Dual memristive Hopfield neural networks: Bistable coexistence attractor and image encryption

Zhen Yang(杨振)[1] and Xiaolin Ye(叶晓林)[2,†]

[1] *School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan* 114051, *China*
[2] *School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan* 114051, *China*

(Received 26 June 2026; revised manuscript received 20 August 2026; accepted manuscript online 25 August 2026)

This paper proposes a novel dual memristive Hopfield neural network (HNN) system which can generate coexisting attractors. The time-domain waveform and hysteresis loop of two memristors are displayed. And the dynamic behavior is shown with different dual memristors' parameters. Particularly, based on the characteristic that the equilibrium point of the system is symmetrical about the origin, we select two initial values which are symmetrical about the origin such as $[0.1, 0.1, 0.1]$ and $[-0.1, -0.1, -0.1]$, find a pair of bistable coexisting attractors, and analyze their dynamic behavior such as coexisting bifurcation diagrams and Lyapunov exponents (LEs). And we design a new encryption process. It provides new ideas for research on memristor-coupled HNN.

Keywords: chaos, memristor, Hopfield neural network (HNN), Lyapunov exponents (LEs)

PACS: 05.45.–a, 95.75.Mn, 95.10.Fh

DOI: 10.1088/1674-1056/ae9e1a **CSTR:** 32038.14.CPB.ae9e1a

1. Introduction

Recently, some chaotic systems have attracted extensive attention due to their applications in the engineering fields.[1–3] When a chaotic system's equations have a bunch of solutions, it may have the ability to produce multistability.[4,5] In some further studies on multistable systems, the extremely multistable system can be obtained by introducing some trigonometric functions to the original system.[6,7] This type of chaotic system depends extremely on initial value, and it can generate complex dynamic behaviors under different initial conditions.[8–10] However, in the current study of the multistable chaotic system, the research focus is basically on the dynamic analysis of systems. There are relatively few specific applications for this type of system. Thus, it is particularly important to study the generation mechanism and application prospects of numerous stable systems.

Due to better memory characteristics and nonlinear characteristics, the neural network system is often applied to the related field.[11–13] In these systems, due to the advantages such as the high sensitivity to initial values and high complexity, Hopfield neural network (HNN) is applied as a system model to the research of nonlinear correlation fields.[14,15] Reference [16] gives two capacitors to show the bioinspired bimembrane effect, and replaces the nonlinear ionic currents by using memristive Josephson currents. Reference [17] shows the potential VO_2 encoding based on neurons memristor and genetic information, and shows the corresponding dynamic characteristics. Reference [18] gives a 4D memristive HNN model with non-bifurcation regulation, and constructs an offset control function. The new HNN model can produce multiple coexisting attractors. Compared with previous studies, this article further highlights the practicality of the HNN system. Based on the study of coexisting multiple stable states, the corresponding image encryption algorithm is designed. It provides a direction for the research of HNN in the field of cryptography. The super strong memory characteristic of memristor makes them often combined with HNN.[19] Reference [20] gives a brain-inspired memristor-coupled HNN which has rich dynamical behaviors, analyzes its 2D dynamical distribution maps and LEs, and displays its coexisting attractors. Reference [21] gives a new digital watermarking algorithm by a new HNN with memristor, and tests the security of the algorithm. Memristors and HNN system reflect many similar characteristics. Thus, the research on memristive HNN is of great significance to the field of nonlinearity.

Whereas, with the continuous improvement of construction methods of chaotic attractors, there are also more and more methods to obtain the coexisting attractors.[22–25] In particular, as a PRNG, these also have made many exploratory attempts in fields such as image encryption.[26–31] Due to the pseudo-random nature of chaotic sequences, bidirectional encryption based on chaos becomes possible.[32–36] Some traditional structural schemes can only generate the multi-stability system for position movement. The research on the construction methods of HNN and memristor is relatively scarce. And it is a very interesting study on the construction method of memristive HNN coexisting attractor. In this paper, we propose an HNN system and give the corresponding weights

†Corresponding author. E-mail: yexl0207@163.com

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

CPB Online In-Press

which can produce chaos. Different from some ordinary systems, the chaotic orbits of the proposed system are bistable. The unpredictable bistable chaotic orbit is the better choice for PRNG. And the construction of bistable state causes coexistence effect. Particularly, the influence of mixed parameters on the dynamical behavior is elaborated.

Sections can be summarized as follows. In Section 2, we give a dual memristor model. In Section 3, we display a memristive HNN and analyze the mixed-parameter dynamics. In Section 4, an encryption technique is designed. Finally, we realize chaotic attractors by DSP.

2. Performance test of dual memristors

Memristors have received extensive attention since their proposal. The memory characteristic of a memristor is its most prominent feature. This characteristic is often characterized by its hysteresis loops. In this section, we introduce two mathematical models of memristors into a third-order HNN model. The two memristors' mathematical models are the first-order linear-charge-controlled memristor (LCCM) and the second-order flux-controlled memristor (FCM).

2.1. Linear-charge controlled memristor

A first-order linear-charge controlled memristor was proposed by HP Labs. Charge is used to control the magnitude of resistance. Its mathematical expression is

$$\begin{cases} \xi(q) = a_\xi q + b_\xi q^2, \\ M(q) = d\xi(q)/dq = a_\xi + b_\xi q, \end{cases} \tag{1}$$

where q and $M(q)$ reflects charge and resistance obtained by the differential of $\xi(q)$ respect to q. $M(q)$ just is resistance of LCCM. The resistance value can be controlled by the amount of electric charge q. a_ξ and b_ξ are two controllable real constants. In the test of characteristic curve, a_ξ and b_ξ are set as 10.

Figure 1 gives the characteristics among various variables within LCCM. Set the test frequency to $f = 0.1, \ldots, 0.4$ Hz respectively; Figs. 1(a)–1(c) reflect the time domain waveform when the time ranges from 0–100 seconds. From the current curve versus time obtained by Fig. 1(a), as the frequency value increases, the current waveform shows no obvious change. Unlike the current curve given by Fig. 1(a), as the frequency value increases, the corresponding waveform diagrams of voltage and magnetic flux change significantly in terms of amplitude. Overall, the trend is that the higher the frequency, the smaller the amplitude. Figures 1(d)–1(f) give the hysteresis loops corresponding to u–i, Φ–u, Φ–i, respectively. Three variables show hysteresis loops in pairs; this is also the root cause of the memory characteristics of memristors.

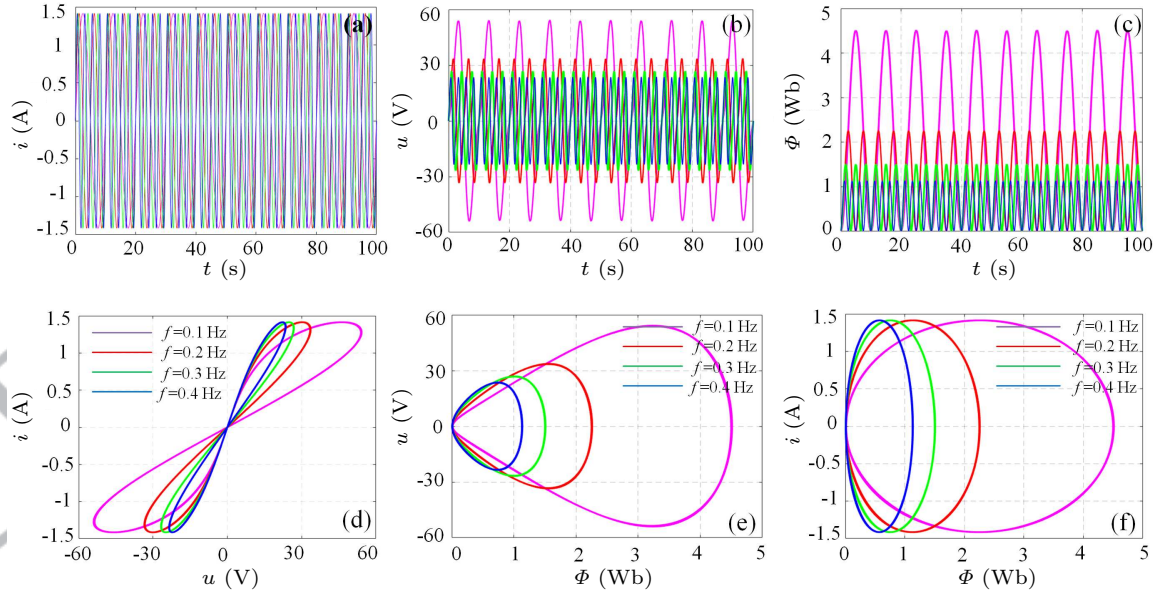


Fig. 1. Characteristic curve of LCCM. (a) Current curve. (b) Voltage curve. (c) Magnetic flux curve. (d) Hysteresis loop of u–i. (e) Hysteresis loop of Φ–u. (f) Hysteresis loop of Φ–i.

2.2. Secondary flux-controlled memristor

The secondary magnetic-controlled memristor controls the resistance value by adjusting the magnitude of the magnetic flux. Its equation is

$$\begin{cases} \rho(\Phi) = a_\psi \Phi - b_\psi \Phi^3, \\ W(\Phi) = d\rho(\Phi)/d\Phi = a_\psi \Phi - 3b_\psi \Phi^2, \end{cases} \tag{2}$$

where Φ and $W(\Phi)$ reflects magnetic flux and impedance of memristor obtained by the differential of $\rho(\Phi)$ respect to Φ. $W(\Phi)$ just is impedance of FCM. The impedance value can be controlled by the amount of magnetic flux Φ. a_ψ and b_ψ are two controllable real constants. In the test of characteristic curve, $a_\xi = 10$ and $b_\xi = 1$. Figure 2 gives the characteris-

tics among various variables within FCM. The frequency are $f = 0.3, \ldots, 0.6$ Hz, Figs. 2(a)–2(c) reflects the time domain waveform when the time ranges are given from 0–100 seconds. In particular, the amplitude of magnetic flux gradually decreases as the frequency increases. Figures 2(d)–2(f) gives the hysteresis loops corresponding to $u-i$, $\Phi-u$, $\Phi-i$, respectively. And all characteristic curves are characterized by hysteresis loops.

