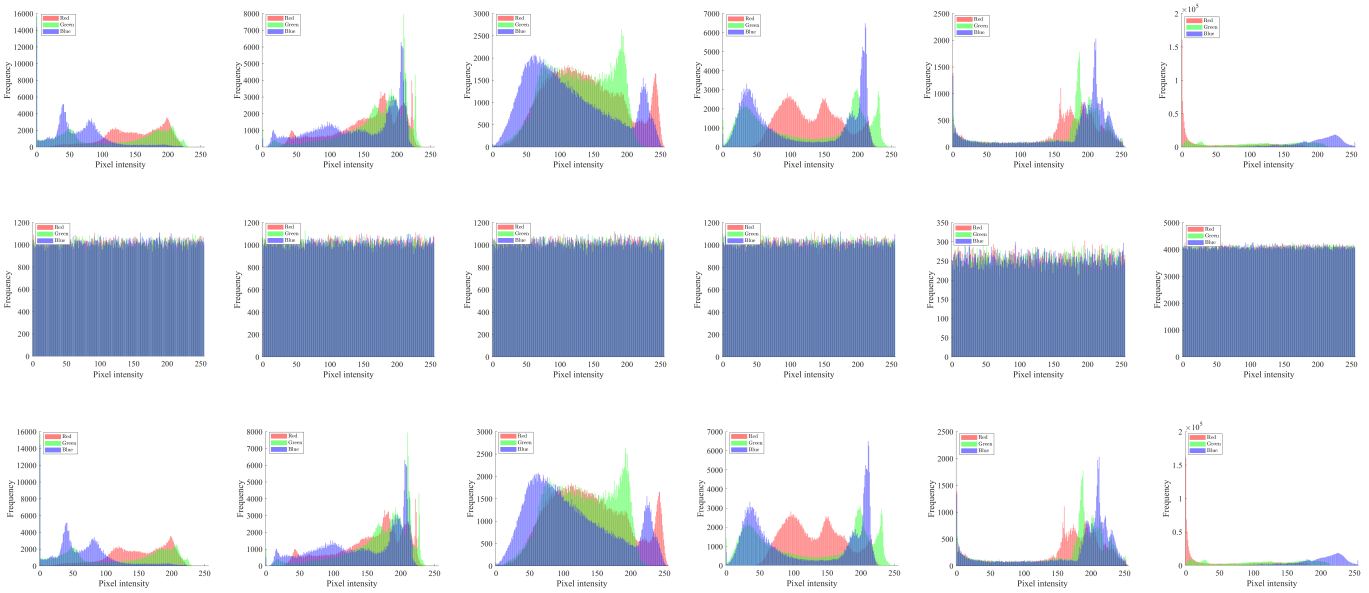


Figure 9. Histogram analysis of the six test images. Top row: original images; middle row: encrypted images; bottom row: decrypted images. Columns from left to right: Peppers, House, Baboon, Sailboat, Military, Medical.

Table 4. Chi-square (χ^2) values of the encrypted test images.

Image	Red Channel	Green Channel	Blue Channel	Result
Peppers	215.3750	243.8926	223.4629	Pass
House	259.0762	236.6309	263.4062	Pass
Baboon	256.9473	257.8789	268.7520	Pass
Sailboat	241.6133	291.6191	237.9238	Pass
Military	252.5938	273.2656	238.5859	Pass
Medical	227.3145	241.7451	246.8232	Pass

4.4. Information Entropy Analysis

Information entropy measures how uniformly the pixel values are distributed. For an 8-bit image, the ideal value is 8. Table 5 lists the entropy of the original and encrypted images. For every cipher image, the entropy is very close to 8, so the encrypted pixels are nearly uniformly distributed. Table 6 compares the entropy of the Peppers image with recent schemes, and the proposed method achieves superior values.

Table 5. Information entropy of original and encrypted test images.

Image	Channel	Original Entropy	Cipher Entropy
Peppers	R	7.3389	7.9994
	G	7.4963	7.9993
	B	7.0584	7.9994
House	R	7.4156	7.9993
	G	7.2295	7.9994
	B	7.4354	7.9993
Baboon	R	7.7067	7.9993
	G	7.4744	7.9993
	B	7.7522	7.9993

Table 5. *Cont.*

Image	Channel	Original Entropy	Cipher Entropy
Sailboat	R	7.3124	7.9993
	G	7.6429	7.9992
	B	7.2137	7.9993
Military	R	7.4954	7.9972
	G	7.3061	7.9970
	B	7.1992	7.9974
Medical	R	6.5093	7.9998
	G	7.6611	7.9998
	B	6.9173	7.9998

Table 6. Comparison of information entropy on the Peppers image with recent schemes.

Scheme	Entropy			Average
	R	G	B	
Proposed	7.9994	7.9993	7.9994	7.9994
[30]	7.9993	7.9993	7.9993	7.9993
[31]	7.9993	7.9993	7.9992	7.9993
[32]	7.9994	7.9993	7.9993	7.9993
[33]	7.9993	7.9993	7.9992	7.9993
[34]	7.9993	7.9993	7.9993	7.9993

4.5. Mean Squared Error Analysis

Mean squared error (MSE) measures the average squared difference between two images:

$$\mathrm{MSE} = \frac{1}{3MN} \sum_{c=1}^{3} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j,c) - C(i,j,c))^2 \tag{13}$$

where P denotes the plaintext image, C is the corresponding cipher or decrypted image, and M and N are the image height and width. A high MSE between the plaintext and ciphertext images indicates that the ciphertext differs substantially from the plaintext, which is a sign of strong confusion and diffusion. A zero MSE between the plaintext and decrypted images indicates perfect recovery of the original image.

Table 7 reports the MSE values between the plaintext and ciphertext images and between the plaintext and decrypted images for all six test images. In every case, the plaintext–ciphertext MSE is high, confirming that the ciphertext is completely different from the plaintext, while the plaintext-decrypted MSE is zero, confirming lossless recovery. Table 8 compares the plaintext–ciphertext MSE of the proposed scheme with recent schemes on four standard images; the proposed values are higher in every case.

