mathematics

MDPI

Article

TCA-Oriented Attractor Editing in a Discrete Memristive Rulkov Neuron with Application to Lightweight IoV Image Encryption

Shan Gao, Hongjie Wang, Weiyao Xu *, Jiayue Qian, Yike Zhou and Yongxin Li

School of Optoelectronic Engineering, Zaozhuang University, Zaozhuang, 277160, China; gaoshan_j1lw@uzz.edu.cn (S.G.); hjwang0909@163.com (H.W.); lrarotaney@163.com (J.Q.); yikezhou_yk@163.com (Y.Z.); yxlinuist@163.com (Y.L.)
* Correspondence: bingbing2016@uzz.edu.cn; Tel.: +86-13561190732

Abstract

The emerging demand for lightweight image security in the Internet of Vehicles (IoV) highlights the need for chaotic sources with programmable dynamical characteristics. This paper develops a three-dimensional discrete memristive Rulkov neuron and proposes a translation-constraint-assignment (TCA) criterion for attractor editing. By partitioning the memristive state space and assigning local gain coefficients to different regions, TCA establishes an explicit mapping among state intervals, local parameters, and attractor properties. Through coordinated translation, regional constraint, and parameter assignment, the proposed criterion enables attractor states to be edited according to prescribed dynamical and spatial requirements within a unified system structure. This mechanism supports the controllable construction of coexisting attractors through regional gain assignments and interval settings, enabling configurable combinations of dynamical states and spatial arrangements. The resulting nonlinear behaviors are analyzed numerically and verified on an STM32 platform. Furthermore, configurable chaotic sequences generated by the edited attractors are combined with SPECK64/128 and bidirectional diffusion to construct a lightweight image encryption scheme. Experimental results demonstrate favorable security, efficiency, and applicability to resource-constrained IoV devices.

Keywords: offset boosting; attractor editing; multistability; lightweight image encryption

MSC: 37G35

1. Introduction

Neurons do not transmit information simply through binary firing or non-firing responses. Instead, their activities are characterized by diverse temporal patterns, including spiking, bursting, periodic oscillations, and chaotic firing [1–3]. These firing modes reflect distinct dynamical states arising from the combined effects of external stimuli, intrinsic parameters, and historical states, and they form an important basis for neuronal information representation and state transitions. From a nonlinear dynamics perspective, such complex activities can be characterized by different attractors in phase space. Accordingly, the study of neuronal models has gradually extended beyond reproducing diverse firing behaviors toward understanding how different dynamical states are formed, coexist, and can be regulated. In particular, when multiple states coexist within the same system, how to further select, configure, and regulate these states becomes an important issue in complex neuronal dynamics [4].

Check for updates

Academic Editor: Andreas Henrici

Received: 29 September 2026
Revised: 24 October 2026
Accepted: 29 October 2026
Published: 5 November 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.

To describe complex neuronal electrical activities, continuous models such as the Hodgkin–Huxley, FitzHugh–Nagumo, and Hindmarsh–Rose models have established mature theoretical frameworks and revealed a variety of dynamical mechanisms associated with excitability, periodic oscillations, and chaotic behaviors [5,6]. However, continuous models generally rely on numerical integration of differential equations, which may introduce relatively high computational and memory costs in digital implementation [7]. In contrast, discrete neuron models describe state evolution directly through iterative mappings and therefore provide more compact mathematical structures while preserving representative neuronal dynamics [8,9]. Among them, the Rulkov model can reproduce typical spiking and bursting behaviors through the interaction between fast and slow variables, making it an important framework for studying discrete neuronal dynamics and digital hardware implementation [10].

On this basis, the incorporation of memristive mechanisms provides additional possibilities for regulating neuronal states. Memristors exhibit pronounced nonlinearity, memory, and state dependence, allowing their internal states to evolve with the system and continuously feed back into neuronal dynamics [11]. Consequently, memristive feedback not only expands the state dimension of neuronal systems but also strengthens the dependence of current firing behaviors on historical states. Existing memristive neuron models have demonstrated rich dynamical phenomena, including multistability, coexisting periodic states, coexisting chaotic states, and complex state transitions, indicating that memristive mechanisms can significantly enhance the capacity of neuronal systems to accommodate multiple dynamical states [12–14]. Meanwhile, the additional internal degree of freedom introduced by the memristive state variable also provides a potential basis for organizing and regulating different firing states within a unified model.

However, the ability to generate multiple coexisting states does not necessarily imply that these states can be configured in a prescribed manner. Existing studies on memristive neurons have mainly focused on the emergence of multistability and coexisting attractors under different parameters or initial conditions, while the dynamical characteristics and spatial distributions of attractors are often governed by the same global system parameters [15–17]. As a result, parameter variations may simultaneously alter attractor morphology, complexity, and phase-space location, making selective regulation of a specific property difficult. Accordingly, the key issue is how to integrate regional dynamical assignment and spatial organization into a unified framework for controllable attractor editing. Building on existing offset-boosting and multistability-editing methods [18–20], this work addresses this issue through an explicit regional feedback construction in a discrete memristive neuron. Such a transition is particularly relevant to neuronal systems, where attractors represent distinct firing states that can be organized according to prescribed dynamical and spatial requirements.

Motivated by this perspective, this paper introduces attractor editing into discrete memristive neurons and proposes a translation-constraint-assignment (TCA) criterion. The proposed criterion partitions the state space through the memristive state variable and assigns corresponding local gains to different regions, thereby establishing a correspondence among memristive state regions, local dynamical parameters, and attractor properties. Through this region-based regulation mechanism, different attractors can be constructed and organized according to predefined rules within a unified system structure, while exhibiting distinct dynamical states. Furthermore, local gains mainly regulate the dynamical characteristics of the attractors, whereas the configuration of state intervals primarily determines their spatial distributions, thereby enabling the coordinated editing of coexisting dynamical states and their spatial organization. In this way, attractor regulation is transformed from conventional global parameter tuning into rule-based editing over

localized state regions, providing a new approach for the configurable construction of coexisting dynamical states in discrete memristive neurons.

The editability of attractors further endows neuronal output sequences with configurable dynamical characteristics. Because different attractors correspond to different long-term evolutionary states, their selection and editing can directly alter the complexity and chaotic properties of the resulting sequences, allowing the neuron model to evolve from a fixed chaotic sequence generator into a configurable complex signal source [21]. This capability introduces additional flexibility for dynamic information encoding and key generation and naturally connects with the demand for lightweight security mechanisms in resource-constrained environments [22]. In particular, image protection in IoV edge devices requires a balance among security, key variability, and computational overhead [23,24]. The compact iterative structure of discrete memristive neurons, together with their configurable chaotic outputs enabled by attractor editing, provides a suitable dynamical basis for lightweight image encryption [25]. Accordingly, the chaotic sequences generated from edited attractors are further employed for dynamic key construction and combined with SPECK64/128 and bidirectional feedback diffusion to realize lightweight grayscale image encryption for IoV applications [26].

The remainder of this paper is organized as follows. Section 2 introduces the three-dimensional discrete memristive Rulkov neuron and the proposed TCA-based attractor editing criterion. Section 3 investigates the nonlinear dynamics, attractor editing behaviors, sequence complexity, and hardware implementation of the system. Section 4 presents the lightweight image encryption scheme for IoV applications and evaluates its performance. Finally, Section 5 concludes the paper.

2. TCA-Guided Memristive Feedback Design for Rulkov Neuron Dynamics

Inspired by Ref. [27], this paper proposes a three-dimensional discrete memristive neuron model, as described by Equation (1):

$$\begin{cases} x_{n+1} = \frac{a}{1+x_n^2} + y_n + d\,[\,b + f(z_n)]x_n \\ y_{n+1} = y_n - cx_n \\ z_{n+1} = z_n + x_n - 0.2f(z_n) \end{cases} \tag{1}$$

Here, d denotes the memristive feedback coefficient, and $f(z_n)$ is the feedback function constructed using the TCA criterion. This construction assigns feedback branches to predefined state intervals, with a separately specified local gain for each interval. This regional parameterization enables local feedback characteristics to be configured and rearranged within a unified model. The differences from existing offset-boosting and multistability-editing methods are summarized in Table 1.

Table 1. Structural comparison of TCA with representative attractor-control methods.

Ref.	Model	Regional Gains	Spacing Control
[18]	3D chaotic flow	×	√
[19]	2D hyperchaotic map	×	√
[20]	5D memristive HNN	×	√
[27]	3D memristive Rulkov map	×	√
This work	3D memristive Rulkov map	√	√

To establish a controllable mapping relationship between memristive state regions and attractor characteristics, a TCA criterion is proposed in this paper. The core idea of

this criterion is to transform the attractor editing problem into a problem of memristive state-space partition and local parameter assignment. By establishing the correspondence among state regions, local gain coefficients, and attractor characteristics, the dynamical properties and spatial distributions of different attractors can be edited. The TCA criterion mainly consists of three steps: translation, constraint, and assignment.

Here, regional gains refer to separately adjustable local gains assigned to predefined state intervals, while spacing control refers to adjusting the separation between coexisting attractors. √ and × indicate features explicitly reported and not explicitly reported, respectively. HNN denotes a Hopfield neural network.