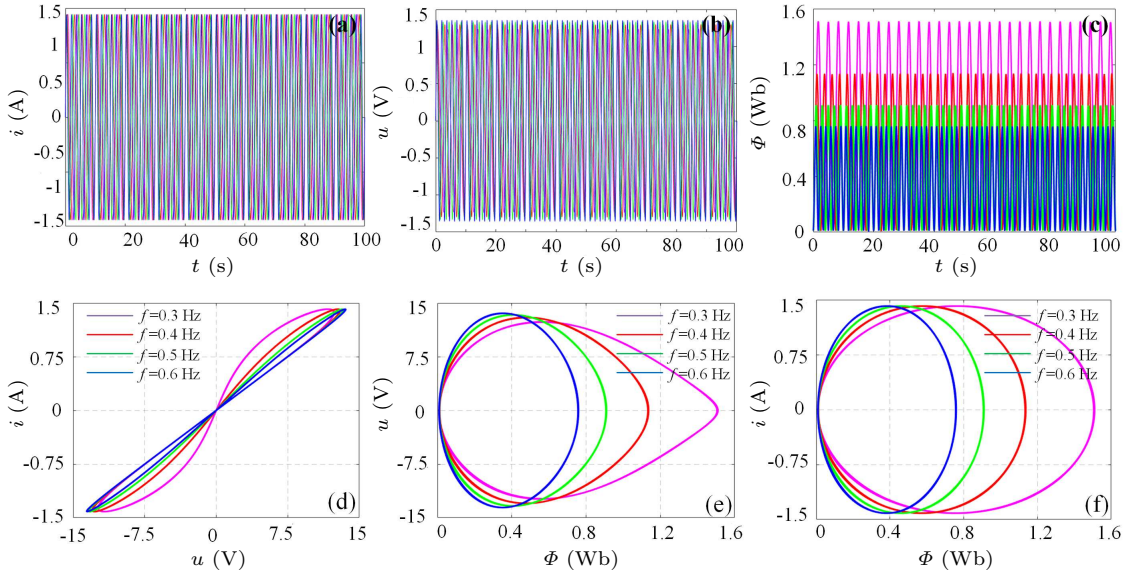


Fig. 2. Characteristic curve of FCM. (a) Current curve. (b) Voltage curve. (c) Magnetic flux curve. (d) Hysteresis loop of u–i. (e) Hysteresis loop of Φ–u. (f) Hysteresis loop of Φ–i.

3. Constructing a new HNN model with dual memristors

A novel kind of HNN based on tripartite neurons is given, and its connection schematic diagram is shown in Fig. 3. w_{11}, w_{22}, and w_{33} reflect the own weights based on tripartite neurons. The other weights are the mutual weights within tripartite neurons. By adjusting these weight values, we can make the model produce chaos.

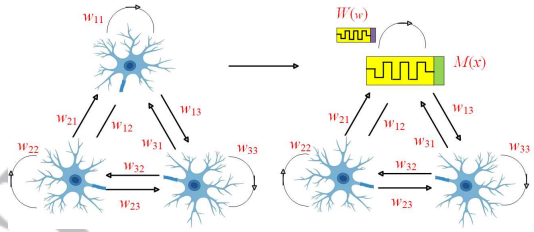


Fig. 3. Schematic diagram based on tripartite neurons.

3.1. HNN model

The newly developed HNN system is an equation based on tripartite neurons; it can be given

$$\dot{x} = -x + W\tanh(x), \tag{3}$$

where $x = [x_1, x_2, x_3]$ is three status variable, $\tanh(x)$ is hyperbolic tangent function, HNN equation can be described

$$\begin{cases} \dot{x} = -M(x) + W(w)\tanh(x) + w_{12}\tanh(y) + w_{13}\tanh(z), \\ \dot{y} = -y + w_{21}\tanh(x) + w_{22}\tanh(y) + w_{23}\tanh(z), \\ \dot{z} = -z + w_{31}\tanh(x) + w_{32}\tanh(y) + w_{33}\tanh(z), \end{cases} \tag{4}$$

where we assign weight values to the weight matrix W and have

$$W = \begin{bmatrix} W(w) & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} -1.4 & 1.5 & -7 \\ 1.1 & 0.02 & 2.8 \\ w_{31} & -2 & 4 \end{bmatrix}. \tag{5}$$

When w_{11} is assigned different weight value, we can get the proposed HNN exhibits a wide range of dynamic behaviors as different w_{31} value.

3.2. Attractors with different memristive parameters

To study the influence of memristive parameters on the system's attractor, set the memristive parameters as $a_\xi = 1$, $b_\xi = 0.01$, $a_\psi = 1$, and $b_\psi = 0.004$; system (4) can generate rich dynamic behavior. Fix the other memristor's parameters and let a_ξ be the variable parameter. We can get different types of attractors as given by Figs. 4(a)–4(d). Similarly, we can get different types of attractors as given by Figs. 4(e)–4(h). In particular, $P-1$ shows a period-1 state, $P-2$ shows a period-2 state, and so on. Fix the other memristor's parameters and let a_ψ be the variable parameter. We can get different types of attractors as given by Figs. 5(a)–5(d). Similarly, we can get different types of attractors as given by Figs. 5(e)–5(h). In particular, $P-1$ shows a period-1 state, $P-2$ shows a period-2 state, and so on. From the variation of the attractors, the system is extremely sensitive to the memristor's parameters. Next, we will further verify the system's dynamic behavior as the dual memristive parameters change.

3.3. Dynamic analysis with different memristive parameters

To display the influence of memristive parameters on the system's dynamic behavior, we demonstrate the effect of each parameter change on the bifurcation trajectory of the system. Keeping other memristive parameters unchanged, let each memristive parameter be the variable. Figure 6 gives the bifurcation diagrams with different memristive parameters. All the bifurcation diagrams present the inverse period bifurcation state. We can clearly observe the orbital evolution process of the system entering chaotic behavior. The system's orbit continuously forks as the parameters change, from a chaotic orbit to a periodic orbit. Lyapunov exponents (LEs) can quantitatively describe the system's dynamic behavior changes. We can see that the exponential and bifurcation diagrams of the system correspond one-to-one. To further understand the influence of sensitive changes in parameters

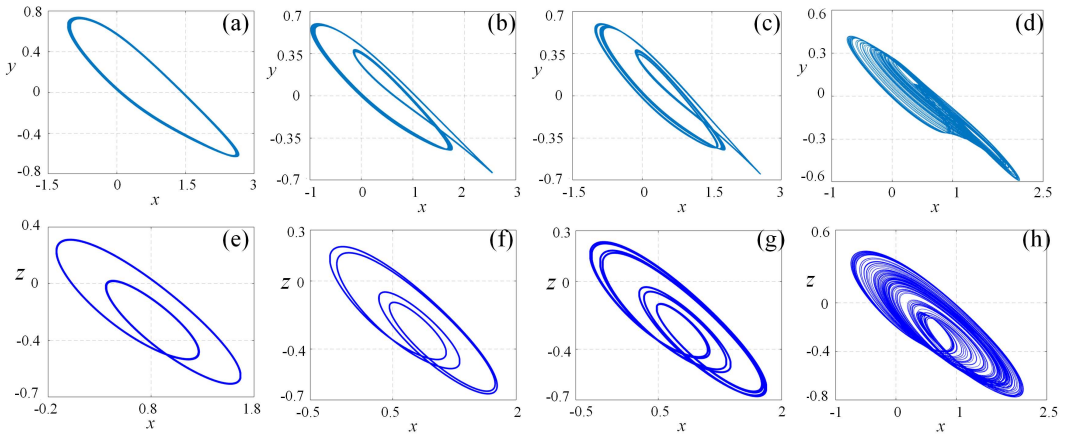


Fig. 4. Attractor with different LCCM's parameters. (a) $P-1$, $a_\xi = 0.45$. (b) $P-2$, $a_\xi = 0.8$. (c) $P-3$, $a_\xi = 0.81$. (d) Chaos, $a_\xi = 1$. (e) $P-1$, $b_\xi = 0.1$. (f) $P-2$, $b_\xi = 0.08$. (g) $P-3$, $a_\xi = 0.07$. (h) Chaos, $a_\xi = 0.01$.

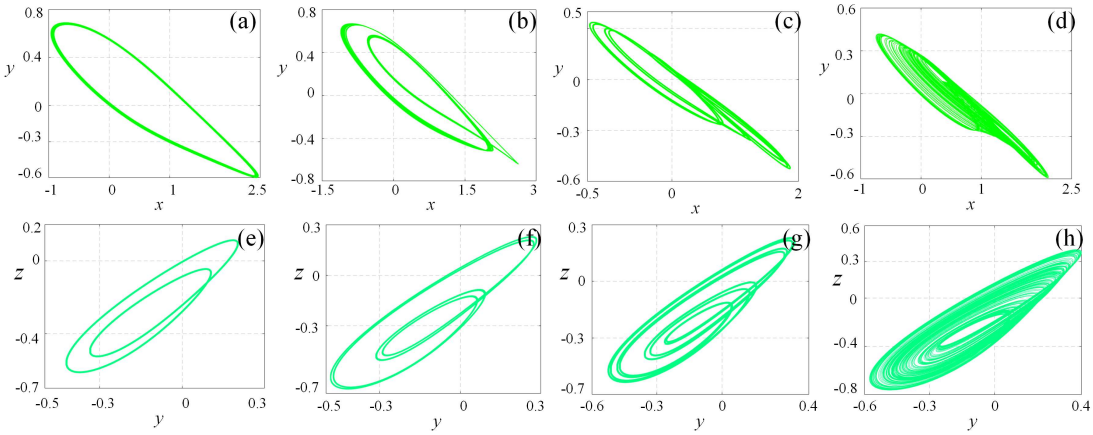


Fig. 5. Attractor with different FCM's parameters. (a) $P-1$, $a_\psi = 0.45$. (b) $P-2$, $a_\psi = 0.56$. (c) $P-3$, $a_\psi = 1.05$. (d) Chaos, $a_\psi = 1$. (e) $P-1$, $b_\psi = 0.03$. (f) $P-2$, $b_\psi = 0.02$. (g) $P-3$, $b_\psi = 0.07$. (h) Chaos, $b_\psi = 0.004$.

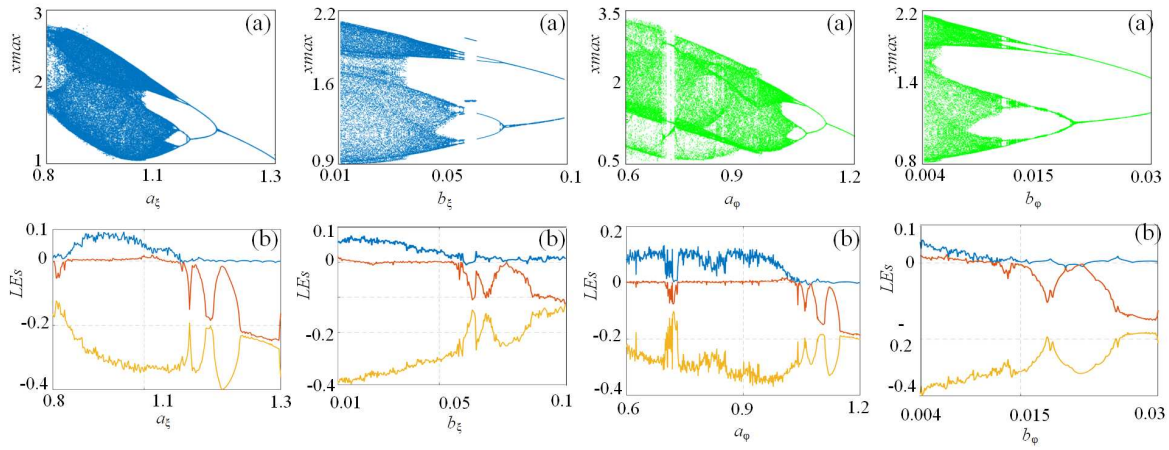


Fig. 6. Dynamic analysis with different memristive parameters. (a) Bifurcation graphs. (b) LEs.