Table 7. Mean squared error (MSE) between the original and cipher images, and between the original and decrypted images.

Image	MSE (Cipher vs. Original)	MSE (Decrypted vs. Original)
Peppers	10,133.8746	0
House	9234.4729	0
Baboon	8620.3278	0
Sailboat	10,121.9187	0
Military	11,623.8112	0
Medical	11,993.4648	0

Table 8. Comparison of MSE with recent schemes.

Scheme	Peppers	House	Baboon	Sailboat
Proposed	10,133.8746	9234.4729	8620.3278	10,121.9187
[26]	10,131.2238	–	8619.5059	10,102.2430
[35]	10,109.4367	–	–	–
[36]	–	–	7984.0000	–

4.6. Peak Signal-to-Noise Ratio Analysis

Peak signal-to-noise ratio (PSNR) is another quantitative metric for measuring the difference between two images. It is defined from the MSE as

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right).$$

A lower PSNR indicates higher dissimilarity between the two images, while a higher value indicates lower dissimilarity. In the literature, PSNR values below 10 dB are considered to indicate strong encryption schemes, while an infinite value indicates perfect recovery of the plaintext. Table 9 reports the PSNR values between the plaintext and ciphertext images and between the plaintext and decrypted images for all six test images. In every case, the plaintext–ciphertext PSNR is below 9 dB, confirming strong encryption, while the plaintext-decrypted PSNR is infinite, confirming lossless recovery.

Table 9. Peak signal-to-noise ratio (PSNR) between the original and cipher images, and between the original and decrypted images.

Image	Cipher PSNR (dB)	Decrypted PSNR (dB)
Peppers	8.0730	∞
House	8.4767	∞
Baboon	8.7756	∞
Sailboat	8.0782	∞
Military	7.4773	∞
Medical	7.3414	∞

4.7. Structural Similarity Index Analysis

Structural similarity index (SSIM) is a visual quality metric that quantifies the similarity between two images by considering their brightness, contrast, and structural information. SSIM takes values in the interval $[0, 1]$, where a value close to zero indicates strong dissimilarity, and a value of 1 indicates that the two images are identical.

Table 10 reports the SSIM values between the plaintext and ciphertext images and between the plaintext and decrypted images for all six test images. The plaintext–ciphertext SSIM is close to zero in every case, confirming that the ciphertext is completely different from the plaintext. The plaintext-decrypted SSIM is exactly 1, confirming lossless recovery.

Table 10. Structural similarity index (SSIM) between the plaintext and cipher images, and between the plaintext and decrypted images.

Image	Cipher SSIM	Decrypted SSIM
Peppers	0.0089	1.0000
House	0.0099	1.0000
Baboon	0.0089	1.0000
Sailboat	0.0091	1.0000
Military	0.0082	1.0000
Medical	0.0080	1.0000

4.8. Correlation Analysis

For an image, the correlation analysis is a very useful test to measure the correlation between adjacent pixels in the horizontal, vertical, and diagonal directions. The value of the correlation lies in the interval $[-1, 1]$. A value close to zero indicates that there is no correlation between the pixels, and a value close to 1 or -1 indicates that they are highly correlated with each other. For secure image encryption, the spatial correlation between the adjacent pixels is very close to zero. The formula for calculating the correlation coefficient is defined as follows:

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{14}$$

where x_i and y_i are adjacent pixel values, $\bar{x}$ and $\bar{y}$ are the means of adjacent pixel sequences, and N is the total number of samples in the sequences. In Figure 10, the correlation analyses for the Peppers and House images are shown for both the encrypted and original images. For the original images, the pixels are highly clustered around the diagonal, which shows that the pixels have strong spatial correlation; on the other hand, for the encrypted images, pixels are uniformly distributed in the plane, confirming that the proposed scheme breaks the spatial correlation. The correlation coefficient values for all the tested images are listed in Table 11. For the plaintext images, the correlation coefficient value is close to 1, indicating that the pixels are highly correlated with each other. In the case of the encrypted images, the values approach zero, which illustrates that the adjacent pixels have no correlation, demonstrating the resilience of the proposed scheme against statistical attacks.

Table 11. Correlation coefficients of plaintext and ciphertext images.

Test Image	Direction	Plaintext Image			Ciphertext Image		
		R	G	B	R	G	B
Peppers	H	0.9636	0.9780	0.9638	−0.0031	0.0096	0.0015
	V	0.9641	0.9761	0.9637	−0.0053	0.0013	0.0030
	D	0.9550	0.9600	0.9405	−0.0004	−0.0043	−0.0041
House	H	0.9538	0.9398	0.9710	−0.0050	−0.0051	−0.0092
	V	0.9576	0.9254	0.9578	−0.0011	−0.0016	−0.0002
	D	0.9195	0.8736	0.9331	−0.0003	−0.0019	0.0057
Baboon	H	0.9240	0.8692	0.9093	−0.0012	−0.0009	0.0026
	V	0.8624	0.7699	0.8827	−0.0075	−0.0070	−0.0093
	D	0.8490	0.7404	0.8402	0.0053	0.0052	0.0059
Sailboat	H	0.9534	0.9679	0.9689	0.0006	0.0027	−0.0021
	V	0.9522	0.9625	0.9688	0.0067	0.0053	0.0002
	D	0.9372	0.9467	0.9486	−0.0030	−0.0037	−0.0055
Military	H	0.9967	0.9967	0.9929	−0.0020	0.0057	0.0021
	V	0.9972	0.9974	0.9947	0.0031	−0.0085	−0.0021
	D	0.9937	0.9940	0.9874	0.0109	−0.0016	−0.0063
Medical	H	0.9668	0.9648	0.9711	0.0037	−0.0024	0.0034
	V	0.9590	0.9616	0.9619	−0.0052	0.0053	0.0083
	D	0.9563	0.9568	0.9602	−0.0042	−0.0042	−0.0088

