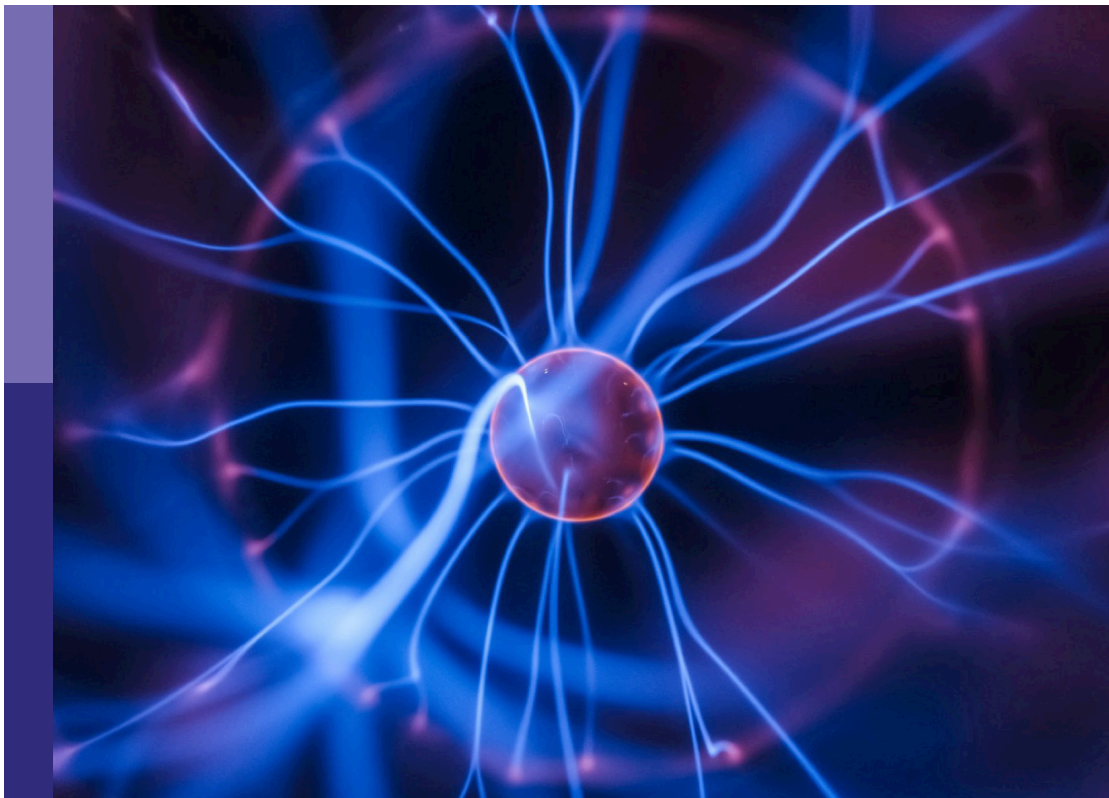
frontiers | Research Topics

Advances in nonlinear systems and networks, volume II

Edited by
Fei Yu, Ho Ching Iu, Hairong Lin
and Viet-Thanh Pham

Published in
Frontiers in Physics



ISSN 1664-8714
ISBN 978-2-8325-4559-1
DOI 10.3389/978-2-8325-4559-1

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Advances in nonlinear systems and networks, volume II

Topic editors

Fei Yu — Changsha University of Science and Technology, China
Ho Ching Iu — University of Western Australia, Australia
Hairong Lin — Hunan University, China
Viet-Thanh Pham — Ton Duc Thang University, Vietnam

Citation

Yu, F., Iu, H. C., Lin, H., Pham, V.-T., eds. (2024). *Advances in nonlinear systems and networks, volume II*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-4559-1

Table of contents

frontiers | Frontiers in Physics

TYPE Editorial
PUBLISHED 26 March 2024
DOI 10.3389/fphy.2024.1396178

OPEN ACCESS

EDITED AND REVIEWED BY
Alex Hansen,
NTNU, Norway

*CORRESPONDENCE
Fei Yu,
yufeiyfyf@csust.edu.cn
Ho-Ching Iu,
herbert.iu@uwa.edu.au
Hairong Lin,
haironglin@hnu.edu.cn
Viet-Thanh Pham,
phamvietthanh@tdtu.edu.vn

RECEIVED 05 March 2024
ACCEPTED 18 March 2024
PUBLISHED 26 March 2024

CITATION
Yu F, Iu H-C, Lin H and Pham V-T (2024), Editorial: Advances in nonlinear systems and networks, volume II.
Front. Phys. 12:1396178.
doi: 10.3389/fphy.2024.1396178

Editorial: Advances in nonlinear systems and networks, volume II

Fei Yu[1]*, Ho-Ching Iu[2]*, Hairong Lin[3]* and Viet-Thanh Pham[4]*

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China, [2]School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth, WA, Australia, [3]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, [4]Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

KEYWORDS

editorial, nonlinear systems, nonlinear networks, nonlinear device, application

Editorial on the Research Topic
Advances in nonlinear systems and networks, volume II

1 Introduction

Nonlinear systems and networks refer to physical, chemical, biological, or engineering systems and networks that have nonlinear relationships within them. Compared to linear systems and networks, the characteristic of nonlinear systems and networks is the nonlinear relationship between input and output. In nonlinear systems and networks, the relationship between input and output does not follow the principle of linear superposition, so simple linear equations cannot be used to describe the behavior of systems and networks.

Nonlinear systems and networks are widely present in various fields, such as mechanics, circuits, chemical reactions, encrypted communication and biology [1–5]. The study of nonlinear systems is of great significance for us to deeply understand the essence of natural phenomena and improve engineering design. Due to the complexity, unpredictability, and adaptability of nonlinear systems and networks, their applications and research face significant challenges. But with the continuous development of science and technology, the research and application of nonlinear systems and networks are also deepening in various fields, such as chaotic systems [6–10], chaotic circuits [11–14], nonlinear devices [15–17], neural networks [18–24], neural circuits [25–28], memristors [29–31], system synchronization and control [32–36], system optimization [37–39], and related application fields [40–43].

Due to the success of the first Research Topic of "Advancements in Nonlinear Systems and Networks" [44], we have decided to continue to focus on the continuous progress of Nonlinear Systems and Networks in the second volume. In this Research Topic, 10 articles about nonlinear systems and networks and their applications are reported. For nonlinear networks, dynamic displacement estimation of structures using onedimensional convolutional neural network is studied (Zhou and He), and two-layer complex propagation network with individual heterogenous decreased behavior is analyzed (Tian et al.). A robust and fast representation learning model RFRL for heterogeneous networks is studied (Lei et al.). A reputation-based user electricity scheduling scheme for the complex network of user electricity consumption is proposed (Tang et al.). For nonlinear devices,

Ternary combinational logic gates (Li et al.) and current mode multi scroll chaotic oscillator (Lin et al.) are studied. For the synchronization control of nonlinear systems and networks, effective vibration control in Nano-Beams (Alsubaie), brain-like coupling synchronization (Fu et al.) and pinning synchronization (Yu et al.) are studied. In addition, considering the application of nonlinear systems and networks, a novel grid multi-structure chaotic attractor and its application in medical image encryption is researched (Hu et al.).

2 Summary of papers presented in this Research Topic

Zhou and He, in the paper "Dynamic displacement estimation of structures using onedimensional convolutional neural network," proposed a structural dynamic displacement estimation method based on one-dimensional convolutional neural networks and acceleration data. To verify the reliability of the method, this paper established a finite element-based framework structure. Collect the acceleration and displacement of each node in the framework model under earthquake response. In addition, a typical neural network was used for comparative research. The results show that the error of the neural network model in dynamic displacement estimation task is 9. 52 times that of the one-dimensional convolutional neural network model. Meanwhile, the proposed modeling scheme has strong noise resistance. To verify the practicality of the proposed method, the authors collected data from actual framework structures. The experimental results show that the mean square error of this method in actual dynamic displacement estimation tasks is only 5. 097, which meets the engineering needs.

Tian et al., in the paper "Dynamics analysis on two-layer complex propagation network with individual heterogenous decreased behavior," constructed a double-layer network model to describe individual behavioral contact and proposes a threshold function to represent individual heterogeneous decreased behavior (IHDB). Meanwhile, the authors use partition theory to explain the mechanism of information dissemination. Through experiments, it has been proven that there is a sustained information explosion in the final adoption scale when an individual exhibits positive IHDB. However, when individuals exhibit passive IHDB, the final adoption of scale will result in discontinuous information bursts. Finally, the experiment shows that the theoretical analysis is consistent with the simulation results.

Lei et al., in the paper "Robust and fast representation learning for heterogeneous information networks," studied a robust and fast representation learning model RFRL for heterogeneous networks. Firstly, the global features of heterogeneous networks are divided into multiple intra type local features and inter type local features, and a type aware biased sampling is designed to generate training samples for each local feature. Secondly, shallow representation strategies using node type perception and link type perception are used to learn intra type and inter type features, respectively. Finally, adversarial learning is used to integrate the above two representation strategies to address invisible network noise and enhance the robustness of representation learning. A large number of experiments on three network analysis tasks and three public datasets have demonstrated the good performance of the RFRL model proposed in this paper.

Tang et al., in the paper "Reputation-based electricity scheduling scheme for complex network of user electricity consumption," proposed a reputation-based user electricity scheduling scheme for the complex network of user electricity consumption. In the scheme of the paper, the authors first model the complex network of user electricity consumption. Then, a method for calculating the reputation of power users was constructed. In addition, the paper uses machine learning methods to train computational models to calculate the adjustment coefficients of power loads, and then adjusts power scheduling tasks based on the calculated adjustment coefficients. Finally, the corresponding power dispatch tasks are assigned to the selected power users for adjusting their electricity consumption. The experimental results demonstrate the effectiveness of the scheme.

Li et al., in the paper "Ternary combinational logic gates design based on tri-valued memristors," proposed a design method for ternary circuits without cascading basic ternary logic gates on the basis of ternary memristors. The proposed method can directly achieve specific logic functions through series memristors. At the same time, this method was used to implement a ternary encoder, ternary decoder, ternary comparator, and ternary data selector. Finally, the authors verified the effectiveness of the circuit through LTspice simulation.

Lin et al., in the paper "Current mode multi scroll chaotic oscillator based on CDTA," proposed a current mode chaotic oscillation circuit based on a current differential transconductance amplifier (CDTA). This circuit fully utilizes the advantages of current differential transconductance amplifiers. The linear and nonlinear parts of the proposed circuit operate in current mode, achieving a true current mode multi scroll chaotic circuit. Finally, the authors conducted simulations using Pspice, and the results showed that the proposed current type chaotic circuit can generate multi scroll chaotic attractors.

Alsubaie, in the paper "a neural state-space-based model predictive technique for effective vibration control in nano-beams," proposed a system recognition method based on deep neural networks and combines it with MPC. In addition, the paper ensures the robustness and convergence of the closed-loop system by adding control terms. Then, the control equation for non local strain gradient (NSG) nanobeams was given. Finally, the proposed control scheme will be applied to the vibration suppression of NSG nanobeams. To verify the effectiveness of the proposed method, the controller is applied to an unknown system. The simulation results ultimately proved the significant performance of the method proposed by the authors in effectively suppressing vibration.

Fu et al., in the paper "Multi-scroll Hopfield neural network under electromagnetic radiation and its brain-like coupling synchronization," proposed a new non-volatile magnetic controlled memristor and uses it to simulate the effects of membrane flux changes caused by neuronal exposure to electromagnetic radiation. Through dynamic analysis, a series of complex chaotic phenomena were discovered, including multi vortex chaotic attractors controlled by memristors, symmetric bifurcation behavior, and coexisting phenomena with enhanced initial offset. Secondly, the authors also proposed a dual memristive HNN coupled synchronization model to simulate synchronization schemes between different regions of the human

brain. The feasibility of the synchronization scheme was verified by establishing a Simulink model and conducting simulation experiments.