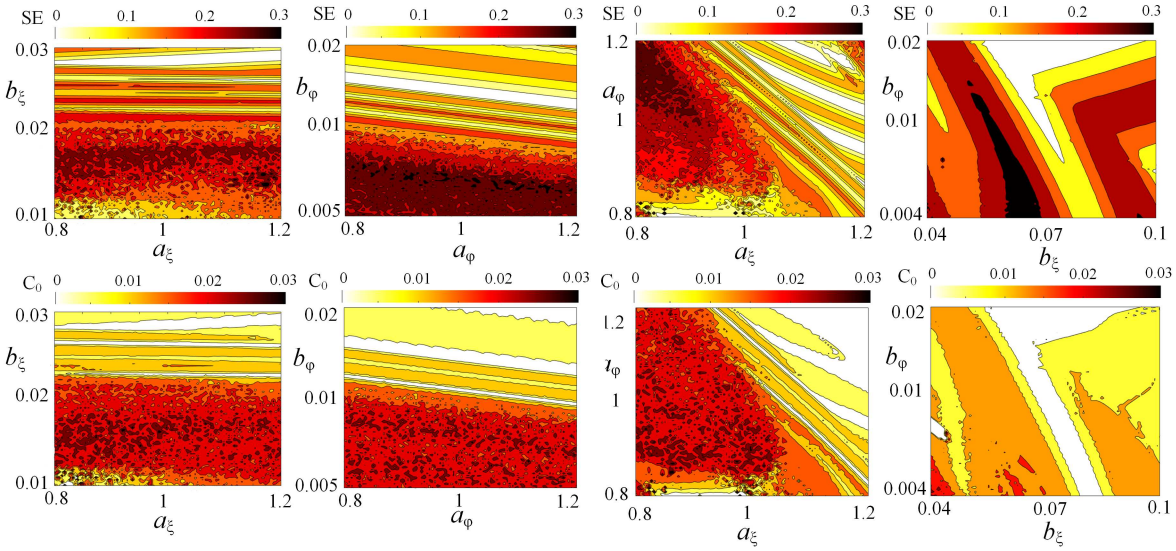


Fig. 7. Two-dimension SE and C_0 complexity with two different memristive parameters.

on the dynamic behavior of the system. We conducted some further experiments that system's dynamic behavior changes along with the two memristive parameters and the cross memristive parameters. We can see that the system is extremely sensitive to the memristive parameters. This is closely related to the sensitive parameter characteristic of chaos. From the perspective of the influence of the above parameter change on the dynamical, the system is quite sensitive to changes within memristive parameters. To further verify, we studied the influence on two parameters to system's dynamic behavior. Figure 7 gives the SE and C_0 complexity with two changing memristive parameters. The dark part is chaos, low value shows periodic state. The dynamic behavior of system under the action of double memristive parameters is clearly visible.

3.4. Dynamical analysis with different weights and coexisting behaviors

In this section, keeping other parameters unchanged, we obtain different attractors by changing the weights W_{31}. From the attractors given by Fig. 8, we can see that the system is also sensitive to the weight value. The coexisting attractors are a multistable phenomenon in nonlinear dynamical systems, referring to the evolution of multiple independent and stable long-term behavior patterns due to different initial states. The chaotic attractors may include periodic oscillations, quasi-periodic motions, or chaotic states; each of them has an independent attractive basin and together they constitute the complex dynamic structure of the system. Since the proposed system is a system symmetrical about the origin, keeping all the parameters unchanged, let w_{31} be the variable. We select a set of initial values symmetrical about the origin, $[0.1, 0.1, 0.1]$ and $[-0.1, -0.1, -0.1]$. Figure 9 shows different coexistence attractors on different planes. The Lyapunov exponent originated from the theory of nonlinear dynamical systems; it is used to quantify the average separation or convergence rate as it evolves over time. Fixing the other weight according to Eq. (5), Fig. 10 gives the LEs spectrum with different weights. The exponential quantitative description of dynamic behavior is clearly visible.

In nonlinear dynamical systems, a chaotic orbit refers to the trajectory of the system's state evolving over time. When a system is adjusted by system parameters, its long-term behavior may undergo a qualitative change, which is called bifurcation. The coexisting bifurcation orbits originate from different initial values. They develop into a coexisting chaotic orbit through multiple cycles. Select a set of initial values $[0.1, 0.1, 0.1]$ and $[-0.1, -0.1, -0.1]$. The blue orbit reflects $[0.1, 0.1, 0.1]$ and the red orbit reflects $[-0.1, -0.1, -0.1]$. Figure 11 gives coexisting bifurcation diagrams with different weights. The bifurcation orbits are conducive to the evolution process of dynamic behavior.

3.5. The influence of simultaneous changes in mixed parameters on system behavior

In this section, we explored the influence of simultaneous changes in memristive parameters and weight parameters on the dynamic behavior of the system. Look at Fig. 12; it gives some two-dimensional dynamics testing tools, e.g., dynamic map and complexity. Complexity reflects the oscillation degree of the system's sequence. When an HNN system is in a chaotic state, the sequence's vibration degree shows its intensity. Two-dimensional complexity testing expresses the system's dynamic behavior as two mixed parameters change based on the principle of contour lines. The joint variation of mixed parameters enriches the HNN's dynamics. Table 1 gives the chaotic performance comparison; the proposed HNN with dual memristors has better robustness.

CPB Online In Press

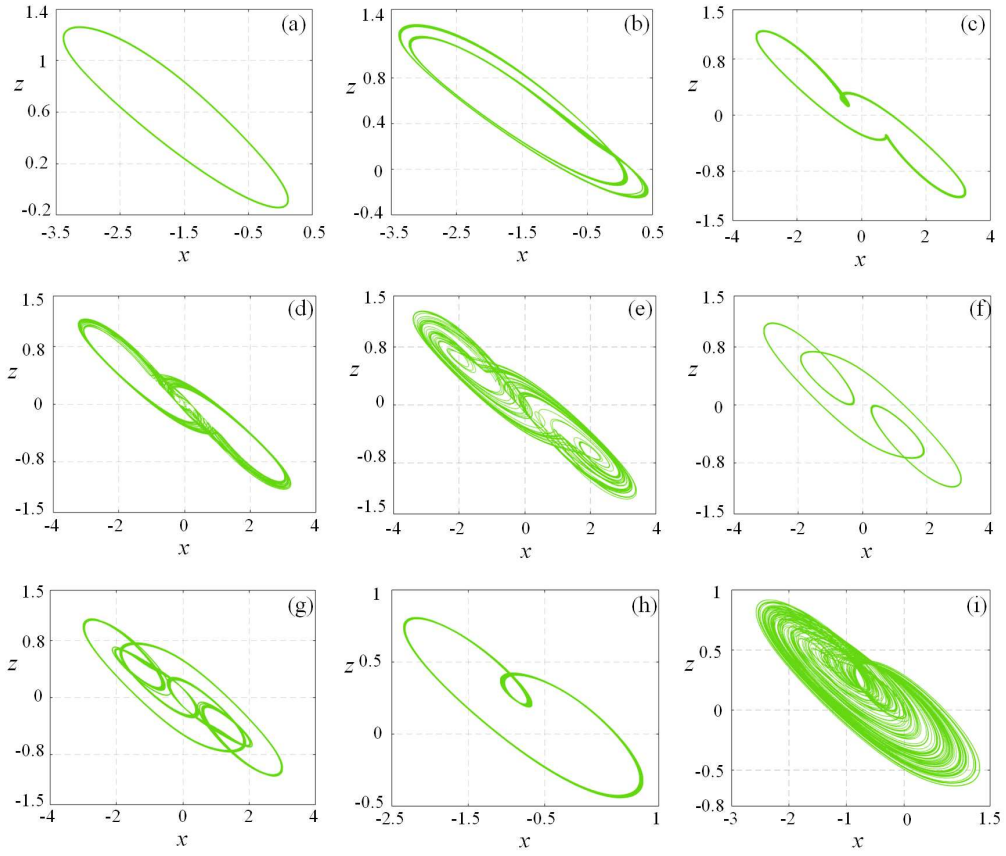


Fig. 8. Different dynamic trajectories on x–z plane. (a) Type-I limit cycle when $w_{31} = 0.58$. (b) Type-I periodic-2 when $w_{31} = 0.59$. (c) Type-II limit cycle when $w_{31} = 0.61$. (d) Type-II periodic-3 when $w_{31} = 0.62$. (e) Type-II chaotic attractor when $w_{31} = 0.63$. (f) Type-III limit cycle when $w_{31} = 0.73$. (g) Type-III periodic-3 when $w_{31} = 0.79$. (h) Type-IV limit cycle when $w_{31} = 0.9$. (i) Type-IV chaotic attractor when $w_{31} = 0.92$.

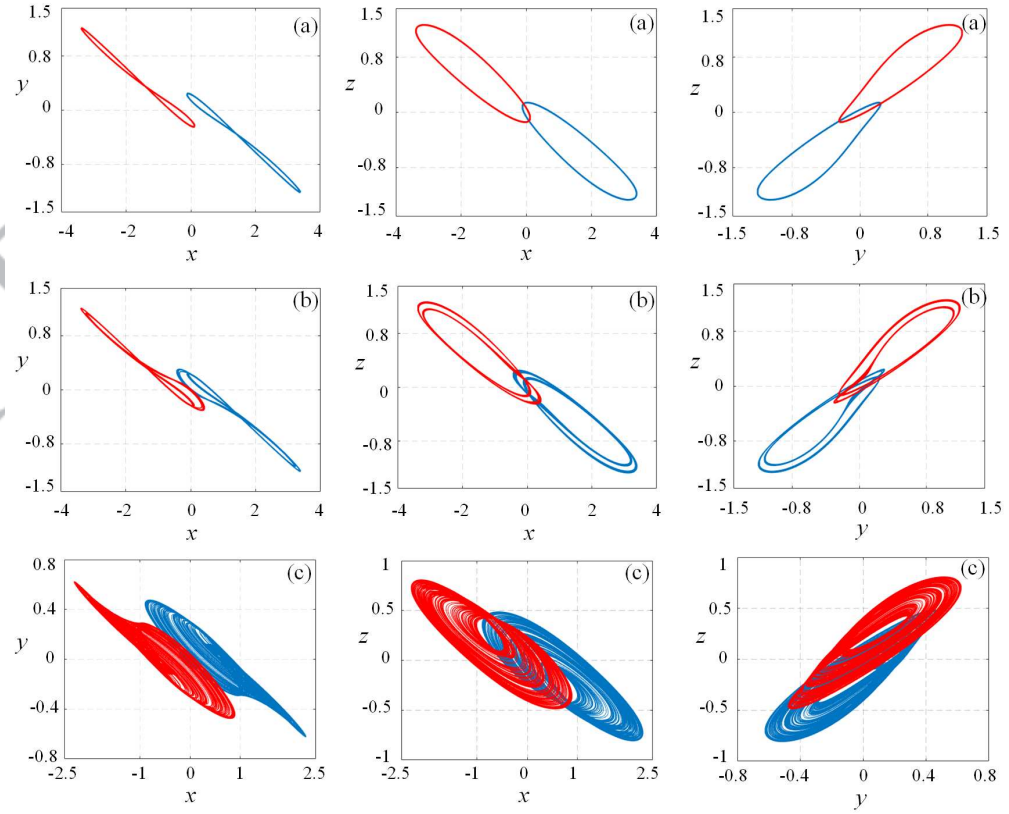


Fig. 9. Coexisting behavior on different planes. (a) Periodic-1 coexistence when $w_{31} = 0.58$. (b) Periodic-2 coexistence when $w_{31} = 0.59$. (c) Coexistence attractor when $w_{31} = 0.95$.