Step A: Translation

The translation operation is performed on the memristive state variable z_n as $z_n + \Delta_i$, where the translation offset set is defined as:

$$\Delta_i \in \{3D,\ D,\ -D,\ -3D,\ -5D\} \tag{2}$$

where $D > 0$ represents the interval width parameter. By selecting different translation offsets, the memristive state regions corresponding to different initial states can be mapped into predefined ranges, providing the basis for subsequent state-space partitioning.

Step B: Constraint

To establish a unified reference state space, the translated memristive variable is constrained as follows:

$$z_n + \Delta_i \in [-D,\ D) \tag{3}$$

Based on the constraint in Equation (3), the state space is partitioned into the five subintervals defined in Equation (4).

$$I_1 = [-4D,\ -2D),\ I_2 = [-2D,\ 0),\ I_3 = [0,\ 2D),\ I_4 = [2D,\ 4D),\ I_5 = [4D,\ 6D) \tag{4}$$

Through the state-space partition, different memristive state regions are associated with corresponding subintervals, providing regional foundations for the assignment of local dynamical parameters.

Step C: Assignment

The local gain coefficients k_i are introduced in different state regions to define the memristive feedback function:

$$f(z_n) = k_i \cdot (z_n + \Delta_i),\ z_n \in I_i \tag{5}$$

where k_i denotes the local gain coefficient corresponding to the i-th state region. Since the memristive feedback term is modulated by $f(z_n)$, the variation of k_i changes the feedback intensity within the corresponding state region. When the memristive state variable falls outside the predefined five active piecewise regions, the feedback function is set to zero, and this region is referred to as the dead zone. Based on the local representation in Equation (5), the complete piecewise memristive feedback function is formulated as Equation (6).

Collectively, the three operations determine the position, width, and local feedback strength of each active region. The resulting piecewise configurations for different D and k_i settings are illustrated in Figure 1.

$$f(z_n) = \begin{cases} k_1(z_n + 3D), & -4D \le z_n < -2D \\ k_2(z_n + D), & -2D \le z_n < 0 \\ k_3(z_n - D), & 0 \le z_n < 2D \\ k_4(z_n - 3D), & 2D \le z_n < 4D \\ k_5(z_n - 5D), & 4D \le z_n < 6D \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

To clarify the local dynamical structure of the TCA construction, consider the feedback branch associated with the i-th active interval. Its feedback function and local coordinate are defined as follows:

$$f_i(z_n) = k_i(z_n + \Delta_i), \quad u_n = z_n + \Delta_i \tag{7}$$

For evolution confined to this interval, substitution into the state equations yields:

$$\begin{cases} x_{n+1} &= \frac{a}{1+x_n^2} + y_n + d(b + k_i u_n)x_n \\ y_{n+1} &= y_n - cx_n \\ u_{n+1} &= u_n + x_n - 0.2k_i u_n \end{cases} \tag{8}$$

In local coordinates, each branch therefore retains the dynamical form of the unshifted model with gain k_i. The translation determines the branch location, while the assigned gain specifies its local feedback characteristics. This correspondence applies while the trajectory and its local perturbations remain within the associated interval.

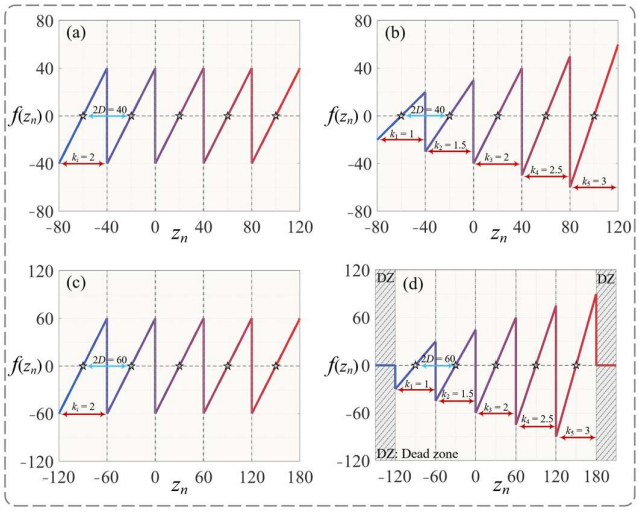


Figure 1. Schematic diagram of the dynamical editing principle based on the TCA criterion. (**a**) reference configuration with uniform local gains and fixed interval spacing; (**b**) region-dependent local gain assignment with unchanged interval spacing; (**c**) increased interval spacing with uniform local gains; (**d**) combined adjustment of interval spacing and local gains, with dead zones outside the active regions.

When the state moves from interval i to interval ℓ, the next-state coordinate is defined using the translation of the new interval, giving:

$$u_{n+1} = z_{n+1} + \Delta_\ell = u_n + x_n - 0.2k_i u_n + \Delta_\ell - \Delta_i \tag{9}$$

To examine the regional dynamical behavior associated with the above construction, numerical checks were performed for representative configurations. Within the observation window, the tested representative trajectories exhibited no branch switches or departures from their initially assigned regions.

In addition, initial-condition perturbation tests were conducted around the five regional baseline states. Most perturbed trajectories converged to the same dominant regions as their corresponding baselines, although a few isolated points exhibited transitions to neighboring regions. These results provide numerical evidence of predominantly robust regional attraction within the tested neighborhoods.

Figure 1a presents the reference configuration with fixed D and uniform k_i. Figure 1b,c independently vary k_i and D, respectively, based on this configuration, while Figure 1d varies both and additionally illustrates the dead zones outside the active regions.

The local gain coefficients in Figure 1a are uniformly set to $k_i = 0.07$ to examine its memristive behavior. As shown in Figure 2, the corresponding v-i responses exhibit typical origin-crossing pinched hysteresis loops with clear amplitude- and frequency-dependent evolution, confirming the memristive characteristics of the constructed feedback function.

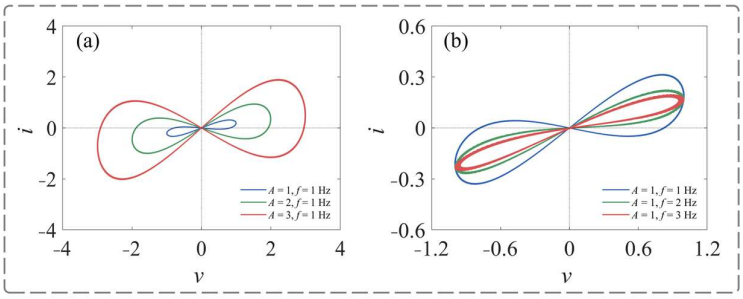


Figure 2. Pinched hysteresis loops obtained using the piecewise feedback function in Equation (6) with $k_i = 0.07$ and $z_0 = 0$: (**a**) varying excitation amplitude; (**b**) varying excitation frequency.

3. Dynamical Editing and Hardware Validation

3.1. Feedback-Gain-Induced Dynamical Regime Transition

To identify the dynamical regimes associated with different gain values, Equation (1) is first analyzed under the unshifted linear feedback $f(z_n) = kz_n$, where k provides the reference value for the uniform local-gain assignment $k_i = k$ in the subsequent TCA construction.

The influence of k on the system dynamics is characterized using the bifurcation diagram and Lyapunov exponent spectra. With $a = 15$, $b = 0.2$, $c = 0.4$, and $d = 0.6$, and the initial state fixed at $(x_0, y_0, z_0) = (1, 1, 0)$, k is varied over the interval $[0.02, 0.07]$. As shown in Figure 3, variations in k induce transitions among periodic, chaotic, and hyperchaotic regimes over different parameter intervals.

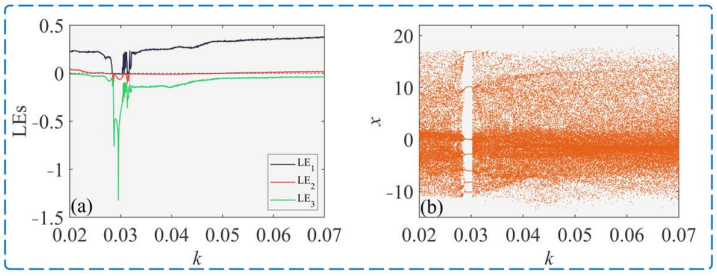


Figure 3. Dynamical evolution of the map (1) with $a = 15$, $b = 0.2$, $c = 0.4$, $d = 0.6$, under $(x_0, y_0, z_0) = (1, 1, 0)$, where k varies in $[0.02, 0.07]$: (**a**) Lyapunov exponents; (**b**) bifurcation diagram.

The full Lyapunov spectrum shown in Figure 3a is computed in double precision using the analytical Jacobian matrix and QR orthogonalization at each iteration. For each parameter value, 30,000 iterations are performed, with the first 5000 discarded as transients. The three Lyapunov exponents are obtained by averaging the logarithms of the absolute diagonal entries of the upper triangular matrices over the remaining 25,000 iterations.

For the corresponding bifurcation diagram, each parameter case starts from the same initial state $(1, 1, 0)$. The system is iterated for 2000 steps, and the final 100 values of x are retained for plotting.