Yu et al., in the paper "Moment-based analysis of pinning synchronization in complex networks with sign inner-coupling configurations," investigated the pinning synchronization problem of complex networks with symbolic intra coupling configuration using a moment-based analysis method. Firstly, two representative nonlinear systems with dynamic parameter changes are presented. Then, a detailed study was conducted on the impact of symbol internal coupling configuration on network synchronization. Research has found that adding negative parameters to the internal coupling matrix can significantly improve the synchronization of the network. Finally, the authors provided explanations through numerical simulations.

Hu et al., in the paper "A novel grid multi-structure chaotic attractor and its application in medical image encryption," proposed a memristive Hopfield neural network model using the memristor synaptic control method. This model can generate new grid multi structure chaotic attractors. Firstly, the generation mechanism of grid multi structure chaotic attractors were analyzed from the perspectives of equilibrium points and stability. Secondly, its basic dynamic characteristics were analyzed. Thirdly, the simulation circuit of the neural network model was designed and implemented using Multisim. Finally, combining the principle of chaotic encryption, the authors designed an image encryption scheme based on a generated grid multi structure attractor. The experimental results show that compared with existing schemes, this scheme has greater information entropy, higher key sensitivity, and good application prospects.

3 Concluding remarks

Overall, the research on the application and development of nonlinear systems and networks requires continuous advancement from multiple aspects, in order to better respond to challenges and explore their broad application prospects. The exploration and research of nonlinear systems and networks will undoubtedly bring us more new modeling, control, prediction, and optimization methods in the future.

Finally, we would like to thank all the authors of the 10 articles in this Research Topic for their outstanding contributions, all of which are well suited to the scope of this Research Topic. In addition, we would also like to sincerely thank all the reviewers, editors, and editorial staff of Frontiers in Physics journal for their support.

Author contributions

FY: Conceptualization, Formal Analysis, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing–original draft, Writing–review and editing. H-CI: Conceptualization, Methodology, Project administration, Supervision, Validation, Visualization, Writing–review and editing. HL: Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing–review and editing. V-TP: Formal Analysis, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Natural Science Foundation of Hunan Province under Grants 2022JJ30624, 2022JJ10052 and 2021JJ30741; the Scientific Research Fund of Hunan Provincial Education Department under grant 21B0345; the National Natural Science Foundation of China under Grant 62172058; and the Postgraduate Training Innovation Base Construction Project of Hunan Province under Grant 2020-172-48.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

References

1. Lew AJ, Buehler MJ. Single-shot forward and inverse hierarchical architected materials design for nonlinear mechanical properties using an attention-diffusion model. *Mater Today* (2023) 64:10–20. doi:10.1016/j.mattod.2023.03.007

2. Xu Q, Wang Y, Chen B, Li Z, Wang N. Firing pattern in a memristive Hodgkin–Huxley circuit: numerical simulation and analog circuit validation. *Chaos, Solitons & Fractals* (2023) 172:113627. doi:10.1016/j.chaos.2023.113627

3. Yu F, Kong X, Chen H, Yu Q, Cai S, Huang Y, et al. A 6D fractional-order memristive Hopfield neural network and its application in image encryption. *Front Phys* (2022) 10:847385. doi:10.3389/fphy.2022.847385

4. Sun J, Wang Y, Liu P, Wen S, Wang Y. Memristor-based circuit design of PAD emotional space and its application in mood congruity. *IEEE Internet Things J* (2023) 10(18):16332–42. doi:10.1109/jiot.2023.3267778

5. Deng Q, Wang C, Sun J, Sun Y, Jiang J, Lin H, et al. Nonvolatile CMOS memristor, reconfigurable array, and its application in power load forecasting. *IEEE Trans Ind Inform* (2023) 1–12. doi:10.1109/TII.2023.3341256

6. Yu F, Yuan Y, Wu C, Yao W, Xu C, Cai S, et al. Modeling and hardware implementation of a class of Hamiltonian conservative chaotic systems with transient quasi-period and multistability. *Nonlinear Dyn* (2024) 112:2331–47. doi:10.1007/s11071-023-09148-5

7. Lin H, Wang C, Sun Y. A universal variable extension method for designing multiscroll/wing chaotic systems. *IEEE Trans Ind Elect* (2023). doi:10.1109/TIE.2023.3299020

8. Kong X, Yu F, Yao W, Xu C, Zhang J, Cai S, et al. A class of 2n+1 dimensional simplest Hamiltonian conservative chaotic systems and fast image encryption schemes. *Appl Math Model* (2024) 125:351–74. doi:10.1016/j.apm.2023.10.004

9. Yu F, Xu S, Xiao X, Yao W, Huang Y, Cai S, et al. Dynamics analysis, FPGA realization and image encryption application of a 5D memristive exponential hyperchaotic system. *Integration* (2023) 90:58–70. doi:10.1016/j.vlsi.2023.01.006

10. Yu F, Zhang W, Xiao X, Yao W, Cai S, Zhang J, et al. Dynamic analysis and FPGA implementation of a new, simple 5D memristive hyperchaotic sprott-C system. *Mathematics* (2023) 11(3):701. Article ID 701. doi:10.3390/math11030701

11. Karimov A, Rybin V, Kopets E, Karimov T, Nepomuceno E, Butusov D. Identifying empirical equations of chaotic circuit from data. *Nonlinear Dyn* (2023) 111(1):871–86. doi:10.1007/s11071-022-07854-0

12. Chen Y, Mou J, Jahanshahi H, Wang Z, Cao Y. A new mix chaotic circuit based on memristor–memcapacitor. *The Eur Phys J Plus* (2023) 138(1):78. doi:10.1140/epjp/s13360-023-03699-7

13. Ahmadi A, Parthasarathy S, Pal N, Rajagopal K, Jafari S, Tlelo-Cuautle E. Extreme multistability and extreme events in a novel chaotic circuit with hidden attractors. *Int J Bifurcation Chaos* (2023) 33(07):2330016. doi:10.1142/s0218127423300161

14. Yu F, Kong X, Mokbel AAM, Yao W, Cai S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive hopfield neural network with a novel local active memeristor. *IEEE Trans Circuits Systems–II: Express Briefs* (2023) 70(1):326–30. doi:10.1109/tcsii.2022.3218468

15. Karankova S, Kovalchuk O, Lee S, Ryu B, Uddin S, Moon H, et al. Optical saturable absorption of conformal graphene directly synthesized on nonlinear device surfaces. *Appl Surf Sci* (2023) 611:155641. doi:10.1016/j.apsusc.2022.155641

16. Wang M, Gu L. Multiple mixed state variable incremental integration for reconstructing extreme multistability in a novel memristive hyperchaotic jerk system with multiple cubic nonlinearity. *Chin Phys B* (2024) 33(2):020504. doi:10.1088/1674-1056/acddd0

17. Schreurs DMMP, Verspecht J, Vandenberghe S, Vandamme E. Straightforward and accurate nonlinear device model parameter-estimation method based on vectorial large-signal measurements. *IEEE Trans microwave Theor Tech* (2002) 50(10):2315–9. doi:10.1109/tmtt.2002.803427

18. Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi-scroll memristive hopfield neural network. *IEEE Trans Netw Sci Eng* (2023) 10(2):845–58. doi:10.1109/tnse.2022.3223930

19. Deng Q, Wang C, Lin H. Memristive Hopfield neural network dynamics with heterogeneous activation functions and its application. *Chaos, Solitons and Fractals* (2024) 178:114387. doi:10.1016/j.chaos.2023.114387

20. Kong X, Yu F, Yao W, Cai S, Zhang J, Lin H. Memristor-induced hyperchaos, multiscroll and extreme multistability in fractional-order HNN: image encryption and FPGA implementation. *Neural Networks* (2024) 171:85–103. doi:10.1016/j.neunet.2023.12.008

21. Yao W, Liu J, Sun Y, Zhang J, Yu F, Cui L, et al. Dynamics analysis and image encryption application of Hopfield neural network with a novel multistable and highly tunable memristor. *Nonlinear Dyn* (2024) 112(1):693–708. doi:10.1007/s11071-023-09041-1

22. Lu J, Xie X, Lu Y, Wu Y, Li C, Ma M. Dynamical behaviors in discrete memristor-coupled small-world neuronal networks. *Chin Phys B* (2023). doi:10.1088/1674-1056/ad1483

23. Yu F, Kong X, Yao W, Zhang J, Cai S, Lin H, et al. Dynamics analysis, synchronization and FPGA implementation of multiscroll Hopfield neural networks with non-polynomial memristor. *Chaos, Solitons & Fractals* (2024) 179:114440. Article ID 114440. doi:10.1016/j.chaos.2023.114440

24. Wang C, Dong T, Lin H, Yu F, Sun Y. High-dimensional memristive neural network and its application in commercial data encryption communication. *Expert Syst Appl* (2024) 242:122513. doi:10.1016/j.eswa.2023.122513

25. Sun J, Wang Y, Liu P, Wen S. Memristor-based neural network circuit with multimode generalization and differentiation on pavlov associative memory. *IEEE Trans Cybernetics* (2022) 53(5):3351–62. doi:10.1109/tcyb.2022.3200751

26. Deng Z, Wang C, Lin H, Sun Y. Memristive spiking neural network circuit with selective supervised attention algorithm. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2023) 42(8):2604–17. doi:10.1109/TCAD.2022.3228896

27. Tang D, Wang C, Lin H, Yu F. Dynamics analysis and hardware implementation of multi-scroll hyperchaotic hidden attractors based on locally active memristive hopfield neural network. *Nonlinear Dyn* (2024) 112:1511–27. doi:10.1007/s11071-023-09128-9

28. Xu Q, Wang Y, Wu H, Chen M, Chen B. Periodic and chaotic spiking behaviors in a simplified memristive Hodgkin-Huxley circuit. *Chaos, Solitons and Fractals* (2024) 179:114458. doi:10.1016/j.chaos.2024.114458

29. Xu Q, Wang Y, Ho-Ching Iu H, Wang N, Han B. Locally active memristor based neuromorphic circuit: firing pattern and hardware experiment. *IEEE Trans Circuits Syst Regular Pap* (2023) 70(8):3130–41. doi:10.1109/tcsi.2023.3276983

30. Zhang W, Yao P, Gao B, Liu Q, Wu D, Zhang Q, et al. Edge learning using a fully integrated neuro-inspired memristor chip. *Science* (2023) 381(6663):1205–11. doi:10.1126/science.ade3483

31. Ren L, Mou J, Banerjee S, Zhang Y. A hyperchaotic map with a new discrete memristor model: design, dynamical analysis, implementation and application. *Chaos, Solitons & Fractals* (2023) 167:113024. doi:10.1016/j.chaos.2022.113024

32. Tan F, Zhou L, Lu J, Zhang H. Fixed-time synchronization in multilayer networks with delay Cohen–Grossberg neural subnets via adaptive quantitative control. *Asian J Control* (2024) 26:446–55. doi:10.1002/asjc.3217

33. Jin J, Chen W, Ouyang A, Yu F, Liu H. A time-varying fuzzy parameter zeroing neural network for the synchronization of chaotic systems. *IEEE Trans Emerging Top Comput Intelligence* (2024) 8(1):364–76. doi:10.1109/tetci.2023.3301793