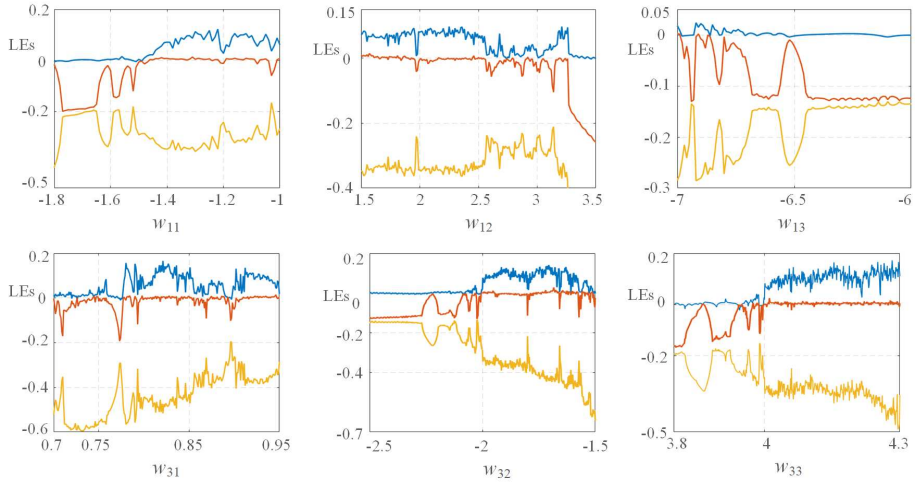


Fig. 10. LEs spectra with different weights.

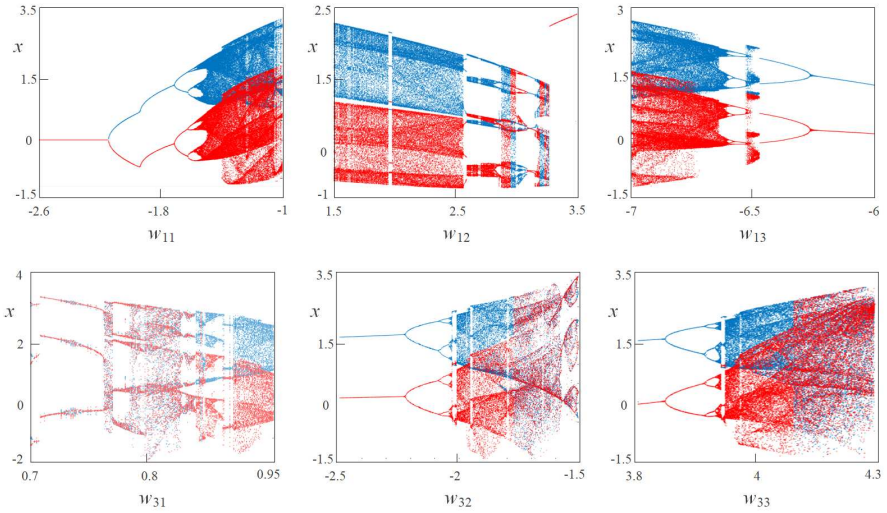


Fig. 11. Coexisting bifurcation diagrams with different weights.

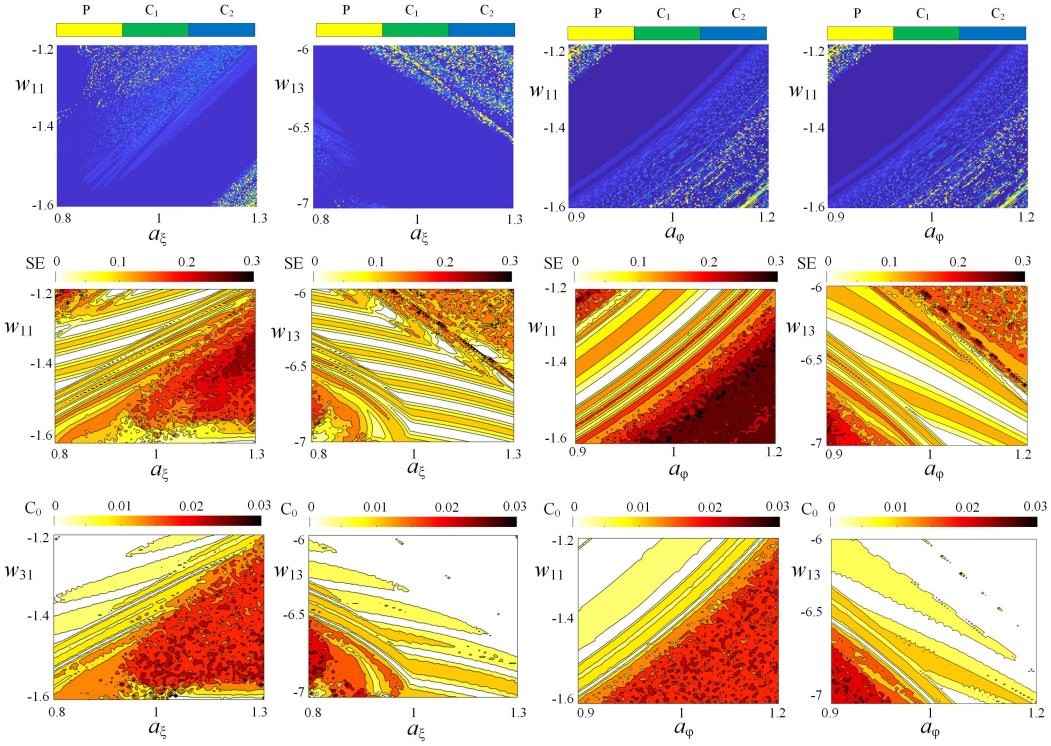


Fig. 12. Parabolic coexisting attractors on different planes.

Table 1. Comparison of chaotic performance.

HNN model	Max LE	Max SE	Max C_0
Original HNN	0.1288	0.2661	0.0271
HNN with dual memristors	0.1334	**0.2935**	**0.0295**
Ref. [18]	0.1308	0.2713	0.0286
Ref. [19]	0.1320	0.2800	0.0289
Ref. [20]	**0.1342**	0.2899	0.0294

3.6. NIST test

When the sequences of a system are applied to the encryption algorithm, the best choice is the true random sequence. Because the true random sequence is essentially a completely random sequence, the encryption effect is also the most ideal. However, the completely random sequence is hard to decrypt, so it is often used for one-way encryption. The chaotic sequence has good pseudo-random property, and the result of the chaotic sequence is unique after the chaotic key is set. Thus, it is often used in the design of the encryption algorithm. In particular, the NIST test often quantitatively describes the pseudo-randomness of the chaotic sequence. Generally, p value and proportion are the two most important evaluation criteria for the pseudo-randomness of the sequence in the NIST test. The reflection of the distribution situation of the sequence, which can be calculated by

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - 0.1n)^2}{0.1n}, \tag{6}$$

where F_i is the number of p values within $[0.1(i-1), 0.1i]$, and n is the number of groups. Then, we can get

$$P\text{-value} = \text{igamc}\left(4.5, \frac{\chi^2}{2}\right), \tag{7}$$

in which igamc is a high-precision incomplete gamma function. If the p value is more than 0.0001, the sequence follows the uniform distribution. Keeping all parameters unchanged, the proposed discrete chaotic sequences can be tested, and the specific test result is shown in Table 2. In particular, relative to Refs. [35,36], most of the test results of the proposed algorithm are better. That is to say, the sequences of the HNN system with dual memristor have better pseudo-randomness.

Table 2. Results of NIST test.

Statistical test	The proposed system		Ref. [35]		Ref. [36]	
	P-value	Proportion	P-value	Proportion	P-value	Proportion
Frequency	0.88622	0.99	0.81679	0.97	0.73679	1
Block frequency	0.85263	0.98	0.87596	1	0.72561	0.97
Cumulative sums	0.79871	0.99	0.48987	0.98	0.742987	0.98
Runs	0.34441	0.97	0.12879	0.97	0.22586	0.98
Longest run	0.56182	1	0.62678	1	0.49312	0.97
Rank	0.37736	0.99	0.35873	0.97	0.19456	1
FFT	0.37951	0.99	0.31798	0.98	0.45798	0.98
Nonoverlapping template	0.17415	0.99	0.22475	0.97	0.19475	0.97
Overlapping template	0.73924	1	0.58974	1	0.49897	0.99
Universal	0.81973	0.98	0.81471	0.97	0.79471	1
Approximate entropy	0.32151	0.98	0.36841	0.97	0.19841	0.97
Random excursions	0.01366	0.99	0.01989	0.98	0.05589	1
Random excursion variant	0.01356	0.99	0.11356	0.98	0.01356	0.97
Serial	0.00820	1	0.10811	1	0.00910	0.97
Linear complexity	0.68454	0.99	0.37895	0.99	0.52395	0.98

4. Design of encryption scheme

Compared with the traditional Z scrambling and Arnold mapping, the designed scrambling algorithm can effectively scramble adjacent pixels to enhance the security of the encryption algorithm. In terms of diffusion algorithms, we adopt an improved design based on the spatio-temporal diffusion algorithm. The introduction of some scrambling parameters achieves a simple algorithm, making the encryption algorithm highly secure.

4.1. Chaotic secret key and key space

As shown in Fig. 13, a more than 250-bit-long secret key is used in the proposed encryption algorithm. The 250-bit-long secret key is divided into block K_1, two memristor parameters b_ξ, b_ψ, a weight parameter W_{31} and an initial value generated by x sequence x_0. Here, they are set as $b_\xi = 0.01$, $b_\psi = 0.004$, $W_{31} = 0.92$ and $x_0 = 0.1$. Then K_1 is divided into (K_{11}, K_{12}, K_{13}, K_{14}, K_{15}), each of which is given 50 bits. Based on the designed secret key, we can design the following encryption algorithm.

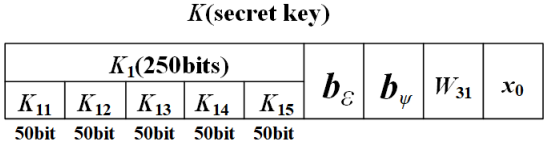


Fig. 13. The secret key based on different parameters.