To further characterize the dynamical states, the Kaplan-Yorke dimension is calculated from the Lyapunov exponents ordered from largest to smallest. For the representative parameter values $k = 0.02$, 0.03, 0.04, 0.05, and 0.07, the estimated dimensions are 3, 0,

3, 3, and 3, respectively. The zero estimate corresponds to the stable periodic response at $k = 0.03$.

To further characterize these representative dynamical states, the attractor structures and temporal responses corresponding to selected k values are presented in Figure 4. The phase portraits provide a direct visualization of the evolution of attractor geometry, while the temporal responses further confirm the differences in oscillatory patterns associated with different dynamical regimes.

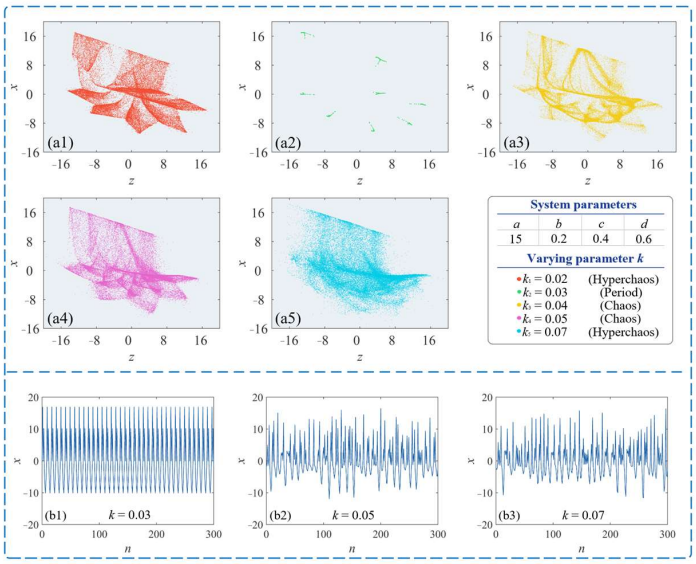


Figure 4. Typical dynamical behaviors of map (1) with fixed parameters $a = 15$, $b = 0.2$, $c = 0.4$, $d = 0.6$ under $(x_0, y_0, z_0) = (1, 1, 0)$.

3.2. TCA-Based Attractor Editing and Complexity Characterization

As shown in Figure 5, the TCA criterion achieves the coexistence of multiple attractors through memristive state-space partitioning and local gain configuration. Unlike conventional multistability induced solely by initial conditions, the proposed approach introduces an explicit correspondence between memristive state regions and attractor characteristics. Different initial states z_0 drive the system into different memristive state regions, where the corresponding local gain coefficients k_i further regulate the dynamical characteristics of the associated attractors. Based on this mechanism, adjusting the local gain coefficients k_i can modify the feedback intensity within the corresponding regions, thereby regulating the dynamical properties of attractors. Meanwhile, varying the interval width D changes the mapping relationships among different memristive state regions, thereby adjusting the spatial distances between coexisting attractors in phase space.

To verify the effectiveness of the TCA criterion for attractor editing and coexistence, the piecewise feedback function $f(z_n)$ defined in Equation (6) is adopted. The parameters are fixed at $a = 15$, $b = 0.2$, $c = 0.4$, and $d = 0.6$. For the basin-of-attraction analysis, the initial state is set to $(x_0, y_0, z_0) = (x_0, 1, z_0)$, with x_0 and z_0 varied over the prescribed ranges. For the phase portraits and temporal responses, the initial state is set to $(x_0, y_0, z_0) = (1, 1, z_0)$, with z_0 selected from different feedback regions. The corresponding basins of attraction, phase portraits, and temporal responses are analyzed, as shown in Figure 6.

Under the baseline configuration, D is set to 20, and the local gain coefficients are assigned as $(0.02, 0.03, 0.04, 0.05, 0.07)$. By varying the initial state z_0, the system enters different piecewise regions and obtains five coexisting attractors. As shown in Figure 6a–c, the temporal responses, x-z phase portraits, and basins of attraction corresponding to different initial conditions exhibit distinct characteristics. These results demonstrate that the

TCA criterion can effectively construct multiple coexisting attractor states under different initial conditions z_0.

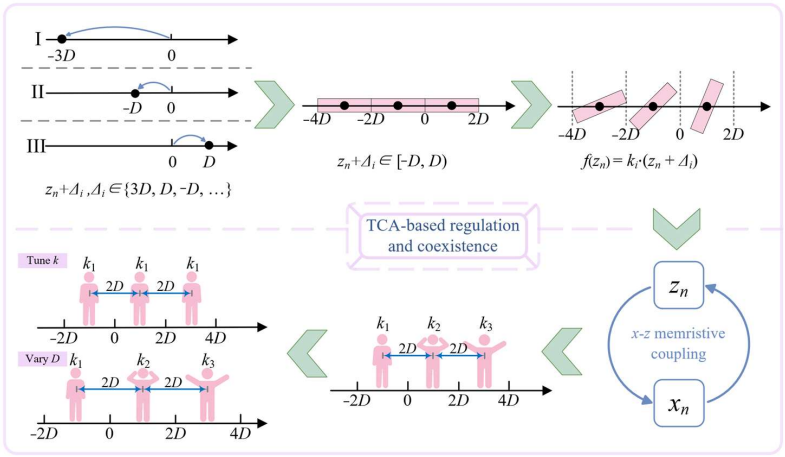


Figure 5. Schematic illustration of the TCA-based attractor construction and editing mechanism.

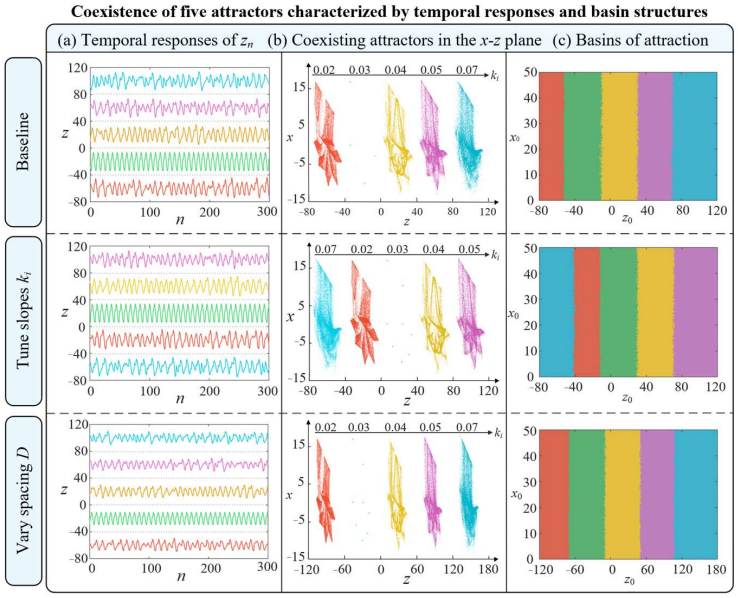


Figure 6. TCA-based editing of five coexisting attractors: (**a**) temporal responses, (**b**) phase portraits, and (**c**) basins of attraction for the baseline, slope-tuned, and spacing-varied configurations.

Furthermore, the local gain coefficients k_i and the interval width D are adjusted separately to investigate their respective regulation effects. When the local gain coefficients are reassigned as (0.07, 0.02, 0.03, 0.04, 0.05), the trajectory structures and temporal responses of the attractors change significantly, whereas their spatial arrangements in phase space remain nearly unchanged. This result indicates that the local gain coefficients k_i mainly affect the dynamical properties of attractors, while having little influence on their spatial distributions.

Tables 2 and 3 quantify spatial and dynamical regulation through regional mean z positions and maximum finite-time Lyapunov exponents, respectively. Results are ordered by state interval in Table 2 and by local gain in Table 3. The interval-wise gains are (0.02, 0.03, 0.04, 0.05, 0.07) for Baseline and Vary spacing, and (0.07, 0.02, 0.03, 0.04, 0.05) for Tune slopes.

Table 2. Regional mean positions across editing configurations.

State Interval	Baseline	Tune Slopes	Vary Spacing
$[-4D, -2D)$	−60.0012	−59.9998	−90.0040
$[-2D, 0)$	−20.0000	−20.0044	−30.0000
$[0, 2D)$	20.0018	20.0000	29.9978
$[2D, 4D)$	59.9997	60.0020	90.0028
$[4D, 6D)$	100.0027	99.9993	150.0010

Table 3. Maximum Lyapunov exponents across editing configurations.

k_i	Baseline	Tune Slopes	Vary Spacing
0.02	0.2229	0.2289	0.2326
0.03	−0.0068	−0.0068	−0.0068
0.04	0.2597	0.2497	0.2583
0.05	0.3255	0.3398	0.3333
0.07	0.3745	0.3817	0.3744

With the baseline gains retained, increasing D from 20 to 30 increases adjacent mean-position spacing from approximately 40 to 60 (Table 2), while the maximum finite-time Lyapunov exponents at matching gains remain broadly similar (Table 3). Thus, D adjusts spatial spacing while approximately preserving the gain-dependent dynamical indicator.