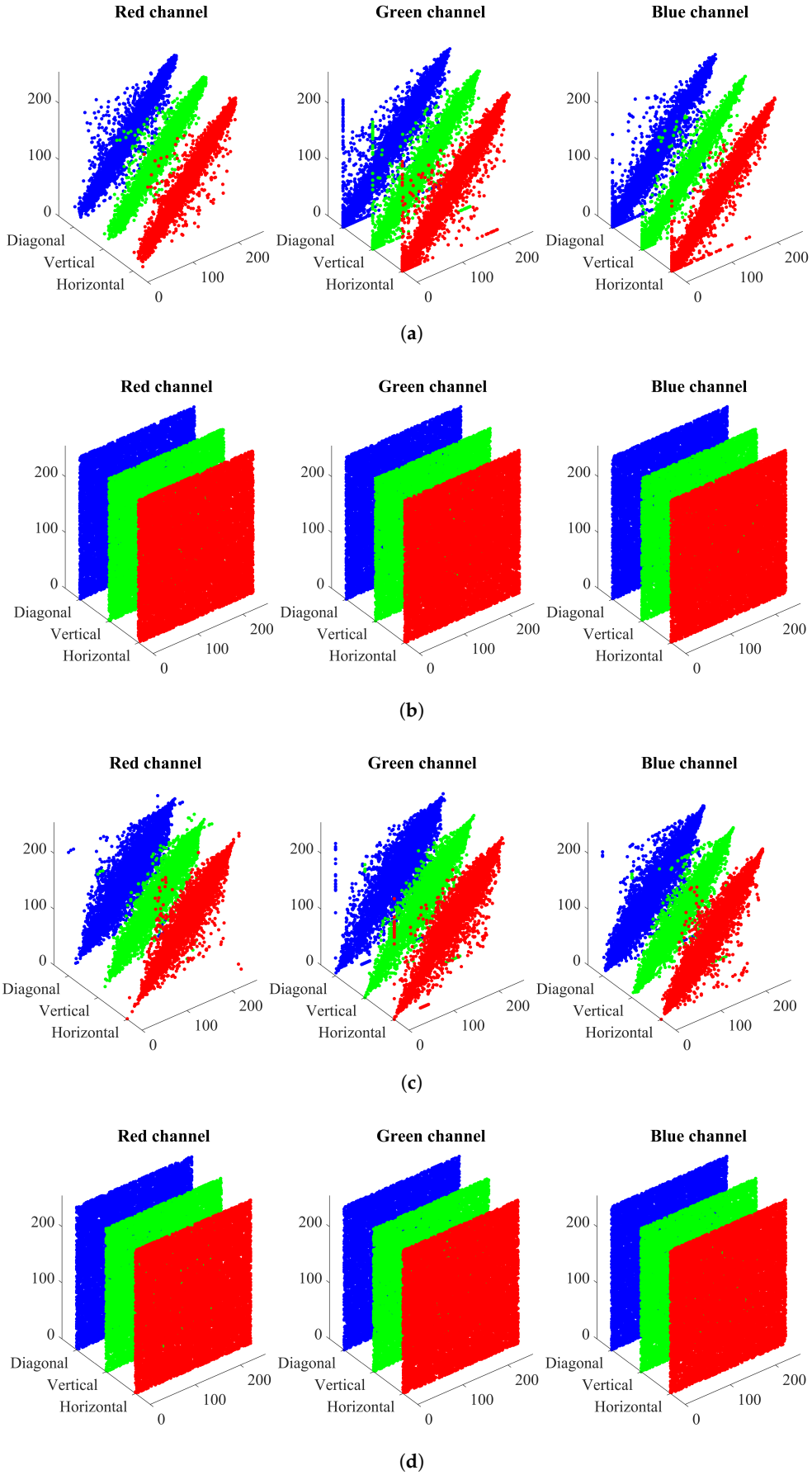


Figure 10. Correlation analysis of original and encrypted images. (**a**) Original Peppers, (**b**) Encrypted Peppers, (**c**) Original House, (**d**) Encrypted House.

4.9. Differential Attack Analysis

Differential attack analysis measures how a single-bit change in the plaintext affects the cipher image. A secure encryption scheme should produce completely different ciphertexts for plaintexts that differ by one bit. Two standard metrics quantify this sensitivity: the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). They are defined as follows:

$$\begin{cases} \text{NPCR} = \dfrac{1}{M \times N} \displaystyle\sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \\ D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \end{cases} \tag{15}$$

$$\text{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{16}$$

where M and N are the height and width of the images, and C_1 and C_2 are the two cipher images obtained from two plaintext images differing by exactly one bit. In Table 12, the NPCR and UACI values are listed. The values are very close to the ideal values, indicating the robustness of the scheme. Table 13 compares the NPCR and UACI values with recent schemes on the Peppers image, and the proposed scheme achieves higher average NPCR and UACI values than the compared schemes.

Table 12. Per-channel NPCR (%) and UACI (%) for the proposed scheme.

Channel	Metric	Peppers	House	Baboon	Sailboat	Military	Medical
R	NPCR	99.6216	99.6250	99.6239	99.6029	99.5834	99.6124
	UACI	33.4471	33.4226	33.4803	33.4849	33.6382	33.4522
G	NPCR	99.6059	99.6052	99.5941	99.5819	99.6231	99.6086
	UACI	33.4798	33.4291	33.4638	33.4315	33.5057	33.5091
B	NPCR	99.6143	99.6094	99.5857	99.6033	99.6414	99.6047
	UACI	33.5007	33.4818	33.3641	33.5043	33.4454	33.4472
Avg	NPCR	99.6140	99.6132	99.6012	99.5960	99.6160	99.6086
	UACI	33.4758	33.4445	33.4361	33.4736	33.5298	33.4695

Table 13. Comparison of NPCR and UACI on the Peppers image (512×512×3) with recent schemes.