34. Zhou L, Zhang H, Tan F, Liu K. Delay-independent control for synchronization of memristor-based BAM neural networks with parameter perturbation and strong mismatch via finite-time technology. *Trans Inst Meas Control* (2024). doi:10.1177/01423312231200514

35. Yao W, Wang CH, Sun YC, Gong SQ, Lin HR. Event-triggered control for robust exponential synchronization of inertial memristive neural networks under parameter disturbance. *Neural Networks* (2023) 164:67–80. doi:10.1016/j.neunet.2023.04.024

36. Ma M, Lu Y. Synchronization in scale-free neural networks under electromagnetic radiation. *Chaos* (2024) 34:033116. doi:10.1063/5.0183487

37. Tang S, He K, Chen L, Fan L, Lei X, Hu RQ. Collaborative cache-aided relaying networks: performance evaluation and system optimization. *IEEE J Selected Areas Commun* (2023) 41(3):706–19. doi:10.1109/jsac.2023.3234693

38. Chen L, Fan L, Lei X, Duong TQ, Nallanathan A, Karagiannidis GK. Relay-assisted federated edge learning: performance analysis and system optimization. *IEEE Trans Commun* (2023) 71(6):3387–401. doi:10.1109/tcomm.2023.3263566

39. Guo X, Bi Z, Wang J, Qin S, Liu S, Qi L. Reinforcement learning for disassembly system optimization problems: a survey. *Int J Netw Dyn Intelligence* (2023) 2(1):1–14. doi:10.53941/ijndi0201001

40. Yu F, Yu Q, Chen H, Kong X, Mokbel AAM, Cai S, et al. Dynamic analysis and audio encryption application in IoT of a multi-scroll fractional-order memristive hopfield neural network. *Fractal and Fractional* (2022) 6(7):370. Article ID 370. doi:10.3390/fractalfract6070370

41. Sha Y, Mou J, Banerjee S, Zhang Y. Exploiting flexible and secure cryptographic technique for multidimensional image based on graph data structure and three-input majority gate. *IEEE Trans Ind Inform* (2024) 20(3):3835–46. doi:10.1109/tii.2023.3281659

42. Yu F, Chen H, Kong X, Yu Q, Cai S, Huang Y, et al. Dynamic analysis and application in medical digital image watermarking of a new multi-scroll neural network with quartic nonlinear memristor. *Eur Phys J Plus* (2022) 137:434. Article ID 434. doi:10.1140/epjp/s13360-022-02652-4

43. Liu X, Mou J, Zhang Y, Cao Y. A new hyperchaotic map based on discrete memristor and meminductor: dynamics analysis, encryption application, and DSP implementation. *IEEE Trans Ind Elect* (2024) 71(5):5094–104. doi:10.1109/tie.2023.3281687

44. Advances in Nonlinear Systems and Networks. Dvances-in-nonlinear-systems-and-networks/magazine (2023). Available at: https://www.frontiersin.org/research-topics/47060/advances-in-nonlinear-systems-and-networks/magazine.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 07 April 2023
DOI 10.3389/fphy.2023.1179469

OPEN ACCESS

EDITED BY
Fei Yu,
Changsha University of Science and Technology, China

REVIEWED BY
Yanzhi Wu,
Kyoto University, Japan
Yongqiang Guan,
Xidian University, China

*CORRESPONDENCE
Linying Xiang,
xianglinying@neuq.edu.cn

SPECIALTY SECTION
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

RECEIVED 04 March 2023
ACCEPTED 24 March 2023
PUBLISHED 07 April 2023

CITATION
Yu Y, Xiang L, Liu B and Xia C (2023), Moment-based analysis of pinning synchronization in complex networks with sign inner-coupling configurations. *Front. Phys.* 11:1179469. doi: 10.3389/fphy.2023.1179469

Moment-based analysis of pinning synchronization in complex networks with sign inner-coupling configurations

Yanying Yu[1], Linying Xiang[1]*, Bo Liu[2] and Chengyi Xia[3]

[1]School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, China, [2]School of Information Engineering, Minzu University of China, Beijing, China, [3]School of Artificial Intelligence, Tiangong University, Tianjin, China

In this paper, pinning synchronization of complex networks with sign inner-coupling configurations is investigated from a moment-based analysis approach. First, two representative non-linear systems with varying dynamics parameters are presented to illustrate the bifurcation of the synchronized regions. The influence of sign inner-coupling configurations on network synchronizability is then studied in detail. It is found that adding negative parameters in the inner-coupling matrix can significantly enhance the network synchronizability. Furthermore, the eigenvalue distribution of the coupling and control matrix in the pinned network is estimated using the spectral moment analysis. Finally, numerical simulations are given for illustration.

KEYWORDS

complex network, spectral moment, pinning control, synchronization, sign inner-coupling

1 Introduction

Synchronization is a typical collective behavior in complex networks [1–5]. In the past two decades, the issues of synchronization, control, and optimization in complex network systems have become focal subjects in network science and engineering [6–31], and numerous works have been reported on such topics as complete synchronization [6], near-synchronization [32], phase synchronization [33], bounded synchronization [34], fixed-time synchronization [35, 36], heterogeneous node dynamics [37], multiplex networks [38], time-delay systems [39–41], and time-varying networks [42, 43].

It has been demonstrated that the local stability of a complex dynamical network under the pinning control can be converted into two independent sub-problems: identifying the synchronized regions of the pinned network and analyzing the scaled eigenvalues of the coupling and control matrix [44]. On one hand, the bifurcation behavior of the synchronized regions has been observed in complex networks with varying node parameters [39, 40, 45]. Various rich bifurcation patterns of the synchronized regions have been found in the pioneer work [45]. On the other hand, the moment-based analysis approach [46–48] has been introduced to successfully estimate the eigenvalue distribution of the coupling and control matrix [49]. Therein, without performing explicit eigenvalue decomposition, the eigenvalue distribution can be estimated only from the network structural parameters and the control mechanism.

It is worth noting that most of the above-reviewed works on network synchronization assume that the inner-coupling matrix consists of zeros and positive parameters. However,

less attention has been paid to the case that the elements in the inner-coupling matrix are negative [50]. Interestingly, negative interactions among the nodes will lead to the enhancement of the synchronization in complex networks [51]. Moreover, a recent work on network controllability has revealed that adding negatively-weighted edges in a signed network can significantly change its average controllability [52]. Indeed, in real-world scenarios, it is more reasonable and accurate to model a complex system using a network with both negative and positive weights on edges. For instance, in social networks, positive edge weights can denote the relations of like and friendships, while negative edge weights on can represent the relations of dislike and foe [53]. Inspired by these observations, a sign inner-coupling matrix with positive and negative parameters is introduced to denote the cooperation and competition relationships, respectively, between the node variables.

The main contributions of this paper are two-fold. First, the influence of sign inner-coupling configurations on network synchronizability is studied. The interesting bifurcation behavior of the synchronized regions is observed in the pinned network with a varying node dynamics parameter. It is shown that the network synchronizability can be improved by adding negative parameters in the inner-coupling matrix, while blindly adding inner-coupling elements with positive parameters may weaken it. This finding provides a good alternative to optimize the network synchronizability. Second, the eigenvalue distribution of the pinned network is analyzed from the moment-based approach. The analytical expressions of the spectral moments for a globally coupled network and a nearest-neighbor coupled network are derived, respectively. It is found that the expected moments depend not only on the structural parameters of the network but also on the control mechanism. The derived expected moments are then used to estimate the eigenvalue distribution. Numerical examples demonstrate the efficiency of the proposed spectral estimation method.

The rest of the paper is organized as follows. Notation and preliminaries are given in Section 2. The influence of sign inner-coupling configurations on network synchronizability is investigated in Section 3. In Section 4, the estimation of the eigenvalues of the coupling and control matrix for two representative regular networks is provided. Section 5 shows the numerical results. Finally, Section 6 concludes the paper.

2 Notation and preliminaries

2.1 Notation

Throughout the paper, let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}^n$ the vector space of n-dimensional real vectors, and $\mathbb{R}^{m\times n}$ the set of $m \times n$ real matrices. Let I_m be an $m \times m$ identity matrix and diag$\{a_1, a_2, \ldots, a_n\}$ an $n \times n$ diagonal matrix. Let $\otimes$ indicate the Kronecker product and $\mathbf{tr}(A)$ the trace of matrix A.

2.2 Graph theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with a node set $\mathcal{V} = \{1, 2, \ldots, N\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A path between two nodes, say i and j, is given by the node sequence $v_1, v_2, \ldots, v_k$, where $v_1 = i$, $v_k = j$, and $(v_l, v_{l+1}) \in \mathcal{E}$. An undirected graph $\mathcal{G}$ is connected if, for any two nodes, there exists a path connecting them. Let $A = (A_{ij}) \in \mathbb{R}^{N\times N}$ denote the adjacency matrix of the undirected graph $\mathcal{G}$. If there is an edge between nodes i and j, then $A_{ij} = A_{ji} = 1$, and $A_{ij} = 0$ $(j \neq i)$ otherwise. The degree of node i is the number of edges directly connected to it and can be denoted by $d_i = \sum_{j=1}^{N} A_{ij}$. The degree sequence of $\mathcal{G}$ is the list of node degrees, denoted by $\{d_1, d_2, \ldots, d_N\}$. The degree matrix is, thus, defined as $D =$ diag$\{d_1, d_2, \ldots, d_N\}$. The corresponding Laplacian matrix is given by $L = D - A$.

2.3 Problem statement

We consider a complex dynamical network of N nodes described by

$$\dot{x}_i(t) = F(x_i(t)) - \sigma \sum_{j=1}^{N} L_{ij} H x_j(t),\ i = 1, 2, \ldots, N, \tag{1}$$

where $x_i(t) = [x_{i1}(t), x_{i2}(t), \ldots, x_{im}(t)]^T \in \mathbb{R}^m$ is the state vector of node i. The non-linear function $F(\cdot)$ is continuously differentiable denoting the self-dynamics of the nodes. $\sigma > 0$ is the global coupling strength. The matrix $H \in \mathbb{R}^{m\times m}$ describes the inner-coupling of the state variables of nodes, while the Laplacian matrix $L = (L_{ij}) \in \mathbb{R}^{N\times N}$ describes the outer-coupling among the nodes. We assume that the network is undirected and connected. If there is a connection between node i and node j, then $L_{ij} = L_{ji} = -1$; otherwise, $L_{ij} = L_{ji} = 0$ $(j \neq i)$. In addition, the diagonal elements of L are given by

$$L_{ii} = -\sum_{j=1, j\neq i}^{N} L_{ij},\ i = 1, 2, \ldots, N, \tag{2}$$

which satisfy the diffusion condition $\sum_{j=1}^{N} L_{ij} = 0$. It can be verified that L is a symmetric and diagonalizable matrix.

Suppose that all the nodes have a common equilibrium $\bar{x}$, satisfying $F(\bar{x}) = 0$. In order to synchronize network (1) at the state $\bar{x}$, pinning control is applied. The pinned network is, thus, described as follows:

$$\dot{x}_i(t) = F(x_i(t)) - \sigma \sum_{j=1}^{N} L_{ij} H x_j(t) - \delta_i \sigma b_i H(x_i(t) - \bar{x}),\quad i = 1, 2, \ldots, N, \tag{3}$$

where the variable δ_i denotes whether node i is under control. If control is directly applied to node i, then $\delta_i = 1$ with $b_i = b > 0$, otherwise $\delta_i = b_i = 0$. Here, b denotes the feedback gain to be designed. Let l $(1 \leq l < N)$ be the number of pinned nodes. Therefore, $\sum_i \delta_i = l$.