The complexity of chaotic sequences reflects the richness and irregularity of the dynamics generated by a chaotic model. To quantitatively evaluate the generated sequences, spectral entropy (SE) and C_0 complexity are employed [28]. As defined in Equations (10) and (11), SE measures the distribution uniformity of spectral energy through the normalized spectral probability p_s, whereas C_0 complexity evaluates the relative energy of the irregular component by comparing the original sequence $x(n)$ with its reconstructed component $x_r(n)$.

$$\mathrm{SE} = -\frac{1}{(\ln(N))} \sum_{s=1}^{N} p_s \ln(p_s) \tag{10}$$

$$C_0 = \frac{\sum_{n=1}^{N} (x(n) - x_r(n))^2}{\sum_{n=1}^{N} x^2(n)} \tag{11}$$

Larger values generally indicate higher sequence complexity. Figure 7 presents the distributions of SE and C_0 complexity in the (k, d) parameter plane. The two measures exhibit similar distribution patterns, with largely overlapping high-complexity regions. In particular, k significantly influences the complexity distribution, whereas d mainly regulates the extent of the high-complexity regions.

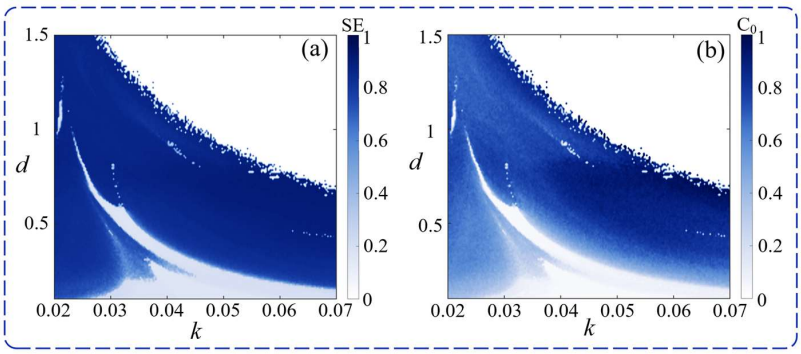


Figure 7. Distributions of SE and C_0 complexity in the (k, d) parameter plane. (**a**) SE; (**b**) C_0 complexity.

3.3. Hardware Validation of TCA-Enabled Attractor Editing

To verify the feasibility of hardware implementation of the proposed three-dimensional discrete memristive Rulkov neuron model and the TCA criterion, an experimental platform based on the STM32H750VBT6 (STMicroelectronics, Geneva, Switzerland) microcontroller unit (MCU) is constructed. The system state variables are iteratively updated on the MCU, and the transient states of the first 500 iterations are discarded. The steady-state variables are then quantized and output through dual-channel digital-to-analog converters (DACs) to obtain the x-z phase portraits, enabling hardware verification of the coexisting attractors selected by the TCA criterion, as shown in Figure 8.

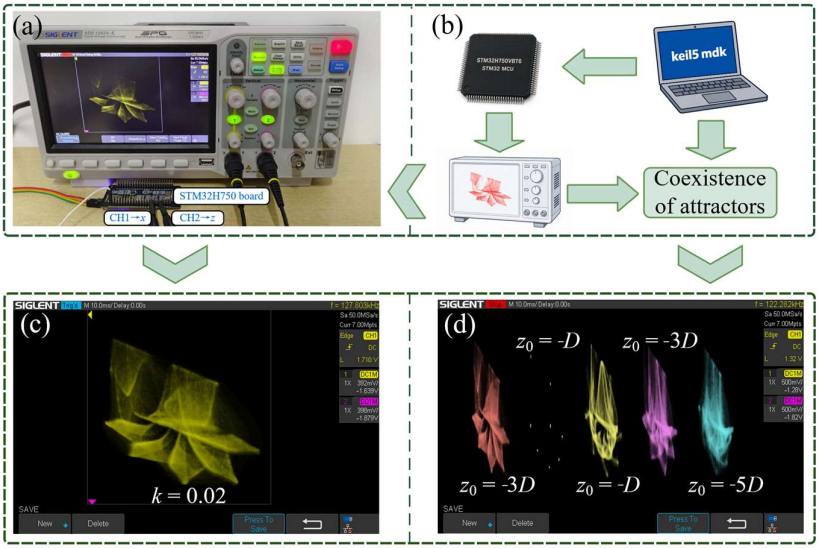


Figure 8. Hardware implementation of the TCA-regulated memristive neuron model on the MCU. (**a**) Experimental platform. (**b**) Implementation flow. (**c**) Attractor trajectory for $k = 0.02$. (**d**) Coexistence of five attractors under different initial memristive states z_0.

Experimental results demonstrate that the hardware-generated trajectories are consistent with the numerical simulation results. Under fixed system parameters and local gain configurations, multiple coexisting attractors can be stably obtained on the hardware platform by only changing the initial memristive state z_0. This verifies the feasibility of implementing the TCA criterion on digital hardware and the consistency between the hardware realization and numerical analysis. Furthermore, the experiment demonstrates the potential of the proposed model and editing method for digital implementation.

4. Chaos-Driven Lightweight Image Security Framework for IoV Applications

4.1. IoV-Oriented Image Security Scenario and Protection Strategy

Digital transformation supports innovation across industrial applications, including green innovation and circular manufacturing [29]. With the development of intelligent transportation systems and IoV technologies, onboard cameras, driving recorders, and roadside monitoring devices continuously generate a large amount of image data containing vehicle appearance, license plate information, and road environments. These images may suffer from unauthorized access and privacy leakage during onboard storage, edge processing, or wireless transmission. Meanwhile, vehicle-mounted and roadside embedded devices are generally constrained by limited computational capability, storage capacity, and power consumption. Therefore, lightweight image encryption methods with both security and efficiency are highly demanded.

To address these requirements, this work exploits the complex nonlinear dynamical characteristics generated by the discrete memristive neuron model and proposes a chaos-driven image encryption scheme for IoV scenarios. As shown in Figure 9, image data collected by onboard terminals, edge nodes, and roadside devices may face security threats during acquisition, storage, and transmission.

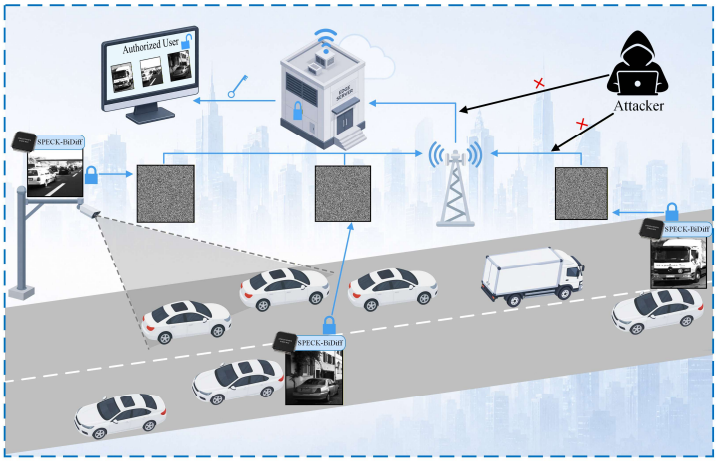


Figure 9. Application scenario of image encryption in IoV environments.

The proposed scheme generates a dynamic master key using the chaotic model and integrates the lightweight SPECK cryptographic algorithm with the proposed SPECK-BiDiff structure to achieve secure protection of image data in IoV environments.

4.2. Memristive Chaos-Driven Dynamic Key Scheduling

For a grayscale image of size $H \times W$, the pixel position is denoted as (h, w), where h represents the image row index ranging from 0, 1, ..., $H - 1$, and w represents the image column index ranging from 0, 1, ..., $W - 1$. The entire process consists of transient state discarding, row-wise iterative sampling, state normalization and quantization, 128-bit master key construction, and SPECK round-key expansion, as illustrated in Figure 10.

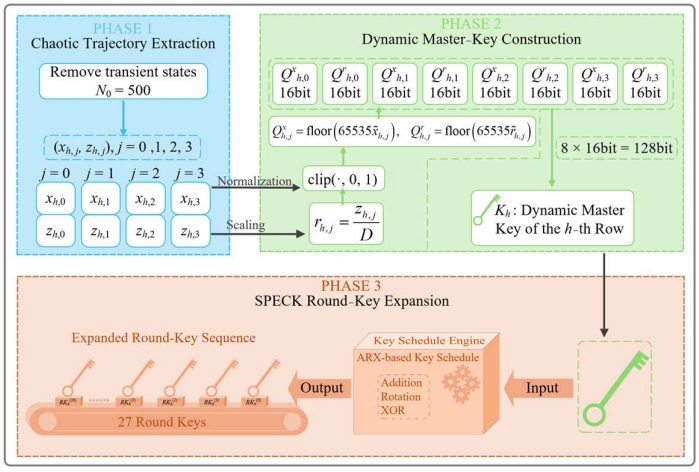


Figure 10. Workflow of chaotic dynamic master-key generation and SPECK round-key expansion.