CPB Online In Press

The large key space can effectively resist the strong attack. For the 250-bit-long secret key, the remaining key parameters are locked in this key space. Thus, the sum of the key space size is 2^{250}.

4.2. Design of scrambling scheme

As shown in Fig. 14, the designed spiral arrangement algorithm has $n \times n$ vertices on a square plane, and the loop path is exactly able to completely cover each pixel point. The spiral arrangement is to scramble of next pixel. If the access is beyond the boundary of matrix A or this pixel is already accessed, we change the direction of loop arrangement. Until all the pixels in matrix A are stored in matrix B, the loop will be stopped. Finally, we obtain the image *img_s* after spiral arrangement scrambling. The pixels gradually shrink from the edge of the image towards the inside, and the arrangement mode of each layer depends on the results of the previous layer. That means the spiral arrangement can effectively break the plaintext pixels' correlation. This damage makes it more difficult to crack the original image and enhances the resistance to cracking of the algorithm. In addition, the spiral arrangement algorithm is not limited by the size or color space of images, so it can be widely used in various image encryption systems. For a detailed description of the spiral arrangement scrambling algorithm, please refer to Algorithm 1.

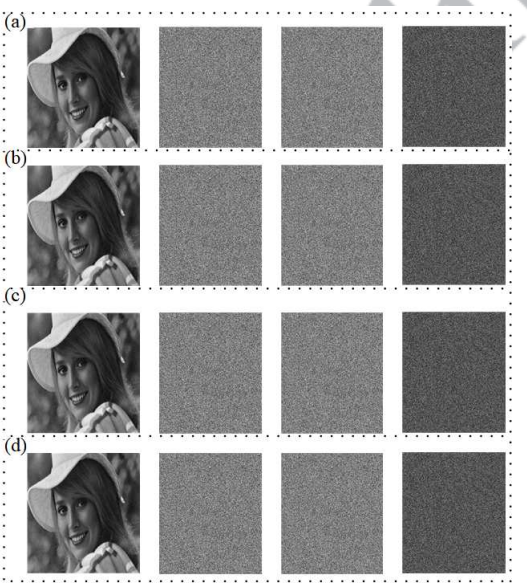


Fig. 14. The sensitivity of the secret key on plaintext elaine, original secret key (osk), changed secret key (csk) and difference between ask and csk. (a) Test of secret key b_ξ. (b) Test of secret key b_ψ. (c) Test of secret key W_{31}. (d) Test of secret key x_0.

In Algorithm 1, we first obtain the size of matrix A according to test image *img_e* and initialize the output matrix B. Then we define the direction of the spiral arrangement (right, down, left, up) and set the move step size to 1. Starting at the pixel point (r,c), we perform the loop operation in spiral arrangement order and store the pixels of matrix A into matrix B. When the current pixel is accessed, we can mark it and calculate the position.

Algorithm 1 Spiral arrangement algorithm.

Input: Pixels matrix of the original frame: *img_e*.
Output: Pixels matrix after the spiral arrangement: *img_s*.

$[M,N] = \text{size}(A)$;
Define the direction of spiral arrangement: right $\rightarrow$ down $\rightarrow$ left $\rightarrow$ up;
direction $= [0\ 1;\ 1\ 0;\ 0\ -1;\ -1\ 0]$;
start $= 1$; $r = 1$; $c = 1$;
Mark the visited points;
$v = \text{false}(M,N)$;
for $i = 1 : M \times N$ **do**
    $B(i) = A(r,c)$;
    Indicates that the current pixel is accessed;
    $v(r,c) = \text{true}$;
    Calculate the next position;
    $\text{next}_r = r + \text{direction}(\text{start},1)$; $\text{next}_c = c + \text{direction}(\text{start},2)$;
    If the boundary is exceeded or visited, change direction;
    if $\text{next}_r < 1 \parallel \text{next}_r > M \parallel \text{next}_c < 1 \parallel \text{next}_c > N \parallel v(\text{next}_r, \text{next}_c)$ **then**
        $\text{start} = \text{mod}(\text{start},4) + 1$; $\text{next}_r = r + \text{direction}(\text{start},1)$; $\text{next}_c = c + \text{direction}(\text{start},2)$;
    end if
    $r = \text{next}_r$; $c = \text{next}_c$;
end for
Await all pixels to complete the spiral arrangement.

In addition, the spiral arrangement algorithm has two kinds of scan scrambling rules. Figure 15(a) shows method 1 of scan scrambling. First, we scramble the original image according to the pixel spiral arrangement algorithm, and then convert it into one-dimensional data. Finally, we reassemble it into an encrypted image. Figure 15(b) shows method 2 of scan scrambling. First, we arrange the original image into a one-dimensional array, and then reassemble this array according to the pixel spiral arrangement algorithm, finally generating the encrypted image. Figure 15(c) shows the scrambling and restoring effects of the remote sensing image according to scan scrambling method 1. Figure 15(d) shows the test effects according to scrambling method 2.

As shown in Figs. 16, the proposed spiral arrangement partitioning algorithm is first divided into four square regions on average. By applying the two scan scrambling methods to these four square regions, the same original image can generate sixteen scrambling methods. If the initial partition region is n, it can generate 2^n scrambling methods. Therefore, the proposed spiral arrangement partitioning algorithm is an irregular arrangement, which can effectively improve the randomness of ciphertext images. This algorithm not only disturbs the original structure of image pixels, but also increases the multiple protection of the original image. Compared with more

Chinese Physics B In Press

complex scrambling algorithms, the proposed spiral arrangement partition scrambling algorithm can provide higher security for the scrambling process without affecting the computational efficiency, and it is particularly well-suited for large-scale data processing, including remote sensing images. As an example, we select a 10×10 pixel region from the center of a remote sensing image to illustrate the spiral arrangement of pixels; the corresponding illustration is reflected in Figs. 17 and 18. On the basis of the PRNs produced by the HNN system, we can design the next diffusion algorithm. The proposed encryption scheme aims to eliminate the correlation characteristics of adjacent pixels and establish a connection between non-adjacent pixel points. Especially in the diffusion strategy section, we construct an adjacent pixel diffusion algorithm by using spatio-temporal encryption methods.

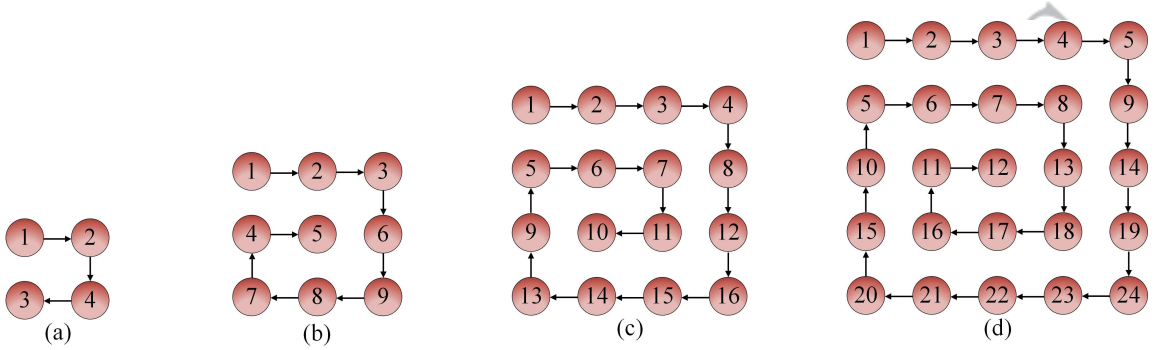


Fig. 15. Spiral arrangement of pixels. (a) 2 × 2. (b) 3 × 3. (c) 4 × 4. (d) 5 × 5.

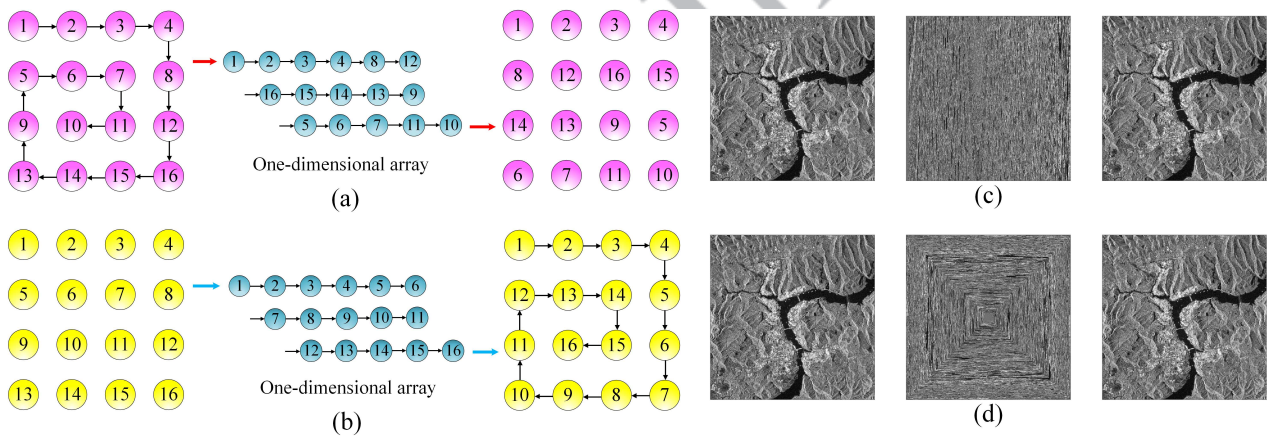


Fig. 16. Scrambling methods for spiral arrangement scan scrambling. (a) The method 1 of spiral arrangement. (b) The method 2 of spiral arrangement. (c) The scrambling effect of method 1. (d) The scrambling effect of method 2.

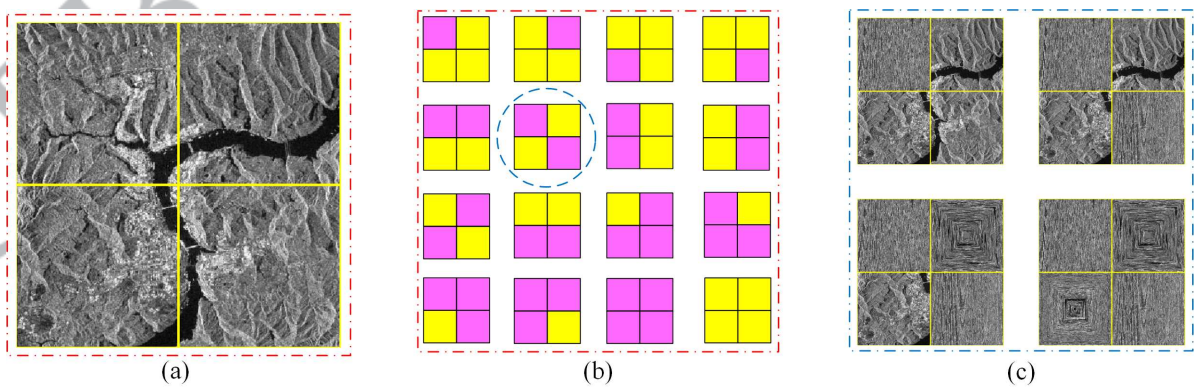


Fig. 17. The spiral arrangement partitioning algorithm. (a) The example diagram of the partitioning method. (b) The generated 2^n scrambling methods. (c) The partition scrambling effect of spiral arrangement.