Let $e_i(t) = x_i(t) - \bar{x}$ and $E(t) = [e_1^T(t), e_2^T(t), \ldots, e_N^T(t)]^T \in \mathbb{R}^{mN}$. Linearizing system (3) at $\bar{x}$ leads to the following error system:

$$\dot{E}(t) = (I_N \otimes J_F(\bar{x}) - \sigma C \otimes H) E(t), \tag{4}$$

where $J_F(\bar{x})$ is the Jacobian matrix of $F(\cdot)$ evaluated at $\bar{x}$, $C = L + B$ is the coupling and control matrix, and $B =$ diag$\{b_1, b_2, \ldots, b_N\}$ is the feedback gain matrix.

It is worth noting that the matrix C is a real symmetric matrix, which can be written as $\Lambda = \Phi^{-1} C \Phi$, where $\Lambda =$ diag$\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$

with λ_i, $i = 1, 2, \ldots, N$ being the eigenvalues of C, and the columns of Φ are the set of the corresponding eigenvectors. It can be verified that the eigenvalues of the matrix $I_N \otimes J_F(\bar{x}) - \sigma C \otimes H$ and those of $J_F(\bar{x}) - \sigma\lambda_i H$, $i = 1, 2, \ldots, N$ are identical. For convenience, let $\alpha_i = \sigma\lambda_i$, $i = 1, 2, \ldots, N$. It has been demonstrated in the literature that the local stability of the pinned network (3) is determined by the following generic system [49]:

$$\dot{\eta}(t) = (J_F(\bar{x}) - \alpha H)\eta(t), \tag{5}$$

where $\eta(t)$ is a new auxiliary variable.

$\lambda_m(\alpha)$ denotes the maximal real part of the eigenvalues of $J_F(\bar{x}) - \alpha H$. The synchronized region $\mathcal{S}$ is defined as the range of α with $\lambda_m(\alpha) < 0$. The synchronization will be achieved if all the eigenvalues of σC are located inside $\mathcal{S}$.

In summary, pinning synchronization in network (3) is separated into two sub-problems: 1) identifying the synchronized regions and 2) analyzing the eigenvalue distribution of σC. Previous works on the types and bifurcation behavior of synchronized regions assume that the elements in the inner-coupling matrix are either zeros or positive parameters. Here, a zero indicates the absence of a relation between some state variables of nodes, while a positive parameter characterizes the cooperative relationship between two corresponding state variables. However, less attention has been paid to the case of negative or competitive interactions between node variables. In this paper, a more general inner-coupling matrix including negative parameters is considered.

Definition 1. If the elements of matrix H consist of the symbols +, −, and 0, H is then called the sign pattern matrix [50].

For example,

$$H = \begin{bmatrix} - & + & 0 \\ 0 & + & 0 \\ 0 & 0 & - \end{bmatrix} \tag{6}$$

is called a sign pattern matrix, in which 0, +, and − represent zero, positive, and negative parameters, respectively.

If the state variables of nodes are coupled through a sign pattern matrix, the networked system is said to have a sign inner-coupling configuration. Without loss of generality, in what follows, the elements of H are denoted by 1, −1, and 0, where "1" indicates cooperative relationship, "−1" indicates competitive relationship, and "0" indicates that there is no relation between some state variables of nodes.

3 Bifurcation of the synchronized regions

In this section, the influence of sign inner-coupling configurations on network synchronizability is studied in detail. In particular, two representative non-linear systems with varying parameters are given to illustrate the bifurcation of the synchronized regions.

3.1 Lü system

A single Lü system [54] is described as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} a(x_2 - x_1) \\ -x_1x_3 + \gamma x_2 \\ x_1x_2 - \beta x_3 \end{bmatrix},$$

where $\beta = 3$ and $\gamma = 20$. Obviously, $\bar{x} = [0, 0, 0]^T$ is an equilibrium point of the aforementioned Lü system, and the Jacobian matrix of the system is as follows:

$$J_F(\bar{x}) = \begin{bmatrix} -a & a & 0 \\ 0 & 20 & 0 \\ 0 & 0 & -3 \end{bmatrix}.$$

In what follows, three different types of inner-coupling matrices are considered.

(i) When the inner-coupling matrix is chosen as

$$H_{l1} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the corresponding characteristic equation is obtained as follows:

$$f(\lambda, \alpha, a) = (\lambda + 3)\left[\lambda^2 + (a - 20)\lambda - \alpha^2 + (2a + 20)\alpha - 20a\right] = 0.$$

One has $\lambda_1 = -3 < 0$. If $a - 20 > 0$ and $-\alpha^2 + (2a + 20)\alpha - 20a > 0$, then $\lambda_{2,3} < 0$; that is, the pinned network can synchronize at $\bar{x} = [0, 0, 0]^T$. Here, the boundary curves of the synchronized region are represented by $\alpha_1 = -\sqrt{a^2 + 100} + a + 10$, $\alpha_2 = \sqrt{a^2 + 100} + a + 10$, and $a > 20$. In this situation, the synchronized region of the Lü system with varying dynamics parameter a and its boundary curves are given as shown in Figure 1A. The cyan-shaded area denotes the synchronized region in which $\lambda_m(\alpha) < 0$. The magenta line denotes the corresponding boundary curve. These notations will be used for Figures 1B, C and Figure 2.

(ii) When the inner-coupling matrix is chosen as

$$H_{l2} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the corresponding characteristic equation is obtained as follows:

$$f(\lambda, \alpha, a) = (\lambda + 3)\left[\lambda^2 + (a - 20)\lambda - 2\alpha^2 + (2a + 20)\alpha - 20a\right] = 0.$$

One has $\lambda_1 = -3 < 0$. If $a - 20 > 0$ and $-2\alpha^2 + (2a + 20)\alpha - 20a > 0$, then $\lambda_{2,3} < 0$; that is, the pinned network can synchronize at $\bar{x} = [0, 0, 0]^T$. Here, the boundary curves of the synchronized region are represented by $\alpha_1 = -\sqrt{a^2/4 - 5a + 25} + a/2 + 5$, $\alpha_2 = \sqrt{a^2/4 - 5a + 25} + a/2 + 5$ and $a > 20$. In this situation, the synchronized region of the Lü system with varying dynamics parameter a and its boundary curves are given as shown in Figure 1B.

(iii) When the inner-coupling matrix is chosen as

$$H_{l3} = \begin{bmatrix} -1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the corresponding characteristic equation is obtained as follows:

$$f(\lambda, \alpha, a) = (\lambda + 3)\left[\lambda^2 + (a - 20)\lambda + (2a + 20)\alpha - 20a\right] = 0.$$

One has $\lambda_1 = -3 < 0$. If $a - 20 > 0$ and $(2a + 20)\alpha - 20a > 0$, then $\lambda_{2,3} < 0$; that is, the pinned network can synchronize at $\bar{x} = [0, 0, 0]^T$.

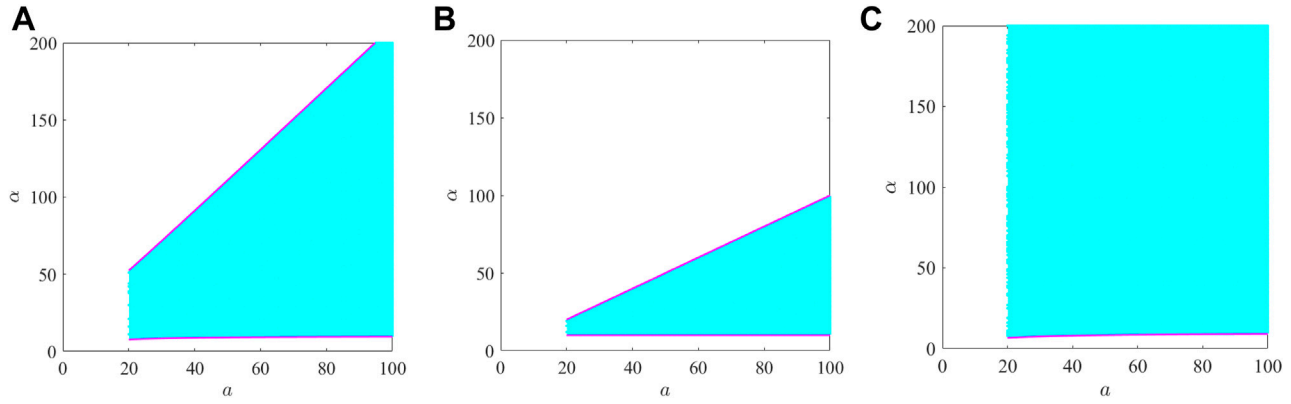


FIGURE 1
Synchronized regions of the Lü system with varying dynamics parameter *a* and its boundary curves.

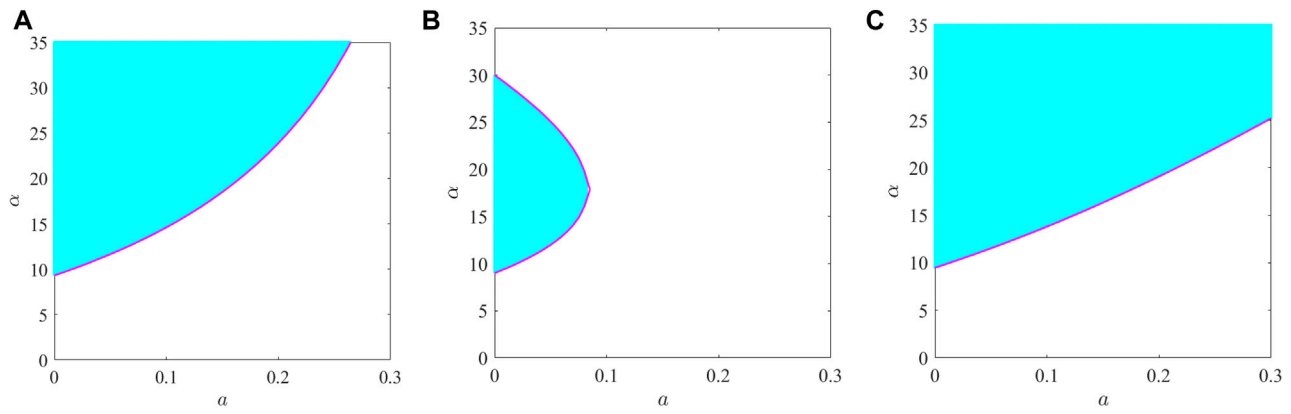


FIGURE 2
Synchronized regions of the unified chaotic system with varying dynamics parameter *a* and its boundary curves.

Here, the boundary curves of the synchronized region are represented by $\alpha_0 = \frac{20a}{2a+20}$ and $a > 20$. In this situation, the synchronized region of the Lü system with varying dynamics parameter a and its boundary curves are given as shown in Figure 1C.

Figure 1 shows the synchronized regions of Lü system for three different sign inter-coupling matrices. Table 1 further summarizes the synchronized regions for three specific values of a. It can be observed from Figures 1A, B that the synchronized region switches from "empty set" to "bounded region" with the increase in the dynamics parameter a, while in Figure 1C, the synchronized region switches from "empty set" to "unbounded region."