Specifically, after discarding the first 500 transient iterations of the system, the dynamic master keys are sequentially generated according to the image row order based on the steady-state trajectories. For the h-th row, four groups of state samples are extracted from the chaotic state sequence and denoted as $(x_{h,j}, z_{h,j})$, $j = 0, 1, 2, 3$, where j represents the sampling

index during the master key generation process of the h-th row. To reduce the short-term correlation between adjacent chaotic samples, the sampling interval between neighboring state samples is fixed as $T_s = 16$. This interval is adopted as an empirical design choice to avoid directly using consecutive states while maintaining a low computational overhead.

According to the mapping relationship between memristive state regions and local dynamical parameters established by the TCA criterion, a memristive state region exhibiting relatively high-complexity dynamical characteristics is selected as the source for dynamic key generation. Specifically, $z_0 = 5D$ is selected, corresponding to the local gain coefficient $k_5 = 0.07$, to obtain the chaotic sequence used for key construction.

Since the boundaries of the piecewise regions in the TCA criterion are determined by the interval width parameter D, the normalized variable $r_{h,j} = z_{h,j}/D$ is introduced to maintain consistency between the memristive state representation and the TCA interval scale. For $z_{h,j} \in [-4D, 6D)$, the normalized variable satisfies $r_{h,j} \in [-4, 6)$.

According to the typical value ranges of the system state variables, the lower and upper bounds for normalization are predefined as $x \in [-20, 20)$. The state variables $x_{h,j}$ and $r_{h,j}$ are then linearly normalized as follows:

$$\tilde{x}_{h,j} = \mathrm{clip}\left(\frac{x_{h,j} + 20}{40}, 0, 1\right),\ \tilde{r}_{h,j} = \mathrm{clip}\left(\frac{r_{h,j} + 4}{10}, 0, 1\right) \tag{12}$$

Here, $\mathrm{clip}(\cdot, 0, 1)$ denotes the operation that restricts the normalized results to the interval $[0, 1]$, where $\cdot$ represents the input variable. Specifically, values smaller than 0 are set to 0, values larger than 1 are set to 1, and all other values remain unchanged.

The normalized state variables are subsequently quantized into 16-bit unsigned integers:

$$Q^x_{h,j} = \mathrm{floor}\left(65,535\tilde{x}_{h,j}\right),\ Q^r_{h,j} = \mathrm{floor}\left(65,535\tilde{r}_{h,j}\right) \tag{13}$$

In the equation, floor denotes the floor operation. Each group of state samples generates two 16-bit quantized values with a range of 0 to 65,535.

For the h-th row, the quantized values obtained from the four sampling instants are concatenated according to a predefined order to generate the dynamic master key for the corresponding row:

$$K_h = Q^x_{h,0} \| Q^r_{h,0} \| Q^x_{h,1} \| Q^r_{h,1} \| Q^x_{h,2} \| Q^r_{h,2} \| Q^x_{h,3} \| Q^r_{h,3} \tag{14}$$

Here, the symbol $\|$ denotes the binary concatenation operation. Since each quantized value occupies 16 bits, the eight quantized values jointly form a 128-bit dynamic master key K_h.

Specifically, each 16-bit quantized value is converted into two 8-bit bytes according to the little-endian byte order, where the lower byte is placed before the higher byte. Following the above arrangement order of the quantized values, a 16-byte SPECK64/128 master key is obtained. For a grayscale image with $H = 256$, 256 row-wise dynamic keys $\{K_0, K_1, \ldots, K_{255}\}$ are generated. After obtaining the 128-bit dynamic master key K_h for the h-th row, the SPECK64/128 key expansion algorithm is employed to generate the corresponding round keys:

$$RK_h = \left\{RK_h^{(0)},\ RK_h^{(1)},\ \ldots,\ RK_h^{(26)}\right\} \tag{15}$$

Here, $RK_h{}^{(t)}$ denotes the 32-bit round key used in the t-th round for the h-th row, where $t = 0, 1, \ldots, 26$. Therefore, each 128-bit dynamic master key is expanded into a set of 27 32-bit round keys.

Finally, the row-wise round-key set RK_h is obtained. These round keys are subsequently used for SPECK encryption of the 64-bit data blocks within the corresponding image row during the bidirectional feedback-coupled diffusion process.

4.3. Feedback-Coupled Bidirectional Diffusion Mechanism

Based on the dynamic key generation process, this work introduces two-stage ciphertext feedback processes with opposite propagation directions to construct a SPECK-based bidirectional feedback-coupled image encryption mechanism. The proposed mechanism sequentially performs forward feedback fusion, the first SPECK encryption, backward feedback fusion, and the second SPECK encryption, thereby establishing continuous dependencies among different image data blocks. The overall encryption procedure is illustrated in Figure 11. In the figure, block representations are adopted to describe the data transmission relationships, whereas the actual encryption process divides the input image into multiple 64-bit data blocks.

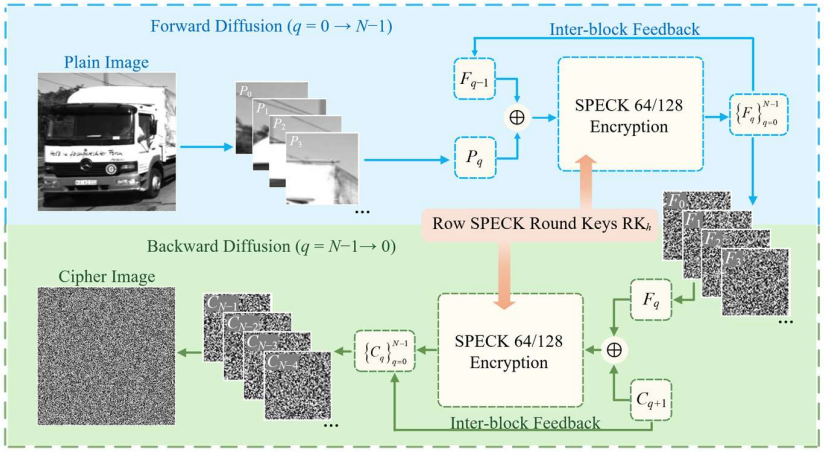


Figure 11. Schematic of the proposed SPECK-BiDiff encryption process.

For an 8-bit grayscale image of size 256 × 256, $H = W = 256$, and the image is divided into $N = HW/8 = 8192$ 64-bit blocks. Because W is divisible by 8, each block is completely contained within one image row and is processed using the corresponding row key.

The image byte sequence arranged in a row-wise order is divided into several 64-bit data blocks, with the q-th plaintext block denoted as P_q, where

$$q = 0,\ 1,\ \ldots,\ N-1,\ N = \frac{HW}{8} \tag{16}$$

The image row index h corresponding to the q-th data block is determined as follows:

$$h = \text{floor}\left(\frac{8q}{W}\right) \tag{17}$$

Therefore, the q-th data block is encrypted using the SPECK round keys generated for the h-th row in the previous section. For the 256 × 256 grayscale images adopted in this work, the pixel data are divided into 8192 64-bit blocks, with 32 blocks corresponding to each row. Accordingly, the row index is calculated as $h = \text{floor}\ (q/32)$.

For the q-th 64-bit input block entering the SPECK64/128 encryption process during the feedback operation, the block is divided into two 32-bit words according to the little-endian byte order. The two 32-bit words at the beginning of the t-th round are denoted as ${u_q}^{(t)}$ and ${v_q}^{(t)}$, where ${u_q}^{(t)}$ represents the first four bytes at the lower address and ${v_q}^{(t)}$ represents the remaining four bytes at the higher address.

If the q-th block belongs to the h-th row of the image, the corresponding round-key set RK_h is employed to perform 27 encryption rounds. In the t-th round, the two 32-bit intermediate words are updated as follows:

$$\begin{cases} u_q^{(t+1)} = [\text{ROR}_8(u_q^{(t)}) + v_q^{(t)}] \bmod 2^{32} \oplus \text{RK}_h^{(t)} \\ v_q^{(t+1)} = \text{ROL}_3(v_q^{(t)}) \oplus u_q^{(t+1)},\ t = 0,\ 1,\ \ldots,\ 26 \end{cases} \tag{18}$$

Here, $\mathrm{ROR}_8(\cdot)$ and $\mathrm{ROL}_3(\cdot)$ denote the 32-bit word-wise right rotation by 8 bits and left rotation by 3 bits, respectively. The symbol $\oplus$ represents the bitwise XOR operation, and $\mathrm{mod}2^{32}$ indicates that only the lower 32 bits are retained after the operation. After completing 27 encryption rounds, the two 32-bit output words are recombined according to the original byte order to obtain the 64-bit output block. For simplicity, the SPECK64/128 encryption and decryption operations using the round-key set of the h-th row are denoted as $E_h(\cdot)$ and $D_h(\cdot)$, respectively.