Scheme	NPCR (%)				UACI (%)			
	R	G	B	Avg	R	G	B	Avg
Proposed	99.6216	99.6059	99.6143	99.6140	33.4471	33.4798	33.5007	33.4758
[33]	99.6078	99.6078	99.6006	99.6054	33.5078	33.4365	33.4673	33.4705
[34]	99.6106	99.6097	99.6099	99.6101	33.4861	33.4523	33.4572	33.4652
[37]	99.6076	99.6106	99.6092	99.6091	33.4663	33.4646	33.4796	33.4702
[38]	99.5975	99.6052	99.6037	99.6021	29.0588	33.4382	33.4001	31.9657

4.10. Key Sensitivity Analysis

A secure image encryption scheme must be highly sensitive to its secret key. To assess the key sensitivity of the proposed scheme, we encrypted the same plaintext image using the original key and a modified key that differs by one bit (we changed the external key S by replacing the first hexadecimal digit `d` with `f`; in binary, `d` = `1101` and `f` = `1111`, so only one bit differs). We then compared the two ciphertexts using the NPCR and UACI metrics. Table 14 lists the NPCR and UACI values for all six tested images. The computed values

are very close to the ideal values for NPCR and UACI, showing that the two ciphertexts differ significantly and confirming the robustness of the scheme against key-based attacks.

Table 14. Key sensitivity analysis: NPCR (%) and UACI (%) between ciphertexts from original and one-bit-changed keys.

Channel	Metric	Peppers	House	Baboon	Sailboat	Military	Medical
R	NPCR	99.6208	99.5998	99.6361	99.5975	99.5941	99.6132
	UACI	33.4781	33.4450	33.4715	33.4952	33.4637	33.4484
G	NPCR	99.6063	99.5983	99.6044	99.6014	99.6292	99.6020
	UACI	33.3968	33.4617	33.4547	33.3775	33.5943	33.4978
B	NPCR	99.6059	99.6071	99.5949	99.5911	99.5880	99.6056
	UACI	33.4427	33.5600	33.4572	33.4808	33.6333	33.4710
Avg	NPCR	99.6110	99.6017	99.6118	99.5967	99.6038	99.6069
	UACI	33.4392	33.4889	33.4611	33.4512	33.5638	33.4724

We examined the sensitivity of the seven initial conditions and control parameters for the House image under three conditions: the original inputs, a one-bit plaintext change, and a one-bit key change. The plaintext change flips the least significant bit of the red channel at position $(1,1)$; the key change replaces `d` with `f`, matching the modification used in the key sensitivity analysis. Table 15 reports the resulting values.

Table 15. Initial conditions and control parameters of the House image under original conditions, a one-bit plaintext change, and a one-bit key change. Values are rounded to five decimal places.

Case	x_0	y_0	z_0	a	b	c	d
Original	0.15929	0.77392	0.25805	17.98141	3.99163	2.57715	4.04622
Plaintext LSB flipped	0.71436	0.12977	0.08354	18.41972	2.75487	2.77198	4.73364
Key `d` $\rightarrow$ `f`	0.03429	0.77392	0.25805	17.98141	3.99163	2.57715	4.04622

In the plaintext case, all seven values change because the SHA-256 avalanche effect produces a completely different hash. In the key case, only x_0 shifts from 0.15929 to 0.03429, because the first key byte feeds only into x_0. This single change is sufficient to alter the entire chaotic trajectory, and the NPCR and UACI values in Table 14 remain close to their ideal values, showing that nearly every ciphertext pixel differs.

4.11. Key Space Analysis

The key space of the proposed scheme is 2^{256}. The keystreams used for the scrambling and diffusion stages are determined by $U = S \oplus V$, where S is the external 256-bit secret key and V is the cryptographic hash of the plaintext. The hash V is transmitted with S over the secure channel, and is unavailable to an adversary who only observes the ciphertext. The adversary must therefore guess U without knowing S. Because S is a uniformly random 256-bit value, U is likewise uniformly distributed over 2^{256} possible values, which is above the 2^{128} threshold required for security against brute-force attacks.

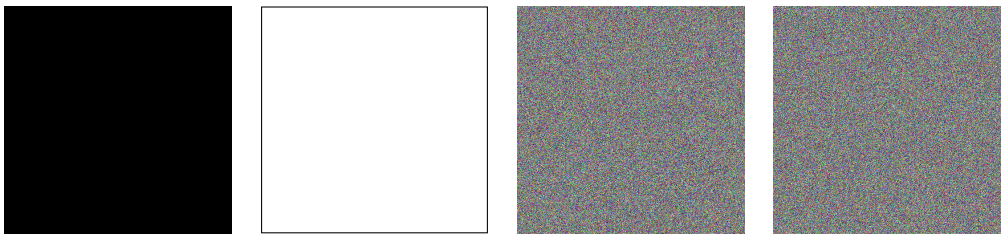
4.12. Chosen-Plaintext and Known-Plaintext Attack Analysis

Chosen-plaintext and known-plaintext attacks are common cryptanalytic techniques used to break cryptographic systems [39]. In these attacks, the adversary either intentionally selects plaintext images to encrypt or possesses a set of plaintext–ciphertext pairs, aiming to observe patterns or recover the secret key. To examine the robustness of the proposed scheme against such attacks, we encrypted two simple plaintext images: an all-white and

an all-black image. The corresponding ciphertexts are shown in Figure 11. These encrypted images appear completely noise-like, with no visible pattern. Table 16 lists the entropy values and the adjacent-pixel correlation coefficients of these ciphertexts. The entropy values are very close to the ideal value of 8, and the correlation coefficients are very close to zero, confirming that the proposed scheme thoroughly destroys the strong regularity present in the chosen plaintexts.