3.2 Unified chaotic system

A single unified chaotic system [55] is described as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} (25a+10)(x_2 - x_1) \\ (28-35a)x_1 - x_1x_3 + (29a-1)x_2 \\ x_1x_2 - \frac{a+8}{3}x_3 \end{bmatrix},$$

where $a \in [0, 1]$. Obviously, $\bar{x} = [0, 0, 0]^T$ is an equilibrium point of the aforementioned unified chaotic system, and the Jacobian matrix of the system is as follows:

$$J_F(\bar{x}) = \begin{bmatrix} -(25a+10) & (25a+10) & 0 \\ 28-35a & 29a-1 & 0 \\ 0 & 0 & -\frac{a+8}{3} \end{bmatrix}.$$

Then, we consider the following three types of different inner-coupling matrices:

(i) We set the inner-coupling matrix as

$$H_{u1} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

TABLE 1 Synchronized regions of the Lü system under different sign inter-coupling matrices.

a	H_{l1}	H_{l2}	H_{l3}
36	(8.64, 83.36)	(10, 36)	(7.83, ∞)
46	(8.93, 103.07)	(10, 46)	(8.21, ∞)
56	(9.11, 122.89)	(10, 56)	(8.48, ∞)

TABLE 2 Synchronized regions of the unified chaotic system under different sign inter-coupling matrices.

a	H_{u1}	H_{u2}	H_{u3}
0.05	(11.64, ∞)	(11.99, 25.07)	(11.64, ∞)
0.15	(18.5, ∞)	∅	(16.325, ∞)
0.25	(31.88, ∞)	∅	(21.97, ∞)

The corresponding characteristic equation is obtained as follows:

$$f(\lambda,\alpha,a)=\left(\lambda+\frac{a+8}{3}\right)\left[\lambda^2+(11-4a+\alpha)\lambda+(29-64a)\alpha-(25a+10)(27-6a)\right]=0.$$

One obtains $\lambda_1=-\frac{a+8}{3}<0$. If $11-4a+\alpha>0$ and $(29-64a)\alpha-(25a+10)(27-6a)>0$, then $\lambda_{2,3}<0$; that is, the pinned network can synchronize at $\bar{x}=[0,0,0]^T$. Here, the boundary curve of the synchronized region is represented by $\alpha_0=(25a+10)(27-6a)/29-64a$. When the inner-coupling matrix is set as H_{u1}, the synchronized region of unified chaotic system with varying dynamics parameter a and its boundary curve are illustrated in Figure 2A.

(ii) We set the inner-coupling matrix as

$$H_{u2}=\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix}.$$

The corresponding characteristic equation is obtained as follows:

$$f(\lambda,\alpha,a)=\left(\lambda+\frac{a+8}{3}\right)\left[\lambda^2+(11-4a+\alpha)\lambda-\alpha^2-(39a-39)\alpha-(25a+10)(27-6a)\right]=0.$$

One has $\lambda_1=-\frac{a+8}{3}<0$. If $11-4a+\alpha>0$ and $-\alpha^2-(39a-39)\alpha-(25a+10)(27-6a)>0$, then $\lambda_{2,3}<0$; that is, the pinned network can synchronize at $\bar{x}=[0,0,0]^T$. Here, the boundary curves of the synchronized region are represented by $\alpha_1=\sqrt{4(25a+10)(6a-27)+(39a-39)^2}/2-(39a-39)/2$, $\alpha_2=-\sqrt{4(25a+10)(6a-27)+(39a-39)^2}/2-(39a-39)/2$. When the inner-coupling matrix is set as H_{u2}, the synchronized region of the unified chaotic system with varying dynamics parameter a and its boundary curves are shown in Figure 2B.

(iii) We set the inner-coupling matrix as

$$H_{u3}=\begin{bmatrix}1&1&0\\-1&0&0\\0&0&0\end{bmatrix}.$$

The corresponding characteristic equation is obtained as

$$f(\lambda,\alpha,a)=\left(\lambda+\frac{a+8}{3}\right)\left[\lambda^2+(11-4a+\alpha)\lambda+\alpha^2+(19-89a)\alpha-(25a+10)(27-6a)\right]=0.$$

One has $\lambda_1=-\frac{a+8}{3}<0$. If $11-4a+\alpha>0$ and $\alpha^2+(19-89a)\alpha-(25a+10)(27-6a)>0$, then $\lambda_{2,3}<0$; that is, the pinned network can synchronize at $\bar{x}=[0,0,0]^T$. Here, the boundary curves of the synchronized region are represented by $\alpha_1=\sqrt{4(25a+10)(6a-27)+(19-89a)^2}/2-(19-89a)/2$ and $\alpha_2=-\sqrt{4(25a+10)(6a-27)+(19-89a)^2}/2-(19-89a)/2$. When the inner-coupling matrix is set as H_{u3}, the synchronized region of the unified chaotic system with varying dynamics parameter a and its boundary curves are shown in Figure 2C.

Figure 2 shows the synchronized regions of the unified chaotic system for three different sign inter-coupling matrices. Table 2 further summarizes the synchronized regions for three specific values of a. It can be observed from Figures 2A, C that the synchronized region switches from "unbounded region" to "empty set" with the increase in the dynamics parameter a, while in Figure 2B, the synchronized region switches from "bounded region" to "empty set."

In summary, there exist bifurcation phenomena in the synchronized regions of complex networks for some specific inner-coupling matrices. The synchronized region can evolve with the varying node dynamics parameter and switch from one type to another type.

Given the node dynamics, the larger the range of the synchronized region corresponding to the sign inner-coupling matrix, the easier it is for the network to achieve synchronization. From the aforementioned simulations, the following conclusions can be drawn:

(i) From Figure 1, it can be seen that when the inner-coupling matrix is chosen as H_{l2}, the synchronized region is smaller than that of H_{l1}. In Figure 2, when the inner-coupling matrix is chosen as H_{u2}, the synchronized region is smaller than that of H_{u1}. It can be seen that H_{l2} and H_{u2} add a cooperative inner-coupling element to H_{l1} and H_{u1}, respectively. This means that blindly adding positive parameters in the inner-coupling matrix may weaken the synchronizability of the network.

(ii) From Figure 1, it can be seen that when the inner-coupling matrix is chosen as H_{l3}, a larger synchronized region is obtained compared to H_{l1}. From Figure 2, it can be seen that when the inner-coupling matrix is chosen as H_{u3}, a larger synchronized region is obtained compared to H_{u1}. It can be observed that H_{l3} and H_{u3} add a competitive inner-coupling element to H_{l1} and H_{u1}, respectively. This implies that the network synchronizability can be significantly enhanced by adding a small number of negative parameters in the inner-coupling matrix.

Remark 1. It should be pointed out that although numerical simulations are performed with the aforementioned two chaotic systems, the extension to other general systems is straightforward.

Remark 2. Recall that the bifurcation behavior of the synchronized regions in a network with a varying node dynamics parameter is analyzed in this section. The assumption that all the nodes have a common equilibrium can ensure that the boundary curves of the synchronized region can be analytically derived. It is found that the boundary curves of the synchronized region are related to the varying node dynamics parameter.

4 Spectral analysis of pinned networks

In this section, the spectral moment method [46] is applied to estimate the eigenvalues of C [49].

4.1 Spectral moments of the matrix C

The nth-order spectral moment of C is defined as

$$\mathcal{Q}_n(C) = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^n = \frac{1}{N}\mathbf{tr}(C^n) = \frac{1}{N}\mathbf{tr}(L+B)^n. \tag{7}$$

The first three spectral moments of C can be obtained as follows:

$$\begin{cases} \mathcal{Q}_1(C) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(d_i + b_i), \\ \mathcal{Q}_2(C) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(d_i^2 + d_i + 2d_ib_i + b_i^2), \\ \mathcal{Q}_3(C) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(d_i^3 + 3d_i^2 + 3d_i^2b_i + 3d_ib_i + 3d_ib_i^2 + b_i^3 - 2t_i), \end{cases} \tag{8}$$

where b_i is the feedback gain, d_i is the degree of node i, and t_i is the number of triangles touching node i.

4.2 Globally coupled network

We consider a globally coupled network composed of N nodes, in which any two nodes are directly connected by an edge. The degree distribution of nodes of the globally coupled network is

$$\begin{aligned} \delta_{N-1} = \delta(d_i - (N-1)) = 0, \quad \text{for } d_i \neq N-1, \\ \int_{-\infty}^{\infty}\delta_{N-1}(d_i)\,dx = 1, \quad \text{for } d_i = N-1. \end{aligned} \tag{9}$$

The first three expected moments of node degree are obtained by

$$\begin{cases} \mathbb{E}[d_i] = (N-1), \\ \mathbb{E}\left[d_i^2\right] = (N-1)^2, \\ \mathbb{E}\left[d_i^3\right] = (N-1)^3. \end{cases} \tag{10}$$

The number of connected triples centered on any node in the globally coupled network is

$$\binom{N-1}{2} = \frac{1}{2}(N-1)(N-2). \tag{11}$$

When the pinned nodes are consecutively distributed in the network, the first three expected moments of C can, thus, be derived as

TABLE 3 Moments for a globally coupled network.

Moment order	1st	2nd	3rd
Numerical realization	18.52	379.5	8259.71
Analytical expectations	18.52	379.5	8259.71
Relative error	0	0	0

$$\begin{cases} \mathbb{E}[\mathcal{Q}_1(C)] = N - 1 + b\dfrac{l}{N}, \\ \mathbb{E}[\mathcal{Q}_2(C)] = (N-1)^2 + N - 1 + 2(N-1)b\dfrac{l}{N} + b^2\dfrac{l}{N}, \\ \mathbb{E}[\mathcal{Q}_3(C)] = (N-1)^3 + 3(N-1)^2 - (N-1)(N-2) \\ \qquad\qquad +3\left[(N-1)^2 + N - 1\right]b\dfrac{l}{N} + 3(N-1)b^2\dfrac{l}{N} + b^3\dfrac{l}{N}. \end{cases} \tag{12}$$

Example 1. We consider a globally coupled network of $N = 17$ nodes. Here, only $l = 4$ consecutively distributed nodes are pinned with $b = 10.4$. Table 3 compares the numerical values of the moments of C with the analytical predictions in (13). It shows that the analytical expectations of the moments are exactly the same as the numerical realizations.

4.3 Nearest-neighbor coupled network

Consider a nearest-neighbor coupled network of N nodes, in which each node is only connected to its $2k$ nearest-neighbor nodes. The degree distribution of nodes of the nearest-neighbor coupled network is

$$\begin{aligned} \delta_{2k} = \delta(d_i - 2k) = 0, \quad \text{for } d_i \neq 2k, \\ \int_{-\infty}^{\infty}\delta_{2k}(d_i)\,dx = 1 \quad \text{for } d_i = 2k. \end{aligned} \tag{13}$$

Then, one obtains the first three expected moments of node degree as follows:

$$\begin{cases} \mathbb{E}[d_i] = 2k, \\ \mathbb{E}\left[d_i^2\right] = 4k^2, \\ \mathbb{E}\left[d_i^3\right] = 8k^3. \end{cases} \tag{14}$$

The number of connected triples centered on any node in the network is

$$\binom{2k}{2} = k(2k-1). \tag{15}$$

When the pinning nodes are uniformly distributed in the network, the first three expected moments of C are then obtained by

$$\begin{cases} \mathbb{E}[\mathcal{Q}_1(C)] = 2k + b\dfrac{l}{N}, \\ \mathbb{E}[\mathcal{Q}_2(C)] = 4k^2 + 2k + 2kb\dfrac{l}{N} + b^2\dfrac{l}{N}, \\ \mathbb{E}[\mathcal{Q}_3(C)] = 8k^3 + 8k^2 + 2k + 12k^2b\dfrac{l}{N} \\ \qquad\qquad + 6kb\dfrac{l}{N} + 6kb^2\dfrac{l}{N} + b^3\dfrac{l}{N}. \end{cases} \tag{16}$$

Example 2. We consider a nearest-neighbor coupled network with $N = 200$ and $k = 6$. It is assumed that $l = \sum_i\delta_i = 20$ uniformly

TABLE 4 Moments for a nearest-neighbor coupled network.