To initialize the forward and backward feedback processes, two 64-bit initial feedback values, IV_f and IV_b, are introduced for the forward and backward directions, respectively. These two initial feedback values are predefined shared system parameters between the encryption and decryption sides. To uniformly represent the initial states of the two feedback chains, the following definitions are introduced:

$$F_{-1} = IV_f, \;\; C_N = IV_b \tag{19}$$

In the forward feedback stage, the plaintext blocks are sequentially processed according to the forward order of image data blocks, where feedback fusion and SPECK encryption are performed successively. For the q-th plaintext block, it is first XORed with the previous forward intermediate state block, and then encrypted using the SPECK round-key set corresponding to the row of the current block, as follows:

$$F_q = E_h\left(P_q \oplus F_{q-1}\right), \; q = 0, 1, \ldots, N-1 \tag{20}$$

Here, F_q denotes the 64-bit data after forward feedback fusion and SPECK encryption, and F_{q-1} represents the forward intermediate state generated by the previous block.

Starting from the second block, the forward intermediate state generated by the previous block is continuously fed back into the current plaintext block. When the encryption process proceeds from one image row to the next, the SPECK round-key set is switched to the corresponding key set of the new row, while the forward feedback state F_{q-1} is not reinitialized. Therefore, the intermediate state generated by the last block of the previous row continues to participate in the encryption of the first block of the next row, thereby establishing cross-row data dependencies. After completing the forward feedback encryption, an intermediate state sequence composed of F_0, F_1, ..., F_{N-1} is obtained.

Subsequently, the forward intermediate state sequence is used as the input of the second stage, where the backward feedback encryption is performed from the last block toward the beginning of the image. For the q-th forward intermediate state block, it is first XORed with the adjacent final ciphertext block C_{q+1} obtained during the backward scanning process, and then encrypted using the SPECK round-key set corresponding to the row of the current block, as follows:

$$C_q = E_h\left(F_q \oplus C_{q+1}\right), \; q = N-1, N-2, \ldots, 0. \tag{21}$$

Since $C_N = IV_b$, when $q = N-1$, Equation (18) naturally represents the encryption process of the last forward intermediate state block after fusion with the backward initial feedback value IV_b. The remaining ciphertext blocks are then sequentially calculated in the backward order until C_0 is obtained. After all blocks are processed, the final ciphertext sequence is composed of C_0, C_1, ..., C_{N-1}. By rearranging all ciphertext blocks according to the original image byte order, the ciphertext image with the same dimensions as the plaintext image is obtained.

In the above process, the forward stage feeds the previous intermediate state block back into the current plaintext block, whereas the backward stage feeds the adjacent final ciphertext block in the reverse direction back into the current intermediate state block. Therefore, the final ciphertext C_q is determined not only by the current plaintext block

and the corresponding row-wise dynamic round keys, but also by the joint influence of the forward and backward feedback chains. The SPECK64/128 algorithm performs the nonlinear transformation within each 64-bit data block, while the bidirectional feedback structure establishes information propagation paths among different data blocks.

The proposed bidirectional feedback encryption process is reversible. During decryption, the backward feedback encryption is first reversed according to the order from the end of the image toward the beginning. The forward intermediate state can be recovered as follows:

$$F_q = D_h\left(C_q\right) \oplus C_{q+1},\ q = N-1, N-2, \ldots, 0. \tag{22}$$

Here, for the last ciphertext block, $C_N = IV_b$ is assigned. After recovering all forward intermediate states, the forward feedback encryption is then reversed according to the order from the beginning to the end of the image. The original plaintext block can be recovered as follows:

$$P_q = D_h\left(F_q\right) \oplus F_{q-1},\ q = 0, 1, \ldots, N-1. \tag{23}$$

Here, for the first plaintext block, the initial feedback value is set as $F_{-1} = IV_f$. After completing the decryption of all blocks, the recovered byte sequence is rearranged according to the row-major order to obtain the original grayscale image.

4.4. Security Characterization and Lightweight Hardware Validation

To evaluate the encryption and decryption performance of the proposed scheme, 36 grayscale images are used for quantitative evaluation at 256 × 256 pixels, consistent with the image size adopted in the related studies. An additional 512 × 512 grayscale example is prepared by cropping and resampling an original KITTI image [30]. Representative results are shown in Figure 12. After encryption, the visual features of the images are effectively concealed, while the corresponding grayscale plaintexts are losslessly recovered using the correct keys.

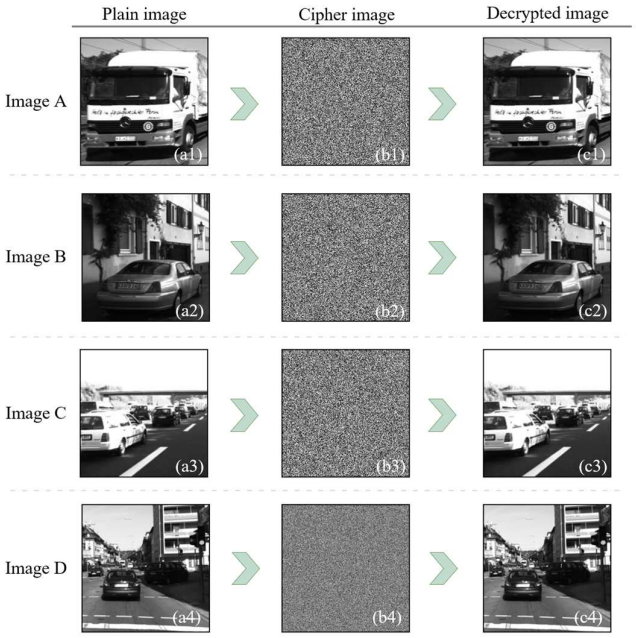


Figure 12. Image encryption and decryption results: (**a1**–**a4**) plaintext images, (**b1**–**b4**) ciphertext images, and (**c1**–**c4**) decrypted images. Images (**A**–**C**) are 256 × 256 pixels, and Image (**D**) is 512 × 512 pixels.

Furthermore, hardware encryption and decryption experiments using 256 × 256 grayscale images are conducted on the STM32H750VBT6 microcontroller, as shown in Figure 13.

The complete encryption and decryption processes are each executed 100 times, yielding average running times of 6.662 ms and 6.617 ms, respectively. The average encryption throughput is approximately 9.84 MB/s, including row-wise dynamic key generation, SPECK round-key expansion, and bidirectional feedback diffusion. The firmware build reports a combined read-write (RW) and zero-initialized (ZI) data allocation of 226,856 bytes and a read-only memory (ROM) load size of 96,752 bytes. These figures cover the complete firmware, including image buffers and supporting code, with the ROM load size also containing a 65,536-byte embedded test image. The additional computation mainly arises from row-wise key expansion and two SPECK encryptions per data block for bidirectional diffusion. Together, these results characterize the processing efficiency and storage requirements of the proposed implementation and demonstrate its feasibility on the STM32 platform.

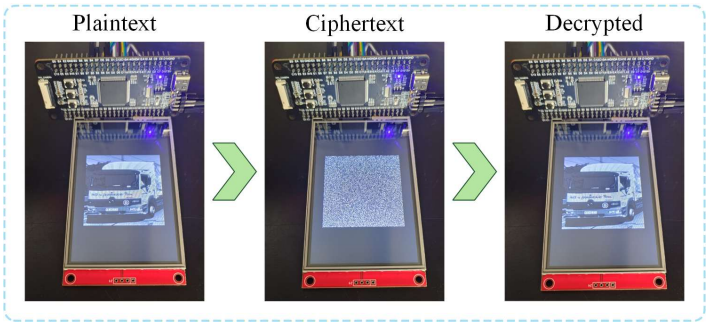


Figure 13. Hardware demonstration of the proposed image encryption and decryption scheme.

Subsequently, the statistical characteristics of the plaintext and ciphertext images are analyzed, with representative results shown in Figure 14. Compared with the plaintext images, the ciphertext images exhibit more uniform gray-level distributions and substantially reduced correlations among adjacent pixels. Across the 36 grayscale images of 256 × 256 pixels, the mean ciphertext information entropy is 7.9971 bits, further supporting the observed uniformity of the ciphertext gray-level distributions.

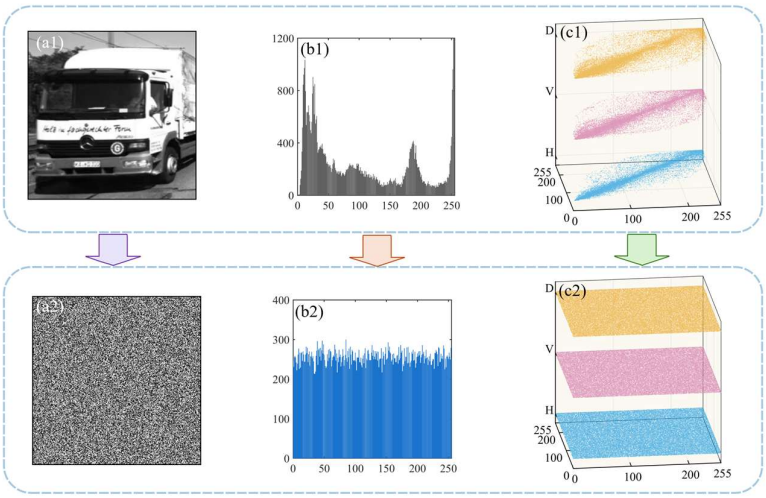


Figure 14. Statistical analysis of the plain and cipher images. (**a1**) plain image; (**a2**) cipher image; (**b1**) grayscale histogram of the plain image; (**b2**) grayscale histogram of the cipher image; (**c1**) adjacent-pixel correlation scatter plots of the plain image in the horizontal (H), vertical (V), and diagonal (D) directions; (**c2**) corresponding scatter plots of the cipher image.