4.3. Design of diffusion scheme

If the scrambling algorithm changes the positional relationship of pixels, the diffusion algorithm changes the value of the pixel points.

$$\begin{cases} E(i+2) = \mathrm{mod}\left(\mathrm{floor}\left(10000 \times 3.99 \times \frac{E(i)}{255} \times \left(1 - \frac{E(i)}{255}\right)\right) + E(i+1) + R(i) + floor(x \times 10^{14})\right), 256), \\ E(i) = E(i) \oplus y_1(i) \oplus R(i), \end{cases} \tag{8}$$

where $R(i)$ is the pixel after scrambling position, $E(i)$, $E(i+1)$, $E(i+2)$ are three adjacent pixels attached to the diffusion scheme and x is the PRNs from x sequence generated by system (4). Mod is a modulus operation and the floor function is used to round down a number. And then, we perform some pixel point value replacement operations

$$\begin{cases} E(10) = \bmod(R(10) + F_1 + \text{floor}(y \times 10^{14})), \\ R(20) = \bmod(E(10) + F_2 + \text{floor}(y \times 10^{14})), \\ E(30) = \bmod(R(20) + F_3 + \text{floor}(y \times 10^{14})), \\ R(40) = \bmod(E(30) + F_4 + \text{floor}(y \times 10^{14})), \end{cases} \tag{9}$$

where F_1, F_2, F_3 and F_4 are interference parameters, and they are set as $F_1 = 100$, $F_2 = 200$, $F_3 = 300$, $F_4 = 400$ and y is the PRNs from y sequence generated by the proposed HNN system. Then, we have second round of diffusion equation. The specific encryption process is as follows:

Input: The secret key, the plaintext image.

Output: Return the ciphered image.

Step 1 The secret key is generated by the design lock, dual memristor, and the initial state of the system.

Step 2 The plaintext image is scrambled by the design transformation reflected in Algorithm 1.

Step 3 The pixels are diffused by the proposed diffusion algorithm based on the improved spatiotemporal chaos. The encryption and decryption test diagrams are given in Fig. 19.

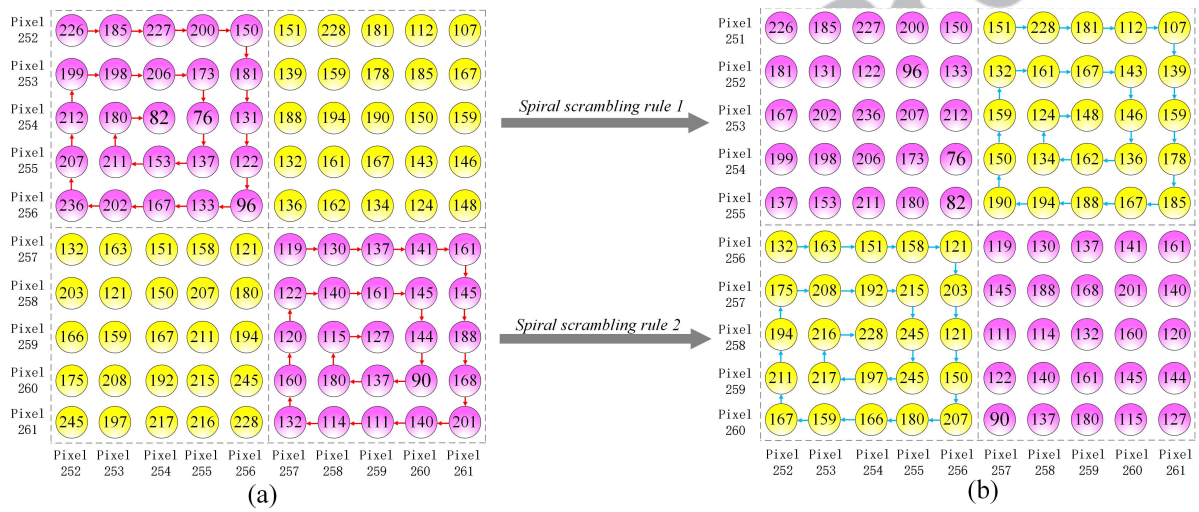


Fig. 18. Graphic description of spiral arrangement partition scrambling process of pixel. (a) The original remote sensing image with 10×10 pixels. (b) The scrambled remote sensing image with 10×10 pixels.

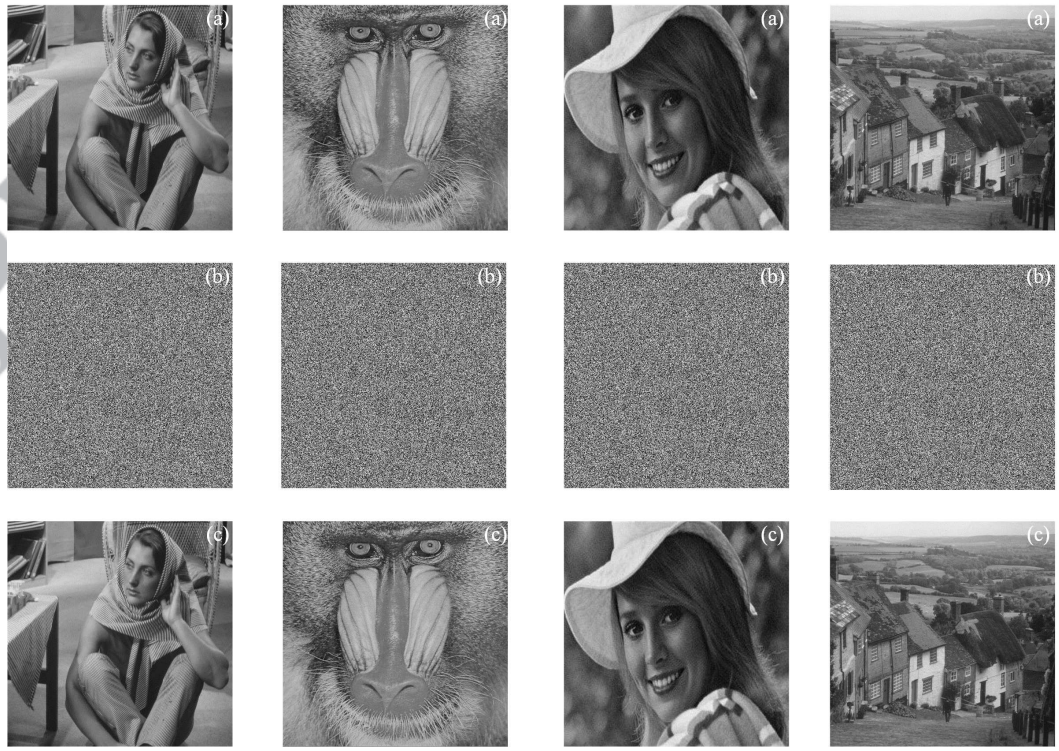


Fig. 19. Test picture. (a) Plaintext picture. (b) Encrypted picture. (c) Decryption picture.

4.4. Histogram and correlation

The histogram is the representation of the distribution of pixels. Relative to the histogram of the plaintext image, the ideal ciphered image should have a uniform and flat distribution of the pixels. It effectively hides the pixel values of the plain image and prevents the attacker from extracting valid information. Conversely, if it is not uniform enough, the attacker can extract valid information from the ciphered image. This greatly reduces the security of the encryption algorithm. The equation for image relevance can be given

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i)(y_i - \frac{1}{N}\sum_{i=1}^{N}y_i)}{\sqrt{\sum_{i=1}^{N}(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i)^2}\sqrt{\sum_{i=1}^{N}(y_i - \frac{1}{N}\sum_{i=1}^{N}y_i)^2}}. \quad (10)$$

Figure 20 presents a test picture histogram. The test graph after the encryption scheme can hide high-frequency information. The corresponding correlation is described in Fig. 21, and Table 3 displays the correlation. We use 5 airport pictures and some classic character pictures to test the correlation. The contrast results fully meet the performance indicators.

Table 3. Correlation test.

Remote Image	Image Size	Plaintext Hor	Plaintext Ver	Plaintext Dia	Ciphertext Hor	Ciphertext Ver	Ciphertext Dia
JiangXi Airport.1	512×512	0.5278	0.6039	0.4628	–0.0001	0.0009	0.0002
JiangSu Airport.2	512×512	0.5255	0.5326	0.4051	–0.0027	0.0032	0.0012
ChongQing Airport.3	512×512	0.7038	0.7639	0.6473	–0.0013	0.0018	0.0007
FuJian Airport.4	512×512	0.8618	0.8739	0.8293	–0.0003	–0.0005	0.0019
FuJian Airport.5	512×512	0.4164	0.5965	0.3627	0.0005	–0.0007	–0.0008
ZheJiang Airport.6	512×512	0.5512	0.5748	0.4423	0.0020	–0.0025	0.0014
Classical Image	**Image Size**	**Plaintext Hor**	**Plaintext Ver**	**Plaintext Dia**	**Ciphertext Hor**	**Ciphertext Ver**	**Ciphertext Dia**
Baboon	512×512	0.8601	0.7541	0.7231	–0.0004	–0.0014	–0.0008
Barb	512×512	0.8591	0.9590	0.8414	–0.0013	0.0016	–0.0004
Boab	512×512	0.9379	0.9718	0.9214	–0.0022	0.0009	–0.0009
Elaine	512×512	0.9855	0.9838	0.9742	–0.0006	–0.0009	–0.0027
Hill	512×512	0.9710	0.9741	0.9520	0.0013	–0.0016	0.0026
Lena	512×512	0.9723	0.9856	0.9593	0.0002	–0.0013	–0.0019
Average	512×512	–	–	–	0.0010	0.0012	0.0015
Avg.of [24]	512×512	0.9909	0.9878	0.9862	–0.0006	0.0003	0.0006
Avg.of [27]	512×512	–	–	–	0.00182	0.00038	0.00138
Avg.of [28]	512×512	–	–	–	0.01248	0.00786	0.01742
Avg.of [29]	512×512	–	–	–	0.0022	–0.0016	0.0110

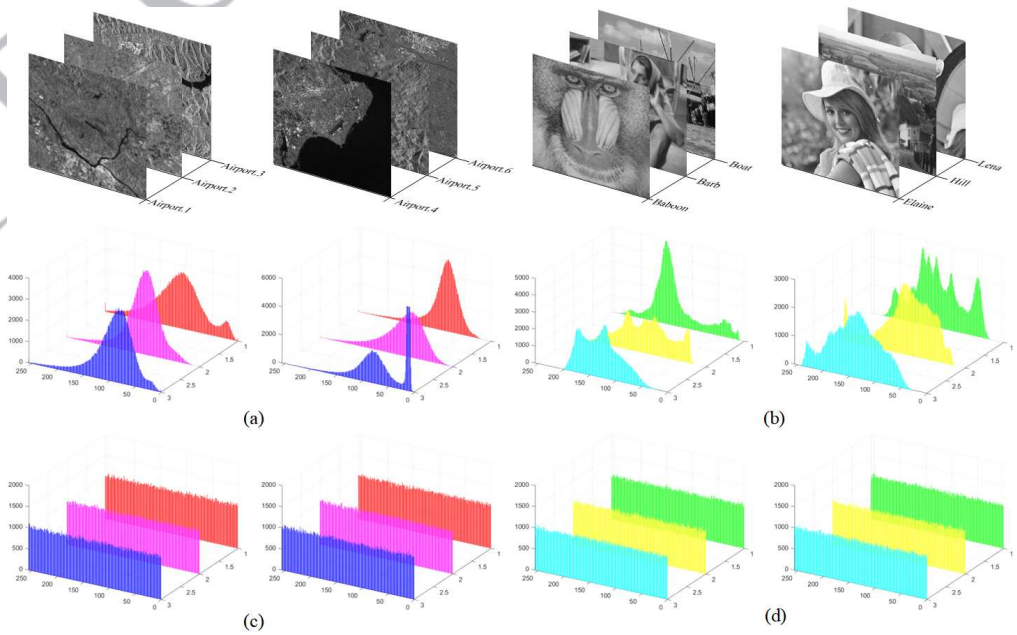


Fig. 20. Histogram. (a) Original remote sensing test images. (b) Original conventional test images. (c) Ciphertext remote sensing images. (d) Ciphertext conventional test images.