Moment order	1st	2nd	3rd
Numerical realization	13.17	197.77	3270.53
Analytical expectations	13.17	197.77	3228.53
Relative error	0	0	1.3%

distributed nodes are pinned with $i = 1, 11, \ldots, 191$. We set $b = 11.7$. Table 4 compares the numerical values of the moments with the analytical predictions in (17). It shows clearly that the analytical expectations of the moments are suited to capture the spectral property of the matrix C.

Remark 3. In this paper, the spectral moment method is extended to the aforementioned two kinds of regular networks. The relationship between the lower-order expected moments and the local structural properties, control scheme including feedback gain and the number of pinned nodes, together with their distributions (i.e., the positions of pinned nodes in the whole network), is established. Note that other network models, such as ER random networks, Chung-Lu random networks, and NW small-world networks, have been given to verify the efficiency of the moment-based estimation method [49].

4.4 Triangular reconstruction of matrix C

In this section, the triangular reconstruction method [56] is generalized to estimate the bounds of the eigenvalues.

We define a triangular distribution $T(\lambda)$ based on a set of abscissas $p_1 \le p_2 \le p_3$ as

$$T(\lambda) = \begin{cases} \dfrac{h}{p_2 - p_1}(\lambda - p_1), & \text{for } \lambda \in [p_1, p_2), \\ \dfrac{h}{p_2 - p_3}(\lambda - p_3), & \text{for } \lambda \in [p_2 p_3], \\ 0, & \text{otherwise}, \end{cases}$$

with $h = K/(p_3 - p_1)$ and $K > 0$. The expected moments of C are obtained as follows:

$$\begin{cases} \mathbb{E}[\mathcal{Q}_1(C)] = \dfrac{1}{3}(p_1 + p_2 + p_3), \\ \mathbb{E}[\mathcal{Q}_2(C)] = \dfrac{1}{6}(p_1^2 + p_2^2 + p_3^2 + p_1 p_2 + p_1 p_3 + p_2 p_3), \\ \mathbb{E}[\mathcal{Q}_3(C)] = \dfrac{1}{10}(p_1^3 + p_1^2 p_2 + p_1^2 p_3 + p_2^3 + p_2^2 p_1 \\ \qquad + p_2^2 p_3 + p_3^3 + p_3^2 p_1 + p_3^2 p_2 + p_1 p_2 p_3). \end{cases} \tag{17}$$

For simplicity, we use $\bar{\mathcal{Q}}$ to represent $\mathbb{E}[\mathcal{Q}(C)]$. The aim is to find a set of abscissas $\{p_1, p_2, p_3\}$ so as to fit a given set of expected moments $\{\bar{\mathcal{Q}}_1, \bar{\mathcal{Q}}_2, \bar{\mathcal{Q}}_3\}$. Using the symmetries of the polynomials, the values of $\{p_1, p_2, p_3\}$ can be determined as roots of the polynomial

$$p^3 - s_1 p^2 + s_2 p - s_3 = 0, \tag{18}$$

with

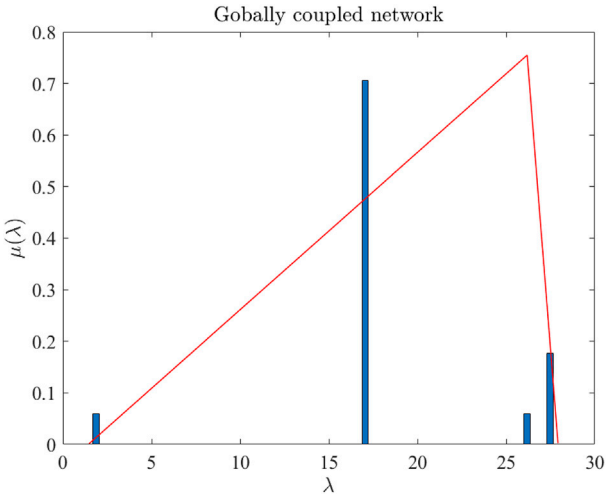


FIGURE 3
Eigenvalue histograms of C and its triangular approximations for a globally coupled network.

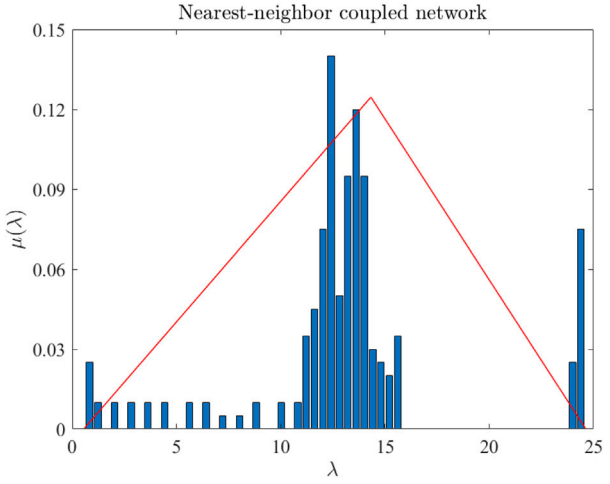


FIGURE 4
Eigenvalue histograms of C and its triangular approximations for a nearest-neighbor coupled network.

$$\begin{cases} s_1 = 3\bar{\mathcal{Q}}_1, \\ s_2 = 9\bar{\mathcal{Q}}_1^2 - 6\bar{\mathcal{Q}}_2, \\ s_3 = 27\bar{\mathcal{Q}}_1^3 - 36\bar{\mathcal{Q}}_1\bar{\mathcal{Q}}_2 + 10\bar{\mathcal{Q}}_3. \end{cases} \tag{19}$$

Example 3. We consider again a 17-node globally coupled network (as shown in Example 1) and a 200-node nearest-neighbor coupled network (as shown in Example 2), respectively. For the globally coupled network, the abscissas for the triangular function are $p_1 = 1.4359$, $p_2 = 26.1860$, and $p_3 = 27.9311$ with $h = 20/(p_3 - p_1)$. For the nearest-neighbor coupled network, the abscissas for the triangular function are $p_1 = 0.5547$, $p_2 = 14.3239$, and $p_3 = 24.6314$ with $h = 3/(p_3 - p_1)$. Figures 3, 4 show the eigenvalue histograms of C and triangular

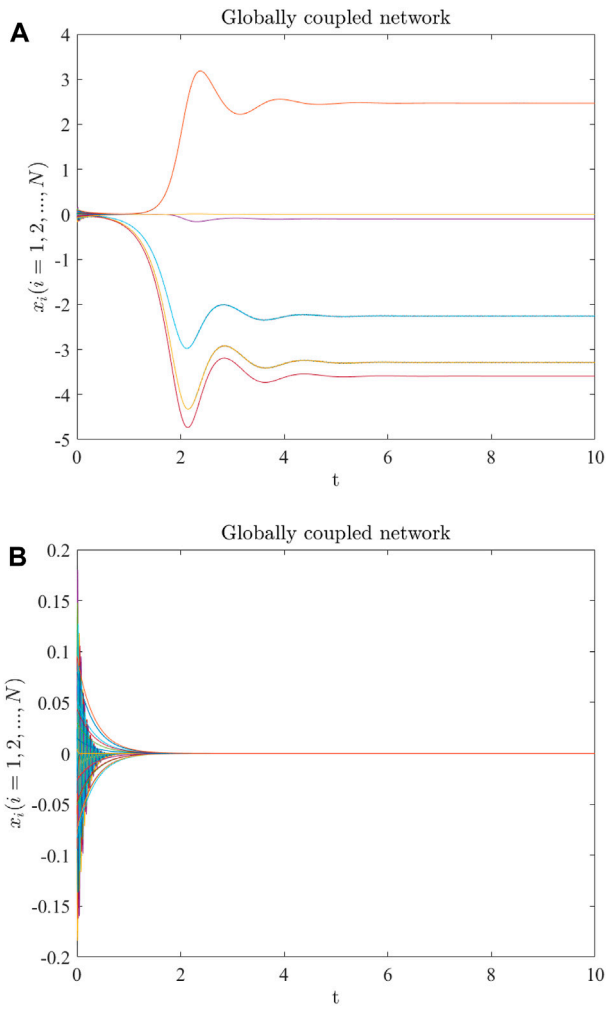


FIGURE 5
Evolution of the node states in a 17-node globally coupled network of Lü systems with **(A)** σ = 5; **(B)** σ = 6.

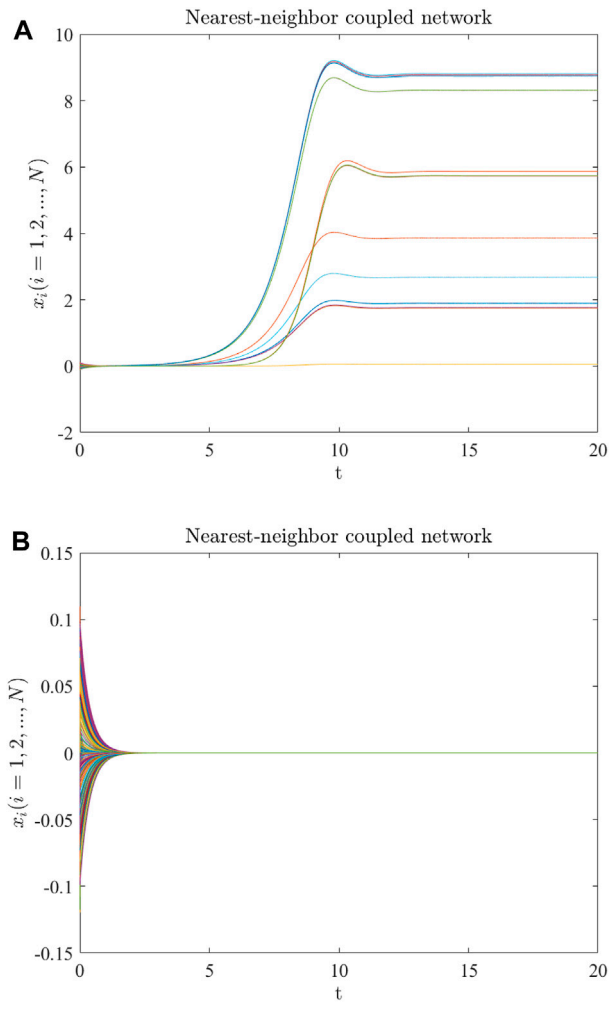


FIGURE 6
Evolution of the node states in a 200-node nearest-neighbor coupled network of unified chaotic systems with **(A)** σ = 15; **(B)** σ = 22.

approximations for the globally coupled network and the nearest-neighbor coupled network, respectively. The ordinate $\mu(\lambda)$ denotes the percentage of the eigenvalue with a certain value in all eigenvalues. It can be seen from Figures 3, 4 that the triangular function can well fit the eigenvalue distribution of the matrix C.

5 Numerical results

We consider the globally coupled network in Example 1 and set the Lü system as the node dynamics. When the node dynamics parameter a = 36 and the inner-coupling matrix is chosen as H_{l3}, it can be obtained from Table 1 that the corresponding synchronized region of the Lü system is (7.83, ∞). According to Example 3, $\bar{\lambda}_1$ = 1.4359 and $\bar{\lambda}_N$ = 27.9311 are good estimations of the lower and upper bounds of the eigenvalues, respectively. The 17-node globally coupled network of Lü systems can achieve synchronization if $\sigma \in (7.83/\bar{\lambda}_1, \infty) = (5.45, \infty)$. Figures 5A, B show the evolution of the node states with σ = 5$\notin$(5.45, ∞) and σ = 6 $\in$ (5.45, ∞), respectively. The numerical results are in good agreement with the theoretical results.