The statistical properties of the chaotic sequences and resulting master-key bytes are evaluated using the autocorrelation function (ACF), approximate entropy (ApEn), and nor-

malized permutation entropy (NPE). Max. |ACF| is the maximum absolute autocorrelation over lags 1–64. ApEn uses an embedding dimension of 2 and a tolerance of 0.2 times the sequence standard deviation; NPE uses an embedding dimension of 4, a delay of 1, and normalization by $\ln(4!)$. As summarized in Table 4, the x and z sequences have lower maximum absolute autocorrelation than the master-key bytes, while all three sequences have NPE values close to unity. The residual autocorrelation in the master-key bytes indicates that high permutation entropy alone does not establish statistical independence, suggesting scope for refining the sampling and key-construction procedures.

Table 4. Statistical properties of key-generation sequences.

Sequence	Sample Size	Max. |ACF|	ApEn	NPE
x sequence	1024	0.0804	1.6063	0.9957
z sequence	1024	0.1106	1.7296	0.9886
Master-key bytes	4096	0.4345	1.6856	0.9931

Key space comprises the possible keys used by an encryption scheme. Each row-wise dynamic master key is constructed from four pairs of chaotic state samples, quantized into eight 16-bit unsigned integers and concatenated into a 128-bit key for the SPECK64/128 key schedule. Its nominal key space comprises 2^{128} possible bit patterns, within which the generated keys are determined by the initial-state and parameter configurations through finite-precision sampling and quantization.

Small initial-state differences may be reduced or eliminated during sampling and quantization, while a row-key change may affect only part of the decrypted image. Sensitivity tests therefore examine ciphertext changes and the extent of decryption errors under perturbed configurations using 36 grayscale images of 256 × 256 pixels. Each initial-state variable is perturbed separately by +1 and +16 units in the last place (ULP), yielding six perturbations per image. ULP denotes the spacing between adjacent representable floating-point values near the tested value. Row-key tests select rows 1, 128, and 256, covering the beginning, middle, and end of the image. Sensitivity tests flip the least significant bit of the first byte of each selected row key; avalanche tests flip each of its 128 bits individually.

Ciphertext bit-change rate (CBCR) measures the proportion of differing bits between ciphertexts generated under original and perturbed configurations. Decryption pixel-error rate (DPER) measures the proportion of pixels differing from the original plaintext when its ciphertext is decrypted under the perturbed configuration. Sensitivity tests evaluate both metrics, whereas avalanche tests evaluate CBCR only. Initial-condition perturbations substantially alter both ciphertext and decrypted images. Single-row key perturbations produce CBCR values close to 50%, with localized decryption errors. The 13,824 avalanche tests yield a mean CBCR of 50.0002%, indicating changes in approximately half of the ciphertext bits. Table 5 summarizes the results.

Table 5. Key sensitivity and avalanche results

Test	Test Count	CBCR (%)	DPER (%)
Initial-condition sensitivity	216	49.8220–50.1755	99.5377–99.6689
Single-row key sensitivity	108	49.8798–50.1905	0.3845–0.4028
Key avalanche	13,824	Mean: 50.0002 Range: 49.7263–50.2499	—

Noise and occlusion experiments assess decryption recovery after ciphertext damage using the same 36 images, with the original keys and initialization vectors retained.

The tested conditions are random bit flips with a per-bit probability of 0.01%, salt-and-pepper noise at 0.1% density, and central square occlusion covering 6.25% of the image area. Each random-noise condition is repeated 20 times per image; occlusion sets a central 64 × 64 ciphertext region to zero without changing its length. Recovery is assessed through DPER and peak signal-to-noise ratio (PSNR). The respective mean PSNR values are 24.4566 dB, 23.5094 dB, and 18.4180 dB, characterizing recovery under the tested sparse-noise and localized-occlusion conditions. Table 6 reports the mean DPER and PSNR, with PSNR averaged arithmetically over individual trials.

Table 6. Decryption recovery under ciphertext damage.

Ciphertext Damage	Damage Level	Mean DPER (%)	Mean PSNR (dB)
Random bit flips	Bit-flip probability: 0.01%	1.9056	24.4566
Salt-and-pepper noise	Noise density: 0.1%	2.3611	23.5094
Central occlusion	Area: 6.25%	7.7810	18.4180

To evaluate the effectiveness of the bidirectional feedback structure, an ablation study is conducted. Under the same test images, chaotic parameters, and dynamic key generation strategy, the performance of the original SPECK, SPECK with unidirectional feedback, and the proposed bidirectional feedback scheme are compared. Table 7 presents the statistical evaluation results averaged over multiple test images for the three schemes. The results demonstrate that, compared with the unidirectional feedback structure, the bidirectional feedback mechanism further establishes forward and backward data dependencies, thereby improving the differential diffusion capability of ciphertext images. Here, SPECK-FD denotes the encryption scheme incorporating the forward diffusion mechanism.

Table 7. Ablation study of the proposed encryption scheme.

Scheme	NPCR (%)			UACI (%)		
	Pixel 0	Pixel 32,768	Pixel 65,535	Pixel 0	Pixel 32,768	Pixel 65,535
SPECK	0.0122	0.0122	0.0117	0.0040	0.0046	0.0042
SPECK-FD	99.6043	49.8103	0.0122	33.5111	16.7897	0.0037
This work	99.6012	99.6114	99.6068	33.4653	33.4681	33.4657

Plaintext differential tests examine how a small plaintext change propagates through the encryption process. For each of the 36 grayscale images of 256 × 256 pixels, three fixed positions—the first, middle, and last pixels—and 100 additional randomly selected, non-repeating positions are tested. Each test flips the least significant bit of one pixel while retaining the same keys and initialization vectors for the original and modified plaintexts. Across the resulting 3708 test cases, the mean number of pixels change rate (NPCR) and unified average changing intensity (UACI) are 99.6097% and 33.4697%, respectively. These results indicate that single-pixel perturbations produce widespread ciphertext changes under the tested conditions.

Furthermore, to evaluate the security performance differences between the proposed scheme and existing image encryption methods, Table 8 presents a comparison of the proposed scheme with related works in terms of information entropy, NPCR, UACI, and other security metrics. It can be observed that the proposed scheme exhibits stable performance in terms of differential characteristics, randomness, and pixel decorrelation, demonstrating its favorable overall statistical security performance. Beyond the representative Image A reported in Table 8, the evaluation covers 36 grayscale images of 256 × 256 pixels. The mean NPCR, UACI, and ciphertext entropy are 99.6097%, 33.4697%, and 7.9971 bits, respectively.

The signed mean correlation coefficients in the horizontal, vertical, and diagonal directions are −0.0003, −0.0003, and −0.0006, respectively, with all individual absolute coefficients below 0.0081.

Table 8. Statistical security comparison with related methods.

Refs.	Image	NPCR (%)	UACI (%)	Entropy	CC (H)	CC (V)	CC (D)
[31] (2025)	Boat	99.6078	33.4201	7.9971	−0.0001	−0.0023	0.0007
[32] (2025)	Image 1	99.6057	33.4672	7.9962	0.0012	−0.0023	−0.0019
[33] (2024)	Shoeprint	99.6108	33.3678	7.9977	—	—	—
[34] (2025)	256 × 256	99.6103	33.5234	7.9970	—	—	—
This work	Image A	99.6094	33.4642	7.9975	−0.0021	−0.0011	0.0001

5. Conclusions

This paper developed a three-dimensional discrete memristive Rulkov neuron and proposed a TCA criterion for rule-based attractor editing. By linking memristive state regions, local gains, and attractor properties, TCA enables the controllable construction of coexisting attractors and configurable editing of their dynamical characteristics and spatial distributions. Quantitative comparisons of regional mean positions and maximum Lyapunov exponents further characterize the respective effects of gain reassignment and spacing adjustment. Unlike conventional global parameter tuning, the proposed mechanism performs localized editing within prescribed state regions, allowing selected attractor properties to be configured without restructuring the overall system. This provides a flexible way to organize multiple neuronal dynamical states within a unified model. Numerical analysis and STM32 experiments verify the proposed dynamics and editing mechanism. Furthermore, the configurable chaotic sequences generated by edited attractors are used for dynamic key generation and combined with SPECK-BiDiff to achieve lightweight IoV image encryption. Tests on 36 grayscale images further characterize plaintext and key sensitivity, ciphertext statistics, and decryption recovery under noise and occlusion. Overall, the proposed framework extends memristive neuronal multistability from attractor generation to configurable attractor editing and provides a practical link between controllable neuronal dynamics and lightweight information security. Future work will extend the TCA framework to coupled or higher-dimensional neuronal systems, explore adaptive attractor editing under time-varying conditions, and investigate its encryption application to color images.