CPL Online In Press

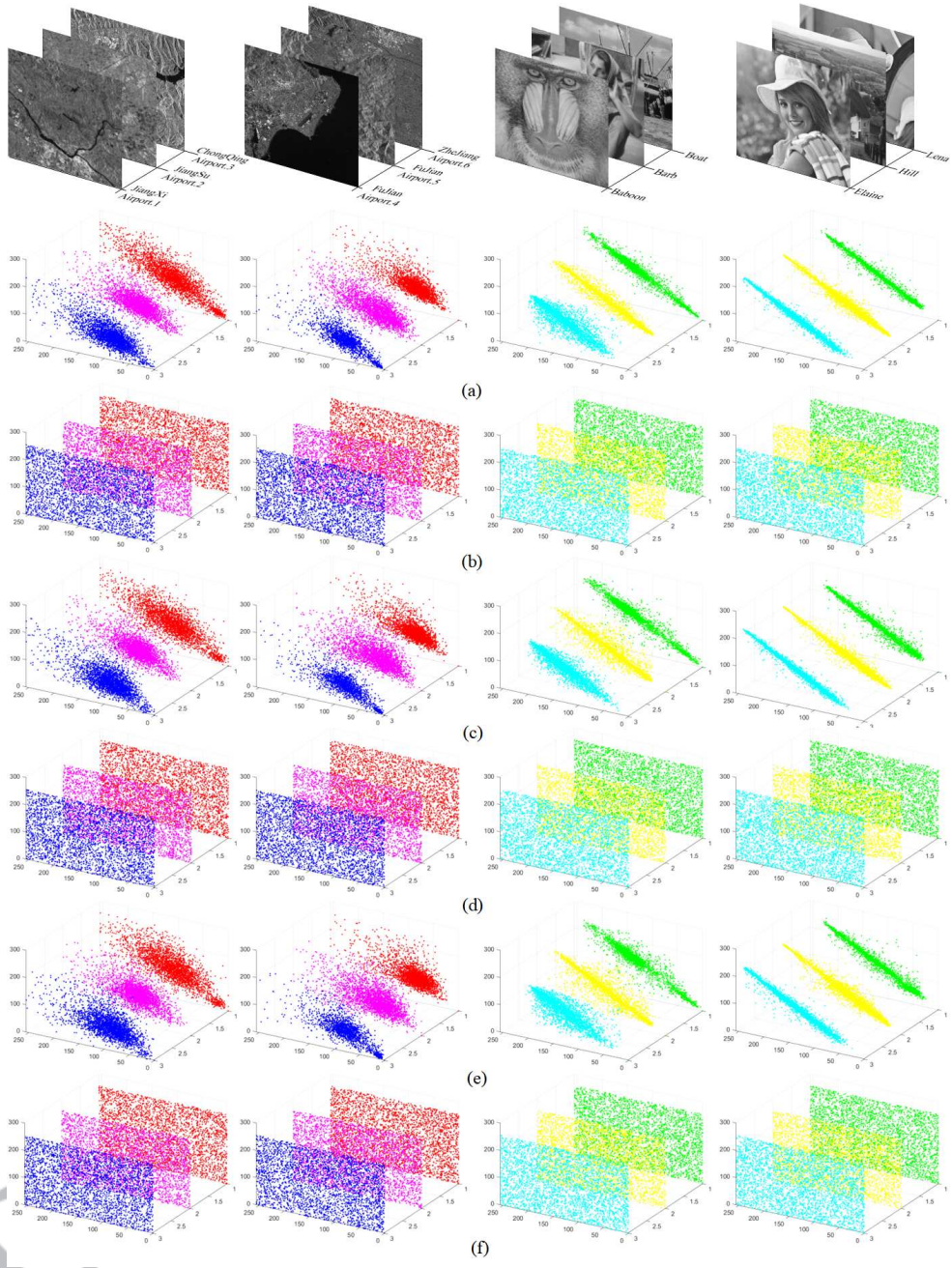


Fig. 21. Correlation distribution graph. (a) Plaintext on H direction. (b) Ciphertext on H direction. (c) Plaintext on V direction. (d) Ciphertext on V direction. (e) Plaintext on D direction. (f) Ciphertext on D direction.

4.5. Robustness testing of encryption algorithms

Information entropy is often used as a quantitative index of information content in a system. It is also a common method to test the security of an encryption algorithm. The specific algorithm is

$$H(k) = \sum_{i=1}^{N} p(k_i) \log_2 \frac{1}{p(k_i)}, \tag{11}$$

where $p(k_i)$ is the probability of si. For a 256 gray-level image, the ideal information entropy is 8. The closer the value of information entropy is to 8, the less possible of the image to show the information and the pixels is closer to randomness. The information entropy of test images are shown in Table 4, and we can obtain the information entropy of the proposed algorithm is closer to the idea value, and it has higher security.

Because of the accuracy of the image encryption algorithm, even a tiny change of pixels in a plain image would cause a huge change in the encryption effect. This change is often measured by the NPCR (number of pixels change rate) and UACI (unified average changing intensity). The specific algorithm is shown in Eq. (12); it also represents the anti-difference ability of the encryption algorithm.

$$\begin{cases} \text{NPCR} = \dfrac{\sum_{i,j} D(i,j)}{L} \times 100\%, \\ \text{UACI} = \dfrac{1}{L} \sum_{i,j} \dfrac{|C(i,j) - C_1(i,j)|}{256} \times 100\%, \end{cases} \tag{12}$$

where C_1, C_2 are the sizes of two images and the size is $W \times H$.

When $C_1(i,j) \neq C_2(i,j)$, we can obtain $D(i,j) = 1$. Oppositely, if $C_1(i,j) = C_2(i,j)$, $D(i,j) = 0$. The average value of NPCR and UACI of five images are shown in Table 5. Relative to the Refs. [33,34], the average value of NPCR and UACI of the proposed algorithm are closer to the ideal values (0.9961 and 0.3346).

4.6. The occlusion attack testing and encrypted time

Noise in actual transmission environments contaminates image data. To test the stability of the algorithm, researchers overlay salt-and-pepper noise with intensities of 1%, 5%, and 10% on four different ciphertext images respectively. Due to the influence of noise, the ciphertext test results contain a certain number of noisy points. However, from the analysis of Fig. 22, it indicates that the core plaintext information of the images is basically preserved. This algorithm effectively adapts to image processing requirements under noise.

Figure 23 presents the results of the occlusion attack. Square blocks with different sizes and letter-shaped masks are used to obscure the cipher images. Although the recovered images contain some distortion after decryption, their main texture and visual content remain recognizable. These results indicate that the proposed encryption algorithm has good tolerance to data loss caused by occlusion attacks.

The complexity of an encryption algorithm can be characterized by the encryption time. As shown in Table 6, due to the use of the spatio-temporal chaos algorithm, the algorithm achieves the characteristics of short time and high security.

Table 4. Information entropy.

Remote image	Image size	Plaintext	Ciphertext
JiangXi Airport.1	512×512	7.195531	7.999428
JiangSu Airport.2	512×512	7.021585	7.999389
ChongQing Airport.3	512×512	7.539411	7.999392
FuJian Airport.4	512×512	6.355809	7.999334
FuJian Airport.5	512×512	7.230801	7.999369
ZheJiang Airport.6	512×512	6.772954	7.999354
Classical image	**Image size**	**Plaintext**	**Ciphertext**
Baboon	512×512	7.145709	7.999420
Barb	512×512	7.464895	7.999334
Boab	512×512	7.213672	7.999365
Elaine	512×512	7.504811	7.999399
Hill	512×512	7.476172	7.999345
Lena	512×512	7.446090	7.999374
Lena[24]	512×512	–	7.9993
Lena[30]	512×512	–	7.9975
Baboon[31]	512×512	–	7.9990
Baboon[32]	512×512	–	7.9994

Table 5. NPCR and UACI test.

Remote Image	Airport 1	Airport 2	Airport 3	Airport 4	Airport 5	Airport 6	Average value	Ref. [33] value
Image size	512×512	512×512	512×512	512×512	512×512	512×512	512×512	512×512
NPCR	99.5884	99.6082	99.6273	99.6262	99.6101	99.5975	99.6096	99.6087
UACI	33.4792	33.4718	33.5320	33.5093	33.4485	33.4681	33.4848	31.4266
Classical Image	Baboon 7	Barb 8	Boat 9	Elaine 10	Hill 11	Lena 12	Average value	Ref. [34] value
Image size	512×512	512×512	512×512	512×512	512×512	512×512	512×512	512×512
NPCR	99.6128	99.6121	99.6037	99.6361	99.6082	99.5853	99.6097	99.5926
UACI	33.4211	33.4106	33.4982	33.5203	33.4016	33.4463	33.4596	30.3921

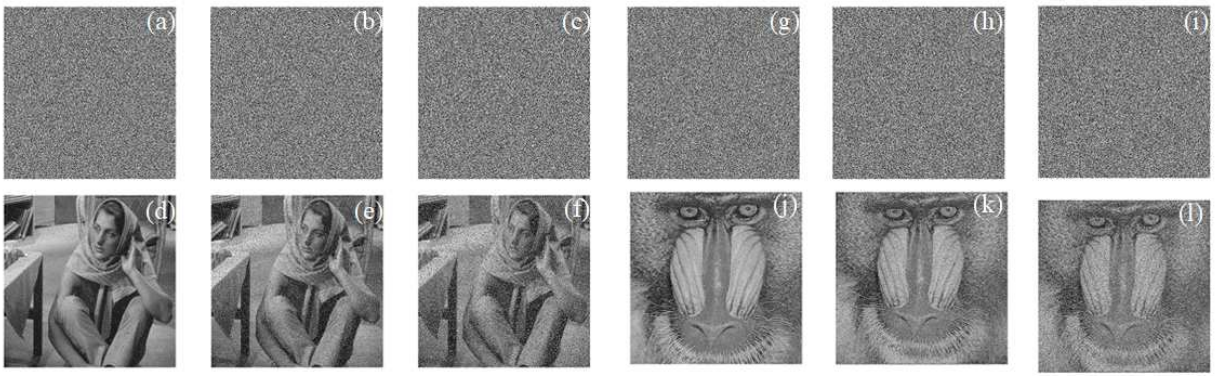


Fig. 22. The result of noise attack. (a) Ciphertext barb with noise intensity 0.01. (b) Ciphertext barb with noise intensity 0.05. (c) Ciphertext barb with noise intensity 0.1. (d) Decryption barb with noise intensity 0.01. (e) Decryption barb with noise intensity 0.05. (f) Decryption barb with noise intensity 0.1. (g) Ciphertext baboon with noise intensity 0.01. (h) Ciphertext baboon with noise intensity 0.05. (i) Ciphertext baboon with noise intensity 0.1. (j) Decryption baboon with noise intensity 0.01. (k) Decryption baboon with noise intensity 0.05. (l) Decryption baboon with noise intensity 0.1.