The key security feature of the proposed scheme against these attacks lies in its plaintext-dependent key generation. As described in Section 3.1, the SHA-256 hash of the plaintext image is combined with the external secret key using a bitwise XOR operation. The resulting combined sequence is then used to derive the initial conditions and control parameters of the 3D memristive hyperchaotic map. Consequently, even a slight change in the plaintext leads to completely different chaotic sequences and, therefore, a completely different keystream. Thus, the combination of strong confusion and diffusion, together with plaintext-dependent key generation, makes the proposed scheme highly robust against both known-plaintext and chosen-plaintext attacks.

A cryptanalytic argument supports this claim. In a chosen-plaintext attack, the adversary selects arbitrary plaintexts and obtains their ciphertexts. Because the plaintext is known, the adversary can compute its hash V, but cannot derive the combined sequence $U = S \oplus V$ without the secret key S. The initial conditions, control parameters, and keystreams are all derived from U. Since S remains unknown, the adversary cannot predict the keystream for any other plaintext. Plaintext–ciphertext pairs obtained through chosen-plaintext queries therefore yield no reusable information. The same reasoning applies to known-plaintext attacks.



(**a**) (**b**) (**c**) (**d**)

Figure 11. Plaintext and ciphertext images used for chosen-plaintext attack analysis: (**a**) all-black image, (**b**) all-white image, (**c**) encrypted all-black image, and (**d**) encrypted all-white image.

Table 16. Entropy and correlation coefficients of the encrypted all-black and all-white RGB images.

Image	Channel	Entropy	H	V	D
Encrypted all-white image	R	7.9994	−0.0000	0.0060	0.0091
	G	7.9993	0.0074	−0.0021	0.0007
	B	7.9993	−0.0034	0.0005	0.0130
Encrypted all-black image	R	7.9992	0.0078	−0.0020	0.0034
	G	7.9993	0.0102	−0.0067	−0.0014
	B	7.9993	−0.0079	0.0026	0.0059

4.13. Cropping Robustness Analysis

In practical scenarios, ciphertexts may be partially corrupted or lost during transmission. A secure image encryption scheme should therefore remain robust to such data loss. To evaluate the robustness of the proposed scheme to ciphertext cropping, we removed 1/16, 1/8, 1/4, and 1/2 of the ciphertext pixels using square and strip masks. Figure 12 shows the corresponding decrypted images. Even at a 1/2 cropping ratio, the decrypted image still retains the main content of the plaintext image. The quantitative degradation

is reported in Table 17: as the cropped fraction increases from 1/16 to 1/2, the MSE rises from 581.7 to 4636.8, the PSNR falls from 20.48 dB to 11.47 dB, and the SSIM drops from 0.735 to 0.121.

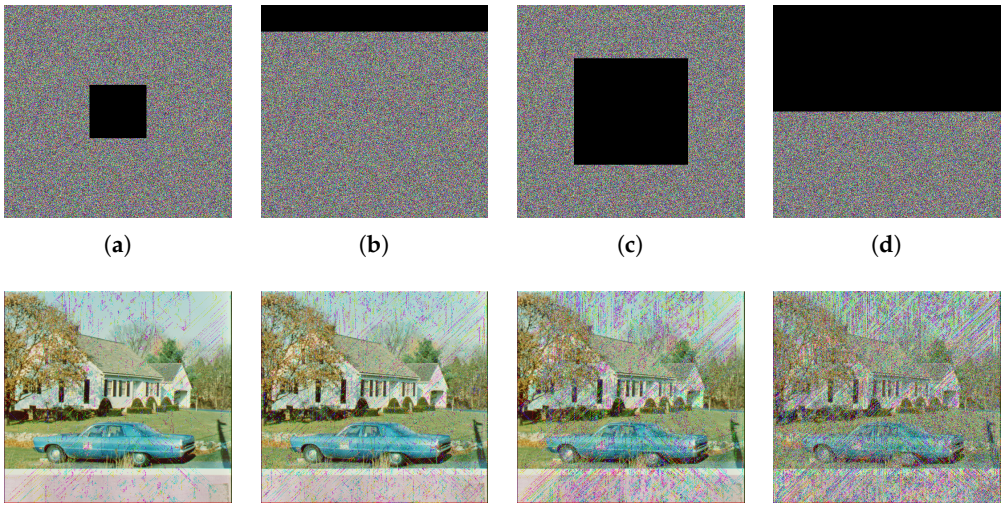


Figure 12. Data loss robustness test on the House image. (**a**–**d**) Encrypted images after removing 1/16, 1/8, 1/4, and 1/2 of the pixels, respectively; (**e**–**h**) the corresponding decrypted versions.

Table 17. MSE, PSNR, and SSIM of decrypted House images after different cropping ratios.

Cropping Ratio	MSE	PSNR (dB)	SSIM
1/16 (square)	581.7164	20.4837	0.7354
1/8 (strip)	1154.7076	17.5061	0.4975
1/4 (square)	2327.8013	14.4613	0.3738
1/2 (strip)	4636.8209	11.4686	0.1208

4.14. Noise Robustness Analysis

Encrypted images may also be corrupted by noise during transmission over untrusted channels. Therefore, a robust encryption scheme must tolerate noise corruption of the ciphertext. To evaluate this, we performed the test on the House image by adding salt-and-pepper noise with densities 0.01, 0.05, and 0.1, as well as Gaussian noise with variances 0.002, 0.01, and 0.05, to the ciphertext. The corresponding decrypted images are shown in Figures 13 and 14, respectively. These figures clearly show that the main content of the images remains recognizable in all cases. The quantitative degradation is reported in Tables 18 and 19. As the noise level increases, the MSE rises while the PSNR and SSIM decrease.