We consider the nearest-neighbor coupled network in Example 2 and set the unified chaotic system as the node dynamics. When the parameter a = 0.05 and the inner-coupling matrix is set as H_{u3}, it can be obtained from Table 2 that the corresponding synchronized region of the unified chaotic system is (11.64, ∞). From Example 3, $\bar{\lambda}_1$ = 0.5547 and $\bar{\lambda}_N$ = 24.6314 are the bound estimations of the eigenvalues. The nearest-neighbor coupled network of unified chaotic systems can achieve synchronization if $\sigma \in (11.64/\bar{\lambda}_1, \infty) = (20.98, \infty)$. Figures 6A, B show the evolution of the node states with σ = 15$\notin$(20.98, ∞) and σ = 22 $\in$ (20.98, ∞), respectively. The numerical simulations are in good agreement with the theoretical results.

6 Conclusion

In this paper, pinning synchronization of complex networks with sign inner-coupling configurations has been investigated. The bifurcation behavior of the synchronized regions has been observed,

and the effect of sign inner-coupling configurations on network synchronizability has been studied in detail. It is shown that the synchronized region can evolve with the varying node dynamics parameter and switch from one type to another type. It is also found that the network synchronizability can be significantly improved by adding negative parameters in the inner-coupling matrix, while blindly adding inner-coupling elements with positive parameters may weaken it. The expected moments of C for the globally coupled network and nearest-neighbor coupled network have been derived. The shape of the eigenvalue distribution of C for each of the aforementioned regular networks can, thus, be estimated to predict pinning synchronization of the network.

It is worth noting that the obtained results in this paper can be generalized to handle control problems with directed topologies or switching topologies. However, directed topology implies that the network is not symmetric, and switching topology means that the network is time-varying. From a technical perspective, this introduces more challenges than its undirected and time-invariant counterpart. In the future, it will be interesting to study the higher-order moments of the matrix C and their corresponding fitting functions. Moreover, pinning synchronization of multiplex networks with time delays [57, 58], noise [59], and disturbances [60, 61] is more challenging but worthy of deep investigation.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

All authors designed and conducted the research. YY and LX performed the analytical and numerical calculations. YY, LX, BL, and CX were the lead writers of the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China (Nos. 61973064 and 62173355), Natural Science Foundation of Tianjin (No. 22JCZDJC00550), and Natural Science Found ation of Hebei Province of China (Nos. F2022501024 and F2019501126).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Chen G. Pinning control of complex dynamical networks. *IEEE Trans Consum Electron* (2022) 68(4):336–43. doi:10.1109/tce.2022.3200488

2. Chen G. Searching for best network topologies with optimal synchronizability: A brief review. *IEEE/CAA J Autom Sin* (2022) 9(4):573–7. doi:10.1109/jas.2022.105443

3. Chen G. Pinning control and synchronization on complex dynamical networks. *Int J Control Autom Syst* (2014) 12:221–30. doi:10.1007/s12555-014-9001-2

4. Tang Y, Jin X, Shi Y, Du W. Event-triggered attitude synchronization of multiple rigid body systems with velocity-free measurements. *Automatica* (2022) 143:110460. doi:10.1016/j.automatica.2022.110460

5. Wu Y, Liu L, Hu J, Feng G. Adaptive antisynchronization of multilayer reaction-diffusion neural networks. *IEEE Trans Neural Networks Learn Syst* (2018) 29(4):807–18. doi:10.1109/tnnls.2017.2647811

6. Pecora LM, Carroll TL. Master stability functions for synchronized coupled systems. *Phys Rev Lett* (1998) 80:2109–12. doi:10.1103/physrevlett.80.2109

7. Wen G, Yu W, Hu G, Cao J, Yu X. Pinning synchronization of directed networks with switching topologies: A multiple lyapunov functions approach. *IEEE Trans Neural Networks Learn Syst* (2015) 26(12):3239–50. doi:10.1109/tnnls.2015.2443064

8. Yu F, Shen H, Yu Q, Kong X, Kumar Sharma P, Cai S. Privacy protection of medical data based on multi-scroll memristive Hopfield neural network. *IEEE Trans Netw Sci Eng* (2023) 10(2):845–58. doi:10.1109/tnse.2022.3223930

9. Yu F, Kong X, Mokbel AAM, Yao W, Cai S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive Hopfield neural network with a novel local active memeristor. *IEEE Trans Circuits Systems-II: Express Briefs* (2023) 70(1):326–30. doi:10.1109/tcsii.2022.3218468

10. Wan Q, Li F, Chen S, Yang Q. Symmetric multi-scroll attractors in magnetized Hopfield neural network under pulse controlled memristor and pulse current stimulation. *Chaos, Solitons Fractals* (2023) 169:113259. doi:10.1016/j.chaos.2023.113259

11. Wang F, Ni Y, Liu Z, Chen Z. Containment control for general second-order multiagent systems with switched dynamics. *IEEE Trans Cybern* (2020) 50(2):550–60. doi:10.1109/tcyb.2018.2869706

12. Jiang Y, Wang F, Liu Z, Chen Z. Composite learning adaptive tracking control for full-state constrained multiagent systems without using the feasibility condition. *IEEE Trans Neural Networks Learn Syst* (2022) 1–13. doi:10.1109/tnnls.2022.3190286

13. Xia C, Hu Z, Zhao D. Costly reputation building still promotes the collective trust within the networked population. *New J Phys* (2022) 24:083041. doi:10.1088/1367-2630/ac8898

14. Li C, Feng B, Li S, Kurths J, Chen G. Dynamic analysis of digital chaotic maps via state-mapping networks. *IEEE Trans Circuits Syst Regular Pap* (2019) 66(6):2322–35. doi:10.1109/tcsi.2018.2888688

15. Feng J, Zhang L, Wang J, Zhao Y. The synchronization of complex dynamical networks with discontinuous dynamics and exogenous disturbances. *Asian J Control* (2021) 23(6):2837–48. doi:10.1002/asjc.2414

16. Liao H, Mariani MS, Medo M, Zhang Y-C, Zhou M-Y. Ranking in evolving complex networks. *Phys Rep* (2017) 689:1–54. doi:10.1016/j.physrep.2017.05.001

17. Zhou M-Y, Zhuo Z, Liao H, Fu Z-Q, Cai S-M. Enhancing speed of pinning synchronizability: Low-degree nodes with high feedback gains. *Sci Rep* (2015) 5:17459. doi:10.1038/srep17459

18. Ruan X, Feng J, Xu C, Wang J, Zhao Y. Dynamic event-triggered pinning synchronization for switched impulsive complex networks with asynchronous switching. *IEEE Trans Circuits Syst Express Briefs* (2022) 69(4):2211–5. doi:10.1109/tcsii.2021.3123285

19. Chen F, Chen J. Minimum-energy distributed consensus control of multi-agent systems: A network approximation approach. *IEEE Trans Autom Control* (2020) 65(3):1144–59. doi:10.1109/tac.2019.2917279

20. Chen F, Feng G, Liu L, Ren W. Distributed average tracking of networked Euler-Lagrange systems. *IEEE Trans Autom Control* (2015) 60(2):547–52. doi:10.1109/tac.2014.2343111

21. Wen G, Yu X, Yu W, Lü J. Coordination and control of complex network systems with switching topologies: A survey. *IEEE Trans Syst Man, Cybern Syst* (2021) 51(10): 6342–57. doi:10.1109/tsmc.2019.2961753

22. Chen K, He W, Han Q-L, Xue M, Tang Y. Leader selection in networks under switching topologies with antagonistic interactions. *Automatica* (2022) 142:110334. doi:10.1016/j.automatica.2022.110334

23. Yi X, Yang T, Wu J, Johansson KH. Distributed event-triggered control for global consensus of multi-agent systems with input saturation. *Automatica* (2019) 100:1–9. doi:10.1016/j.automatica.2018.10.032

24. Wang X, Chen G. Synchronization in scale-free dynamical networks: Robustness and fragility. *IEEE Trans Circuits Syst Fundam Theor Appl* (2002) 49(1):54–62. doi:10.1109/81.974874

25. Barahona M, Pecora LM. Synchronization in small-world systems. *Phys Rev Lett* (2002) 89(5):054101. doi:10.1103/physrevlett.89.054101

26. Li X, Wang X, Chen G. Pinning a complex dynamical network to its equilibrium. *IEEE Trans Circuits Syst Regular Pap* (2004) 51(10):2074–87. doi:10.1109/tcsi.2004.835655

27. Wang X, Chen G. Pinning control of scale-free dynamical networks. *Physica A: Stat Mech its Appl* (2002) 310(3-4):521–31. doi:10.1016/s0378-4371(02)00772-0

28. Moradi Amani A, Jalili M, Yu X, Stone L. Finding the most influential nodes in pinning controllability of complex networks. *IEEE Trans Circuits Syst Express Briefs* (2017) 64(6):685–9. doi:10.1109/tcsii.2016.2601565

29. Liu H, Xu X, Lu J-A, Chen G, Zeng Z. Optimizing pinning control of complex dynamical networks based on spectral properties of grounded Laplacian matrices. *IEEE Trans Syst Man, Cybern Syst* (2021) 51(2):786–96. doi:10.1109/tsmc.2018.2882620

30. Shao S, Cao J, Hu Y, Liu X. Prespecified-time distributed synchronization of Lur'e networks with smooth controllers. *Asian J Control* (2022) 24(1):125–36. doi:10.1002/asjc.2422

31. He S, Wang H, Yu W. Distributed fast finite-time tracking consensus of multi-agent systems with a dynamic leader. *IEEE Trans Circuits Syst Express Briefs* (2022) 69(4):2176–80. doi:10.1109/tcsii.2021.3125700

32. Sun J, Bollt EM, Nishikawa T. Master stability functions for coupled nearly identical dynamical systems. *EPL* (2016) 85(6):60011. doi:10.1209/0295-5075/85/60011

33. Zhou C, Kurths J. Noise-induced phase synchronization and synchronization transitions in Chaotic oscillators. *Phys Rev Lett* (2002) 88:230602. doi:10.1103/physrevlett.88.230602

34. Zhu S, Zhou J, Yu X, Lu J-A. Bounded synchronization of heterogeneous complex dynamical networks: A unified approach. *IEEE Trans Autom Control* (2021) 66(4): 1756–62. doi:10.1109/tac.2020.2995822

35. Liu X, Ho D, Song Q, Xu W. Finite/fixed-time pinning synchronization of complex networks with stochastic disturbances. *IEEE Trans Cybern* (2019) 49(6): 2398–403. doi:10.1109/tcyb.2018.2821119

36. Li N, Wu X, Feng J, Xu Y, Lü J. Fixed-time synchronization of coupled neural networks with discontinuous activation and mismatched parameters. *IEEE Trans Neural Networks Learn Syst* (2021) 32(6):2470–82. doi:10.1109/tnnls.2020.3005945

37. Xiang J, Chen G. On the V-stability of complex dynamical networks. *Automatica* (2007) 43(6):1049–57. doi:10.1016/j.automatica.2006.11.014

38. Tang L, Wu X, Lü J, Lu J-A, D'Souza RM. Master stability functions for complete, intralayer, and interlayer synchronization in multiplex networks of coupled Rössler oscillators. *Phys Rev E* (2019) 99(1):012304. doi:10.1103/physreve.99.012304

39. Tang L, Lu J-A, Lü J, Wu X. Bifurcation analysis of synchronized regions in complex dynamical networks with coupling delay. *Int J Bifurcation Chaos* (2014) 24(1): 1450011. doi:10.1142/s0218127414500114