CPB Online In Press

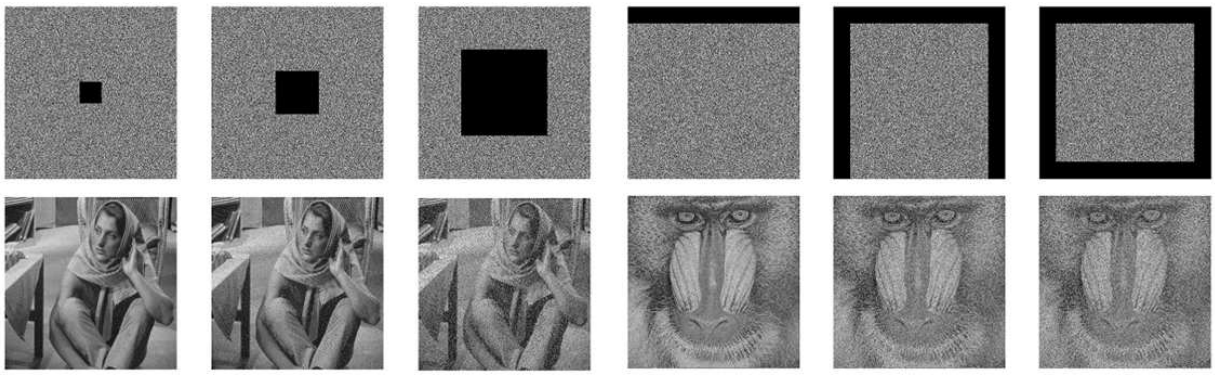


Fig. 23. The result of occlusion attack.

Table 6. Comparison of encryption speed.

Algorithm	CPU	Memory	Video size	AET (ms)
Ref. [24]	Intel Core i5-13490F@2.5 GHz	32 GB	512×512	836
Ref. [25]	Intel Core i5-4210U@1.7 GHz	–	352×192	2652
Ours	Intel Core i5-13490F@2.5 GHz	32 GB	512×512	430

5. Hardware implementation

The experiment selects the TMS320F28335 DSP, which integrates a 150 MHz floating-point operation unit. The iterative calculation of the chaotic system is completed in real time within the DSP. In the experiment, the system parameter was set to an initial value of [0.1, 0.1, 0.1]. Figure 24 shows the scheme flow for DSP-based implementation. And Fig. 25 shows the HNN's attractor realized by the DSP. The results obtained from the circuit experiment are basically consistent with those from the numerical simulation.

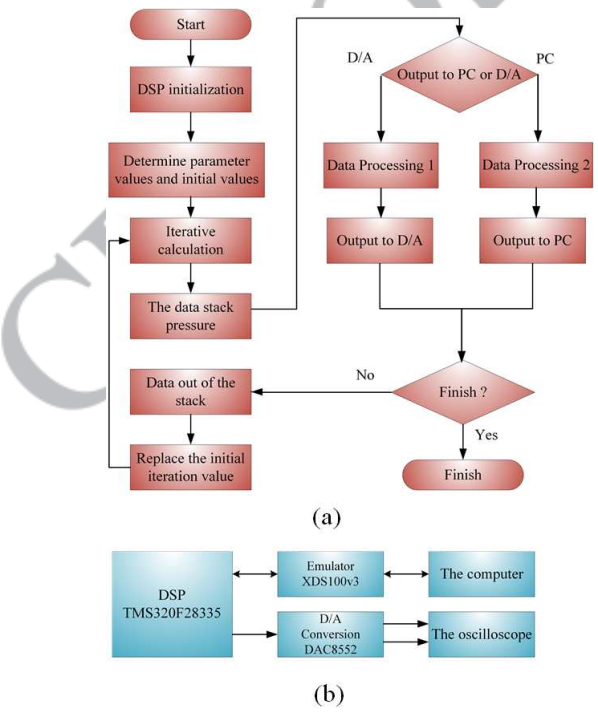


Fig. 24. DSP implementation. (a) DSP implementation flowchart. (b) The schematic circuit diagram.

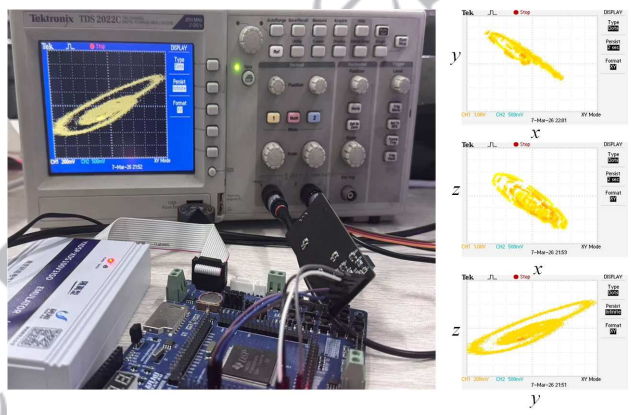


Fig. 25. Attractors obtained in a laboratory environment.

6. Conclusion

In this work, we propose a new HNN model. The new HNN system can produce coexistence attractors with different weights. Dynamic analysis proved that the system has rich dynamic characteristics. There are still some issues worth considering during the study, e.g., apart from bistable behavior, does the system have multi-stable behavior? The mechanism of bistable dynamics generation, whether the proposed dual memristor HNN can generate the chaotic sea. The construction of memristive coexisting HNN attractors provides a direction for the extended research of multistable systems. Next, we will attempt to explore more application prospects of the proposed HNN model.

Acknowledgments

Project supported by the Provincial Department of Science and Technology Project-Joint Program-PhD Research Initiation Project (Grant No. 2025-BSLH-199) and the Fundamental Research Funds for the Liaoning Universities (Grant No. LJ212410146073).

References

[1] Ma C, Banerjee S, Xiong L, *et al.* 2021 *Chin. Phys. B* **30** 120504
[2] He S, Wang H and Sun K 2022 *Chin. Phys. B* **31** 060501
[3] Xu Q, Liu T, Feng C, *et al.* 2021 *Chin. Phys. B* **30** 128702

[4] Rodwell C and Tallapragada P 2022 *Nonlinear Dyn.* **108** 2115
[5] Lai Q, Kuate P, Liu F, *et al.* 2020 *IEEE Trans. Circuits Syst. II Express Briefs* **367** 1130
[6] Ramesh R, Karthikeyan R, Dolvis L, *et al.* 2022 *Chaos Solitons Fractals* **156** 111834
[7] Pratap S, Ankur R and Krishna R 2022 *Eur. Phys. J. Plus* **137** 875
[8] Wang M, Gu L, *et al.* 2024 *Chin. Phys. B* **33** 020504
[9] Ye X, Wang X, Zhao H, *et al.* 2019 *Eur. Phys. J. Plus* **134** 206
[10] Chen M, Wang A, Wang C, *et al.* 2022 *Chaos Solitons Fractals* **160** 112192
[11] Bao H, Chen M, Wu H, *et al.* 2020 *Sci. China Technol. Sci.* **63** 610
[12] Bao B, Tao J, Wang G, *et al.* 2017 *Nonlinear Dyn.* **89** 1160
[13] Bao H, Liu W, Ma J and Wu H 2020 *Int. J. Bifurcation Chaos* **30** 2030029
[14] Lai Q, Lai C, Kuate P, *et al.* 2022 *Int. J. Bifurcation Chaos* **32** 2250042
[15] Wang J, Li P, Banerjee S, *et al.* 2024 *Mob. Netw. Appl.* **29** 620
[16] Feng X, Wu F, Ma J, *et al.* 2025 *Sci. China Technol. Sci.* **68** 2120402
[17] Zhu Z, Dong J, Yu Y, *et al.* 2025 *Sci. China Technol. Sci.* **68** 1920406
[18] Zhang X, Li C, Moroz I, *et al.* 2025 *Eur. Phys. J. Spec. Top.* **113** 15500
[19] Liu W, Li C, Wu J, *et al.* 2026 *Nonlinear Dyn.* **114** 82
[20] Ma T, Mou J and Chen W 2026 *Nonlinear Dyn.* **114** 250
[21] He S, Hu K, Peng Y, *et al.* 2026 *Appl. Intell.* **56** 129
[22] Wang M, Ding J, Li J, *et al.* 2024 *Eur. Phys. J. Plus* **139** 64
[23] Shi F, Xu X, Chen J, *et al.* 2025 *Nonlinear Dyn.* **113** 26760
[24] Wang G and Xu S 2026 *Eur. Phys. J. Plus* **141** 49
[25] Dhingra D and Dua M 2024 *Multimedia Tools Appl.* **83** 1720
[26] Wang G, Xu S and Ye S 2025 *Eur. Phys. J. Plus* **140** 606
[27] Liu P, Wang X and Su Y 2023 *IEEE Trans. Circuits Syst. Video Technol.* **33** 2510
[28] Hosny K, Kamal S, Darwish M and Papakostas G 2021 *Electronics* **10** 1066
[29] Ravichandran D *et al.* 2021 *Med. Biol. Eng. Comput.* **59** 600
[30] Li X, Mou J, *et al.* 2021 *Opt. Laser Technol.* **140** 107074
[31] Abbasi A, Mazinani M and Hosseini R 2020 *Opt. Laser Technol.* **132** 106465
[32] Si R, Hu Q and Yan X 2025 *Phys. Scr.* **100** 045213
[33] Gabr M *et al.* 2023 *IEEE Access* **11** 85010
[34] Alexan W, Alexan N and Gabr M 2023 *Fractal Fract.* **7** 287
[35] Cao C, Sun K and Liu W 2018 *Signal Process.* **143** 128
[36] Wang X, Wang S, Wei N, *et al.* 2019 *IETE Tech. Rev.* **36** 40

CPB Online In-Press