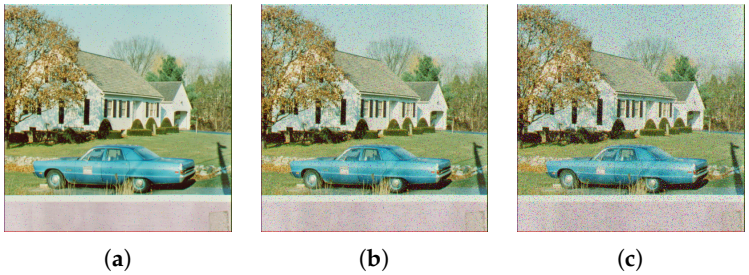
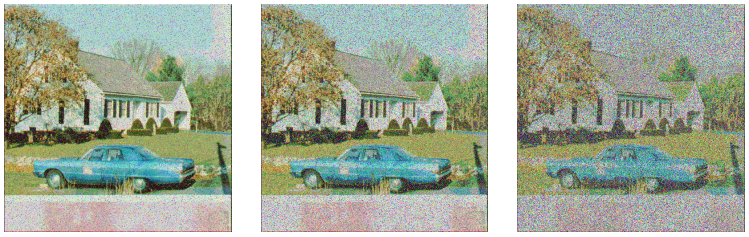


Figure 13. Decrypted House images after salt-and-pepper noise with densities 0.01, 0.05, and 0.1 (**a**–**c**).

Table 18. MSE, PSNR, and SSIM of decrypted House images after salt-and-pepper noise corruption.

Salt-and-Pepper Density	MSE	PSNR (dB)	SSIM
0.01	91.2204	28.5299	0.8709
0.05	461.0845	21.4930	0.5610
0.10	927.4270	18.4580	0.3842



(**a**) (**b**) (**c**)

Figure 14. Decrypted House images after Gaussian noise with variances 0.002, 0.01, and 0.05 (**a**–**c**).

Table 19. MSE, PSNR, and SSIM of decrypted House images after Gaussian noise corruption.

Gaussian Variance	MSE	PSNR (dB)	SSIM
0.002	1237.7225	17.2046	0.2955
0.010	2497.5677	14.1556	0.1806
0.050	4702.0857	11.4079	0.0962

4.15. Computational Complexity and Runtime Analysis

Let $O(\cdot)$ denote the standard asymptotic upper bound on the number of elementary operations, and let $L = MN$ denote the number of pixels per channel. Since the image has three color channels, they contribute only a constant factor and do not affect the asymptotic bounds. The complexity of each stage is described as follows.

- **SHA-256 hashing and key derivation:** Hashing the $3L$-byte plaintext runs in $O(MN)$. The XOR combination with the external secret key and the generation of initial conditions and parameters take $O(1)$ overall. The complexity of this stage is $O(MN)$.
- **Chaotic-sequence generation and preprocessing:** The map is iterated $7L$ times, each step with $O(1)$ arithmetic, so the sequences are generated in $O(MN)$ time and stored in $O(MN)$ memory. Normalizing them and extracting the DNA-rule, key, and mask sequences is one pass over the data, also $O(MN)$. The overall complexity of this stage is $O(MN)$.
- **BEZ scrambling:** Each iteration of the outer loop places one new pixel, so L iterations suffice. Variable-length movement, boundary bouncing, and 3×3 neighborhood escape each take $O(1)$ per step. The global jump reconstructs the unvisited set in $O(MN)$. Writing J for the number of global jumps, $T_{\mathrm{BEZ}} = O(MN + J \cdot MN)$. The worst case $J = \Theta(MN)$ gives $O(M^2N^2)$; in typical cases, global jumps are rare, and the cost is $O(MN)$. Applying the resulting permutation is one gather pass, $O(MN)$.
- **Dynamic DNA encoding and decoding:** Each pixel is encoded through the 8×4 rule table, merged with the key bases by four $O(1)$ operations (ADD, SUB, XNOR, XOR), and decoded back. The DNA stage runs in $O(MN)$.
- **XOR diffusion:** The final mask is applied in a single bitwise XOR pass, $O(MN)$.

The overall encryption complexity is $T_{\mathrm{enc}} = O(MN)$ in typical cases and $O(M^2N^2)$ in the worst case, with $O(MN)$ memory. Decryption is the exact inverse of encryption, so it has the same complexity.

Runtime measurements are reported for the two resolutions used in this study. For the $256 \times 256 \times 3$ color image, encryption required 6.758 s and decryption required 6.682 s. For the $512 \times 512 \times 3$ color image, encryption required 52.522 s and decryption required 50.332 s. These values were obtained on the hardware and software platform described at the beginning of Section 4, using MATLAB 2024. Note that the runtime of a cryptosystem depends on the performance of the executing machine, including the processor, RAM, and storage, as well as on the implementation language and software environment. For instance, an equivalent implementation in Python 3.10 may run at a different speed than one in MATLAB. Accordingly, the reported times should be interpreted as indicative of the behavior of the scheme on the present platform rather than as absolute performance figures, and they are intended mainly to show how the cost scales with image size.

4.16. Ablation Study

Table 20 reports the ablation results. In the scrambling-only comparison, BEZ attains the lowest adjacent-pixel correlation (0.318979), outperforming both the classical zigzag pattern (0.388925) and the block-based dynamic zigzag of Yan et al. [24] (0.371716). This improvement reflects the design features of BEZ: variable-length diagonal runs, four diagonal directions, boundary bouncing, and adaptive collision escape. These features produce a more unpredictable traversal than the fixed classical path or the block-based dynamic zigzag. NPCR increases monotonically from 0.000127% to 88.719559% to 99.031321%, and UACI increases from 0.000000% to 5.162016% to 23.945916%. Entropy remains at 7.297877 for all three scrambling-only configurations, since permutation alone does not change pixel values. Adding DNA diffusion raises the entropy to 7.999249 and lowers the correlation to 0.004832, and the final XOR stage reaches 7.999373, 0.006958, 99.613953%, and 33.475836%.

Table 20. Ablation study on the Peppers image ($512 \times 512 \times 3$). Entropy, NPCR, and UACI are averaged over the three color channels; correlation is the average absolute value over the three channels and three directions.