Author Contributions: S.G.: investigation, software, circuit simulation, writing—original draft. H.W.: conceptualization; methodology, writing—review & editing, project administration, funding acquisition. W.X.: validation, circuit implementation. J.Q.: validation, circuit implementation. Y.Z.: formal analysis, data curation. Y.L.: data analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in the KITTI dataset at https://s3.eu-central-1.amazonaws.com/avg-kitti/data_object_image_2.zip (accessed on 29 July 2026).

Acknowledgments: The authors gratefully acknowledge the technical support provided by Zaozhuang University.

Conflicts of Interest: We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ganguly, C.; Bezugam, S.S.; Abs, E.; Payvand, M.; Dey, S.; Suri, M. Spike frequency adaptation: Bridging neural models and neuromorphic applications. *Commun. Eng.* **2024**, *3*, 22. [CrossRef]
2. Vignesh, D.; He, S.; Banerjee, S. A review on the complexities of brain activity: Insights from nonlinear dynamics in neuroscience. *Nonlinear Dyn.* **2025**, *113*, 4531–4552. [CrossRef]
3. Kipelkin, I.M.; Gerasimova, S.A.; Guseinov, D.V.; Belov, A.I.; Tyukin, I.Y.; Mikhaylov, A.N.; Kazantsev, V.B. Periodic and quasi-periodic dynamics of neuronal oscillators based on an inorganic memristive device. *Chaos Solitons Fractals* **2026**, *208*, 118304. [CrossRef]
4. Le, B.B. Asymmetric coupling of nonchaotic Rulkov neurons: Fractal attractors, quasimultistability, and final state sensitivity. *Phys. Rev. E* **2025**, *111*, 034201. [CrossRef]
5. Mou, J.; Cao, H.; Zhou, N.; Cao, Y. An FHN-HR neuron network coupled with a novel locally active memristor and its DSP implementation. *IEEE Trans. Cybern.* **2024**, *54*, 7333–7342. [CrossRef]
6. Nazari, S.; Jamshidi, S. Efficient digital design of the nonlinear behavior of Hindmarsh–Rose neuron model in large-scale neural population. *Sci. Rep.* **2024**, *14*, 3833. [CrossRef]
7. Goufo, E.F.D. Dynamic analysis of a multistable memristor based coupled Wilson neuron circuits embedded in an Arduino Due microcontroller. *Chaos Solitons Fractals* **2026**, *206*, 117862. [CrossRef]
8. Li, Y.; Li, C.; Li, Y.; Moroz, I.; Yang, Y. A joint image encryption based on a memristive Rulkov neuron with controllable multistability and compressive sensing. *Chaos Solitons Fractals* **2024**, *182*, 114800. [CrossRef]
9. Ghosh, I.; Nair, A.S.; Fatoyinbo, H.O.; Muni, S.S. Dynamical properties of a small heterogeneous chain network of neurons in discrete time. *Eur. Phys. J. Plus* **2024**, *139*, 545. [CrossRef]
10. Lv, W.; Zhang, S.; Wang, Y.; Wu, Q.; Ding, X.; Gao, X. Memristive Multiring Rulkov Neuron Model With Hyperchaotic Firings and Application to Image Cryptosystem. *IEEE Trans. Ind. Inform.* **2026**, *22*, 8492–8503. [CrossRef]
11. Li, M.-R.; Liao, K.-C.; Yao, Y.-C.; Li, Y.-X.; Liu, Q.-H.; Xu, H.-W. Hierarchical regulation of scroll capacity and attractor connectivity in memristive chaotic systems. *Chaos Solitons Fractals* **2026**, *213*, 119327. [CrossRef]
12. Muthukumar, D.; Hajian, D.N.; Natiq, H.; Mehrabbeik, M.; Pal, N.; Jafari, S. Multilayer structure-induced collective dynamics in uncoupled memristive Rulkov neurons: Impact of field coupling and intralayer connections. *Physica D Nonlinear Phenom.* **2025**, *471*, 134464. [CrossRef]
13. Hajian, D.N.; Ramadoss, J.; Natiq, H.; Parastesh, F.; Rajagopal, K.; Jafari, S. Dynamics of Hindmarsh–Rose neurons connected via adaptive memristive synapse. *Chin. J. Phys.* **2024**, *87*, 311–329. [CrossRef]
14. Bao, H.; Wang, R.; Wang, N.; Chen, M.; Bao, B. Discrete Memristive Hopfield Neural Network with Grid-Polyhedral Hyperchaos for FPGA-Based Pseudorandom Number Generator. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **2025**, *196*, 108340. [CrossRef]
15. Zhang, Z.; Li, Y.; Yao, Y.; Xu, W.; Lei, T.; Liu, Q. Offset-boosted reconfigurable multiscroll attractors in memristive Chua system for robot path planning. *Chaos Solitons Fractals* **2026**, *213*, 119228. [CrossRef]
16. Huang, L.; Chen, F.; Yang, Y.; Zheng, L. A novel memristor Hopfield neural network with homogeneous coexisting multi-scroll attractors. *Physica Scr.* **2024**, *99*, 065217. [CrossRef]
17. Fu, B.; Li, Y.; Li, Z.; Yao, Y.; Lei, T.; Xu, H. Controllable attractor dynamics in a memristive Hide dynamo. *Chaos Solitons Fractals* **2026**, *213*, 119328. [CrossRef]
18. Li, C.; Sprott, J.C.; Liu, W. Finding More Chaotic Flows with Complete Control. *Int. J. Bifurc. Chaos* **2026**, *36*, 2650164. [CrossRef]
19. Li, C.; Yu, W.; Moroz, I.; Li, Y.; Zheng, Y. Hybrid offset boosting in a hyperchaotic map. *Chaos Solitons Fractals* **2026**, *204*, 117735. [CrossRef]
20. Li, Y.; Zhang, S.; Wang, Y.; Lei, T.; Liu, Q.; Yao, Y.; Chen, B. Offset boosting-dominated on-demand multistability editing in memristive neural network. *Chaos Solitons Fractals* **2026**, *208*, 118267. [CrossRef]
21. Wang, C.; Li, Y.; Deng, Q. Discrete-time fractional-order local active memristor-based Hopfield neural network and its FPGA implementation. *Chaos Solitons Fractals* **2025**, *193*, 116053. [CrossRef]
22. Zheng, L.; Ren, D. A Light-weight Deep Neural Network for Vehicle Detection in Complex Tunnel Environments. *Instrumentation* **2023**, *10*, 32–44.
23. Huang, X.; Li, L.; Zhang, H.; Yang, J.; Kuang, J. IoVCipher: A low-latency lightweight block cipher for internet of vehicles. *Ad. Hoc Netw.* **2024**, *160*, 103524. [CrossRef]
24. Liang, X.; Xing, H.; Gu, W.; Hou, T.; Wang, X. Hybrid gaussian network intrusion detection method based on cgan and e-graphsage. *Instrumentation* **2024**, *11*, 24–35.
25. Zhang, X.; Zhang, S.; Lu, S.; Wang, X.; Zeng, Z. Memristive Coupled Dual Rulkov Neuron Model with Multi-type Hyperchaotic Firings and Application to Image Encryption for IoMT. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *early access*. [CrossRef]
26. Kuang, W.; Luo, W. Based on STM32 of CNN Speech Keyword Command Recognition System. *Instrumentation* **2023**, *10*, 17–22.
27. Li, Y.; Li, C.; Lei, T.; Yang, Y.; Chen, G. Offset boosting-entangled complex dynamics in the memristive Rulkov neuron. *IEEE Trans. Ind. Electron.* **2023**, *71*, 9569–9579. [CrossRef]

28. Zhang, J.; Zuo, J.; Guo, Y.; Hou, J.; Xie, Q. Nonlinear analysis, circuit implementation, and application in image encryption of a four-dimensional multi-scroll hyper-chaotic system. *Integration* **2024**, *95*, 102126. [CrossRef]
29. Song, M.; Ke, Y.; Du, X. From Digital Transformation to Circular Manufacturing: How Digital Organizational Capabilities Foster Green Innovation Under Regulatory Intervention. In *Corporate Social Responsibility and Environmental Management*; Wiley-Blackwell: Chichester, UK, 2026.
30. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
31. Wang, Y.; Zhang, S.; Wang, Z.; Wang, J.; Peng, X.; Hua, G. Robust enhanced chaos in memristive multiattractor rulkov neuron model with application to hardware image cryptosystem. *IEEE Internet Things J.* **2025**, *12*, 32022–32034. [CrossRef]
32. Wang, H.; Lai, Q. Exploiting compressed sensing and DNA coding for multi-image encryption scheme with application to industrial internet of things. *IEEE Trans. Ind. Electron.* **2025**, *72*, 14543–14552. [CrossRef]
33. Lin, H.; Deng, X.; Yu, F.; Sun, Y. Grid multibutterfly memristive neural network with three memristive systems: Modeling, dynamic analysis, and application in police IoT. *IEEE Internet Things J.* **2024**, *11*, 29878–29889. [CrossRef]
34. Shi, F.; Xu, X.; Chen, J.; Banerjee, S.; Mou, J. A novel memristive autapse neuron with ReLU function: Chaos dynamics, attractor control and application in image encryption. *Nonlinear Dyn.* **2025**, *113*, 26753–26772. [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.