Configuration	Entropy	Correlation	NPCR (%)	UACI (%)
Classical zigzag only	7.297877	0.388925	0.000127	0.000000
Block-based dynamic zigzag only	7.297877	0.371716	88.719559	5.162016
BEZ only	7.297877	0.318979	99.031321	23.945916
BEZ + DNA	7.999249	0.004832	99.608612	33.423196
BEZ + DNA + XOR (full)	7.999373	0.006958	99.613953	33.475836

5. Conclusions

This article presented a color image encryption scheme based on a newly proposed 3D memristive hyperchaotic map. Dynamical analysis validated the suitability of the map for cryptographic use, with two positive Lyapunov exponents and hyperchaotic behavior sustained over a wide parameter range. The Bounce-Escape Zigzag (BEZ) scrambling algorithm addressed a key limitation of existing zigzag-based techniques by integrating variable-length diagonal traversals, boundary bouncing, and adaptive collision escape. The encryption scheme combined BEZ scrambling, dynamic DNA coding, chaotic XOR diffusion, and key generation from both the external secret key and the plaintext hash. This design ensured that even a single-bit change in the plaintext or secret key produced a completely different ciphertext, which helps resist known-plaintext and chosen-plaintext attacks. Extensive security analyses confirmed these properties. The cipher image entropy reached 7.9998, close to the ideal value of 8; the NPCR and UACI values were close to their theoretical ideals, and the correlation coefficients approached zero. The scheme

also showed strong robustness to cropping and noise corruption. The effective key space was 2^{256}, and an ablation study confirmed that the BEZ, DNA, and XOR stages each contributed measurably to the overall security. Overall, the proposed algorithm provides a high level of cryptographic security and is suitable for secure image communication and storage applications.

Author Contributions: Conceptualization, M.H.; Methodology, M.H. and Z.B.; Validation, M.H. and Z.B.; Formal analysis, M.H.; Investigation, M.H. and Z.B.; Resources, M.G.A.M.; Writing—original draft, M.H.; Writing—review and editing, M.H.; Visualization, M.H.; Supervision, M.G.A.M. and Z.B.; Funding acquisition, M.G.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of Prince Sultan University for covering the Article Processing Charges (APC) of this publication.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We are grateful to the editor and anonymous reviewers for reviewing the manuscript and providing constructive comments for improvement. The authors would like to thank Prince Sultan University for their support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Prasad, S.; Singh, A.K. Survey on medical image encryption: From classical to deep learning-based approaches. *Comput. Electr. Eng.* **2025**, *123*, 110011. [CrossRef]
2. Belazi, A.; Talha, M.; Kharbech, S.; Xiang, W. Novel medical image encryption scheme based on chaos and DNA encoding. *IEEE Access* **2019**, *7*, 36667–36681. [CrossRef]
3. Alexan, W.; Maher, E.A.; Mamdouh, E.; Youssef, M.; Ehab, N. A chaos-based augmented image encryption scheme for satellite images using Fredkin logic. *Sci. Rep.* **2025**, *15*, 37345. [CrossRef] [PubMed]
4. Zia, U.; McCartney, M.; Scotney, B.; Martinez, J.; AbuTair, M.; Memon, J.; Sajjad, A. Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. *Int. J. Inf. Secur.* **2022**, *21*, 917–935. [CrossRef]
5. Inam, S.; Kanwal, S.; Amin, E.; Cheikhrouhou, O.; Hamdi, M. Securing face images in UAV networks using chaos and DNA cryptography approach. *Sci. Rep.* **2025**, *15*, 26011. [CrossRef] [PubMed]
6. Jackson, J.; Perumal, R. A robust image encryption technique based on an improved fractional order chaotic map. *Nonlinear Dyn.* **2025**, *113*, 7277–7296. [CrossRef]
7. Ding, P.; Hu, W.; Geng, P.; Zhang, J.; Zhu, J. A new multi-wing hyperchaotic system and its application in image encryption. *J. Supercomput.* **2025**, *81*, 1201. [CrossRef]
8. Jackson, J.; Perumal, R. A novel 2D hyperchaotic sine logistic map based image encryption scheme. *J. Opt.* **2026**, *55*, 1817–1832. [CrossRef]
9. Kusuma, G.; Sree, P.N.; Mahesh, J.; Shree, N.Y.; Ankitha, A. The Role of AES and DES in Securing Data Transmission. In *Proceedings of the 2025 IEEE 2nd International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*; IEEE: New York, NY, USA, 2025; pp. 1–8.
10. Stanek, M. Insufficient properties of image encryption algorithms. *arXiv* **2018**, arXiv:1809.08661.
11. Kaçar, S.; Çavuşoğlu, Ü.; Jahanshahi, H. Chaos-based image encryption. In *Intelligent Fractal-Based Image Analysis*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 47–71.
12. Gao, J.; Shen, Y.; Li, S.; Zhang, J. An enhanced hybrid chaotic system and its application in image encryption. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 295. [CrossRef]
13. Jamil, A.; Hussain, M.; Bashir, Z.; Malik, M.A. Double-Multi-Scroll attractors in locally active memristive hopfield neural network and application in image encryption. *Signal Image Video Process.* **2026**, *20*, 126. [CrossRef]
14. Pal, P.K.; Kumar, D. The coupled Kaplan-Yorke-Logistic map for the image encryption applications. *Comput. Electr. Eng.* **2024**, *120*, 109850. [CrossRef]
15. Malik, M.A.; Hayat, M.; Bashir, Z. Construction and application of new classes of higher order chaotic polynomial maps. *Comput. Appl. Math.* **2025**, *44*, 248. [CrossRef]

16. Akhtar, M.W.; Bashir, Z.; Abbas Malik, M. Modeling and analysis of variable fractional order hyper-chaotic Lü system with radial basis function neural networks. *Clust. Comput.* **2025**, *28*, 451. [CrossRef]
17. Wang, M.; Yi, Z.; Li, Z. A memristive Ikeda map and its application in image encryption. *Chaos Solitons Fractals* **2025**, *190*, 115740. [CrossRef]
18. Huang, X.; Yan, W.; Dong, W.; Ding, Q. A Novel 2D Hyperchaotic Map with Homogeneous Multistability and Its Application in Image Encryption. *Symmetry* **2025**, *17*, 801. [CrossRef]
19. Bao, H.; Fan, Z.; Hua, Z.; Zhang, Y.; Xu, Q.; Bao, B. Two-dimensional multi-tooth hyperchaotic map and application in medical secure transmission. *Expert Syst. Appl.* **2025**, *290*, 128448. [CrossRef]
20. Wang, Q.; Sang, H.; Wang, P.; Yu, X.; Yang, Z. A novel 4D chaotic system coupling with dual-memristors and application in image encryption. *Sci. Rep.* **2024**, *14*, 29615. [CrossRef] [PubMed]
21. Zhang, C.; Wang, H.; Zhang, Y. A novel 3D hyperchaotic map coupled with discrete tangent memristor: Dynamic analysis, DSP implementation, and image encryption application. *Expert Syst. Appl.* **2025**, *286*, 128091. [CrossRef]
22. Feali, M.S. Chaotic dynamics of discrete memristor-coupled Sinh map. *Chaos Solitons Fractals* **2025**, *196*, 116480. [CrossRef]
23. Güvenoğlu, E.; Tunalı, V. ZigZag transform with Durstenfeld shuffle for fast and secure image encryption. *Connect. Sci.* **2023**, *35*, 2162000. [CrossRef]
24. Yan, S.; Liu, Y.; Zhang, J.; Jiang, D. Medical image encryption algorithm based on a novel 4D chaotic system and dynamic zigzag transformation. *Phys. Scr.* **2026**, *101*, 085202. [CrossRef]
25. Chowdhury, A.; Uddin, M.; Hassan, M.R.; Billah, M.M. Image encryption algorithm based on DNA sequence-driven key scrambling and secure hashing. *J. Inf. Secur. Appl.* **2025**, *93*, 104140. [CrossRef]
26. Hassan, A.; Zhou, L. A novel 6D four-wing memristive hyperchaotic system: Generalized fixed-time synchronization and its application in secure image encryption. *Chaos Solitons Fractals* **2025**, *192*, 115986. [CrossRef]
27. Gao, S.; Ding, S.; Ho-Ching Iu, H.; Erkan, U.; Toktas, A.; Simsek, C.; Wu, R.; Xu, X.; Cao, Y.; Mou, J. A three-dimensional memristor-based hyperchaotic map for pseudorandom number generation and multi-image encryption. *Chaos Interdiscip. J. Nonlinear Sci.* **2025**, *35*, 073105. [CrossRef] [PubMed]
28. Nabil, H.; Tayeb, H. A memristive hyperchaotic map with extreme multistability: dynamics, complexity, and application in image encryption. *Eur. Phys. J. Plus* **2025**, *140*, 999. [CrossRef]
29. Wang, X.; Wei, Y.; Sun, X.; Fan, Z.; Du, B. A novel discrete memristive hyperchaotic map with multi-layer differentiation, multi-amplitude modulation, and multi-offset boosting. *Chaos Interdiscip. J. Nonlinear Sci.* **2024**, *34*, 113133. [CrossRef] [PubMed]
30. Zhang, Q.; Han, J. A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. *Multimed. Tools Appl.* **2021**, *80*, 13841–13864. [CrossRef]
31. Vargas Valencia, J.A.; Arango, C.A.M.; García, J.D.; Guerra, R.U.; Gómez, L.F.D. Chaotic Image Encryption by Intra-Channel Diffusion and Inter-Channel Confusion. *J. Cybersecur. Priv.* **2026**, *6*, 101. [CrossRef]
32. Tang, C.; Wang, S.; Shu, Y.; Ren, F. Encryption algorithm based on improved four-dimensional chaotic system and dynamic DNA encoding. *AIP Adv.* **2024**, *14*, 095226. [CrossRef]
33. Hu, G.; Luo, Y.; Ye, G. Color image encryption scheme using time-delayed memristive hyperchaotic system and reversible second-order cellular automaton. *Nonlinear Dyn.* **2026**, *114*, 492. [CrossRef]
34. Yan, X.; Hu, Q.; Teng, L. A novel color image encryption method based on new three-dimensional chaotic mapping and DNA coding. *Nonlinear Dyn.* **2025**, *113*, 1799–1826. [CrossRef]
35. Ullah, A.; Shah, A.A.; Khan, J.S.; Sajjad, M.; Boulila, W.; Akgul, A.; Masood, J.; Ghaleb, F.A.; Shah, S.A.; Ahmad, J. An efficient lightweight image encryption scheme using multichaos. *Secur. Commun. Netw.* **2022**, *2022*, 5680357. [CrossRef]
36. Kolivand, H.; Hamood, S.F.; Asadianfam, S.; Mohd Rahim, M.S.; Hurst, W. Image encryption framework based on multi-chaotic maps and equal pixel values quantization. *Multimed. Tools Appl.* **2025**, *84*, 17769–17804. [CrossRef]
37. Chen, Z.; Chen, Z.; Zou, S. A card-gameplay-inspired color image encryption scheme driven by novel 2D hyperchaotic map. *Phys. Scr.* **2025**, *100*, 115211. [CrossRef]
38. Liu, L.; Zhang, L.; Jiang, D.; Guan, Y.; Zhang, Z. A simultaneous scrambling and diffusion color image encryption algorithm based on Hopfield chaotic neural network. *IEEE Access* **2019**, *7*, 185796–185810. [CrossRef]
39. Liu, H.; Liu, Y. Cryptanalyzing an image encryption scheme based on hybrid chaotic system and cyclic elliptic curve. *Opt. Laser Technol.* **2014**, *56*, 15–19. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.