40. Tang L, Wu X, Lü J, Lu J-A. Bifurcation behaviors of synchronized regions in logistic map networks with coupling delay. *Chaos* (2015) 25:033101. doi:10.1063/1.4913854

41. Yao W, Wang C, Sun Y, Zhou C. Robust multimode function synchronization of memristive neural networks with parameter perturbations and time-varying delays. *IEEE Trans Syst Man, Cybern Syst* (2022) 52(1):260–74. doi:10.1109/tsmc.2020.2997930

42. Ghosh D, Frasca M, Rizzo A, Majhi S, Rakshit S, Alfaro-Bittner K, et al. The synchronized dynamics of time-varying networks. *Phys Rep* (2022) 949:1–63. doi:10.1016/j.physrep.2021.10.006

43. Wen G, Zheng W. On constructing multiple Lyapunov functions for tracking control of multiple agents with switching topologies. *IEEE Trans Autom Control* (2019) 64(9):3796–803. doi:10.1109/tac.2018.2885079

44. Xiang L, Chen F, Chen G. Synchronized regions of pinned complex networks: Spectral analysis. *Nonlinear Dyn* (2014) 78(3):1609–28. doi:10.1007/s11071-014-1538-3

45. Tang L, Lu J-A, Lü J, Yu X. Bifurcation analysis of synchronized regions in complex dynamical networks. *Int J Bifurcation Chaos* (2012) 22(11):1250282. doi:10.1142/s0218127412502823

46. Preciado VM, Verghese GC. Low-order spectral analysis of the Kirchhoff matrix for a probabilistic graph with a prescribed expected degree sequence. *IEEE Trans Circuits Syst Regular Pap* (2009) 56(6):1231–40. doi:10.1109/tcsi.2009.2023758

47. Preciado VM, Jadbabaie A, Verghese GC. Structural analysis of Laplacian spectral properties of large-Scale networks. *IEEE Trans Autom Control* (2013) 58(9):2338–43. doi:10.1109/tac.2013.2261187

48. Preciado VM, Jadbabaie A. Moment-based spectral analysis of large-scale networks using local structural information. *IEEE/ACM Trans Networking* (2013) 21(2):373–82. doi:10.1109/tnet.2012.2217152

49. Xiang L, Yu Y, Zhu J. Moment-based analysis of pinning synchronization in complex networks. *Asian J Control* (2022) 24(2):669–85. doi:10.1002/asjc.2590

50. Yin J, Xiang L, Chen F. Pinning synchronization and optimization of complex networks with sign inner-coupling configurations. In: 13th IEEE International Conference on Control and Automation (ICCA). (2017).

51. Nishikawa T, Motter AE. Network synchronization landscape reveals compensatory structures, quantization, and the positive effect of negative interactions. *PNAS* (2010) 107(23):10342–7. doi:10.1073/pnas.0912444107

52. Zhu J, Xiang L, Yu Y, Chen F, Chen G. Average controllability of complex networks with Laplacian dynamics. *IEEE Trans Circuits Syst Regular Pap* (2022) 69(4): 1704–14. doi:10.1109/tcsi.2021.3133650

53. Xiang L, Chen G. Minimal edge controllability of directed networks. *Adv Complex Syst* (2019) 22(7 and 8):1950017. doi:10.1142/s0219525919500176

54. Lü J, Chen G. A new chaotic attractor coined. *Int J Bifurcation Chaos* (2002) 12(3): 659–61. doi:10.1142/s0218127402004620

55. Lü J, Chen G, Cheng D, Celikovský S. Bridge the gap between the Lorenz system and the Chen system. *Int J Bifurcation Chaos* (2002) 12(12):2917–26. doi:10.1142/s021812740200631x

56. Preciado VM. *Spectral analysis for stochastic models of large-scale complex dynamical networks*. Massachusetts Institute of Technology (2009).

57. Liu S, Li C, Hu Q. Cryptanalyzing two image encryption algorithms based on a first-order time-delay system. *IEEE MultiMedia* (2022) 29(1):74–84. doi:10.1109/mmul.2021.3114589

58. Wang D, Wang D, Wang W. Necessary and sufficient conditions for containment control of multi-agent systems with time delay. *Automatica* (2019) 103:418–23. doi:10.1016/j.automatica.2018.12.029

59. Chen L, Wang Y, Ren W, Hou Z-G, Tan M. On convergence rate of leader-following consensus of linear multi-agent systems with communication noises. *IEEE Trans Automatic Control* (2016) 61(11):3586–92. doi:10.1109/tac.2016.2522647

60. Ye M, Li D, Han Q-L, Ding L. Distributed Nash equilibrium seeking for general networked games with bounded disturbances. *IEEE/CAA J Autom Sin* (2022) 9.

61. Sun H, Sun Q, Sun M, Tao J, Chen Z. Accurate modeling and homing control for parafoil delivery system based on wind disturbance rejection. *IEEE Trans Aerospace Electron Syst* (2022) 58(4):2916–34. doi:10.1109/taes.2022.3141033

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 25 April 2023
DOI 10.3389/fphy.2023.1183419

OPEN ACCESS

EDITED BY
Fei Yu,
Changsha University of Science and Technology, China

REVIEWED BY
Gege Yang,
Beijing Information Science and Technology University, China
Xiong Li,
University of Electronic Science and Technology of China, China

*CORRESPONDENCE
Xiaoming Lin,
411833214@qq.com

RECEIVED 10 March 2023
ACCEPTED 10 April 2023
PUBLISHED 25 April 2023

CITATION
Tang W, Lin X, Zhao Y, Zhou M, Wang Z, Xiao Y and Wang J (2023), Reputation-based electricity scheduling scheme for complex network of user electricity consumption.
Front. Phys. 11:1183419.
doi: 10.3389/fphy.2023.1183419

Reputation-based electricity scheduling scheme for complex network of user electricity consumption

Wenjun Tang[1], Xiaoming Lin[2,3]*, Yuming Zhao[1], Mi Zhou[2,3], Zhenshang Wang[1], Yong Xiao[2,3] and Ji Wang[1,2,3]

[1]Shenzhen Power Supply Bureau Co., Ltd., Shenzhen, China, [2]Electric Power Research Institute, CSG, Guangzhou, China, [3]Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid, Guangzhou, China

With the emergence of various high-powered electrical equipment, the demand for electric energy has increased rapidly. Subsequently, it has highlighted some issues of electricity consumption, such as the adjustment of electricity consumption peak. Although many electricity scheduling schemes have been proposed to adjust and control user load of electricity consumption, the current regulation of user load is not accurate and effective because the load regulation of different regional grid users is a complex network system. In this paper, we propose a reputation-based user electricity scheduling scheme for the complex network of user electricity consumption, whose purpose is to accurately adjust the electricity consumption of related users to further improve the adjustment of electricity consumption peak. In our scheme, we first model a complex network of user electricity consumption. Then we construct a reputation calculation method for electricity users, where the calculated reputation of users is one of the basis for assigning scheduling tasks to users and calculating the price subsidy received by users who complete the scheduling tasks. Further, we use the machine learning method to train a computation model to calculate the adjustment coefficients of electricity load, and then the electricity scheduling tasks are adjusted based on the calculated adjustment coefficients. Finally, the corresponding electricity scheduling tasks are assigned to the selected electricity users respectively for adjusting the electricity consumption of these users. Experiment results show the effectiveness of our proposed scheme. Our scheme can effectively calculate the reputation values of users based on their historical data, and the corresponding electricity scheduling tasks are effectively assigned to related users to accurately adjust the electricity consumption of these users according to their reputation values and the real-time adjustment coefficients, so as to efficiently improve the adjustment of electricity consumption peak.

KEYWORDS

reputation, complex network, electricity consumption, adjustment coefficients, electricity scheduling

1 Introduction

1.1 Background

With the continuous innovation of science and technology, various electrical equipment has emerged, which enriches people's lives and provides greater security for people's lives. However, the increase of electrical equipment has created greater demands for electricity. People also need to use many high-power electrical appliances in their daily lives. Therefore, huge electricity loads are generated, which can influence the stability of power grid systems and people's daily life. Figure 1 shows the electricity consumption framework of power grid system. At present, the most common forms of electricity generation in daily life are thermal power generation, hydroelectric power generation, wind power generation and nuclear power generation, etc. Although there are various electricity generation forms and the amount of electricity generated is considerable, all aspects of life need electricity power to drive, leading to the total electricity load of all users is a too large or too small in certain periods, which increases the burden of electricity lines. And it will also increase the maintenance cost of power grid systems.

Due to the diversity and uncertainty of electricity demand, the electricity generation capacity set to meet the maximum demand of users is largely idle during the low demand period. Therefore, it not only increases the cost of electricity generation, but also increases the electricity bills burden of users. In order to change this situation, electric power enterprises have embarked on research and taken measures to manage the electricity load. In the early stage, some studies adjusted the production shift or the commuting time by guiding enterprises. And some studies suggest shut down large electrical equipment during peak hours to achieve off-peak electricity consumption. These schemes improve the load rate of the power grid. Subsequently, some new studies introduce economic incentives linked to the interests of users, and it further encourages users to voluntarily change the time and modes of their electricity consumption. This could further increase the power grid's load rate and reduce users' electricity bills. With the development of science and technology, power enterprises have adopted direct load control technology for some users.

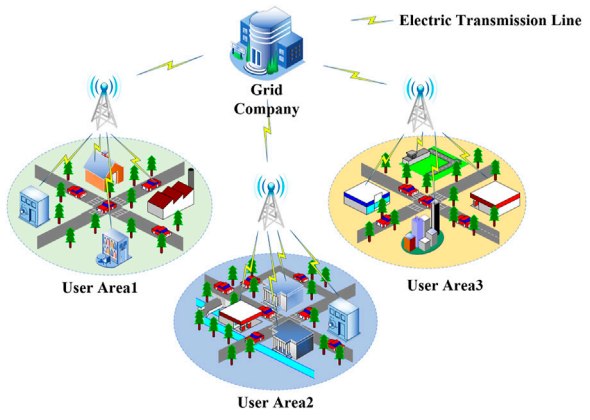


FIGURE 1
Electricity consumption framework of power grid system.

In recent years, many researchers have devoted themselves to reducing the operation cost of electricity systems and improving the quality of electricity consumption in people's daily life. However, many existing traditional solutions are using energy storage to reduce the occurrence of peak and trough of electricity consumption, which solves the problem of storage and waste of electric energy. But these schemes are all used after the occurrence of peak and valley of electricity consumption periods, which uses additional energy storage resources. The emergence of demand-side response avoids the above problems. In order to achieve the demand-side response, there are many studies suggest that indirectly affect the consumption time of users by adjusting the electricity price in different periods. This is a scheme specified for the consumption behavior of users, which prevents the occurrence of peak and valley of electricity consumption. And it does not need additional equipment to store electric energy.

[1] take into account the uncertainty of user's response behavior and construct a high-dimensional parameter space that including multiple influencing factors such as participation, response time and net load power. And this scheme quantifies user's response ability by utilizing the expectation of net load envelope domain before and after response. Then they propose a response capability gradient evaluation index that comprehensively considers power grid incentive cost and user's response cost. There are also some works on the realization of demand-side response by encouraging users to participate in incentive demand response. The incentive demand response requires a contract signed between the power grid and users. The upregulated and downregulated capacity are directly controlled by the scheduling center, which can effectively suppress real-time electricity deviation. In order to fully improve the enthusiasm of users to participate in the incentive demand response, the real-time electricity deviation of the grid could be better suppressed. [2] propose a new idea of users' participation in incentive demand response: for the incentive demand response with faster response speed, the power grid adjusts the incentive price of the incentive demand response according to the real-time electricity deviation. And the power grid allows users to re-declare the incentive demand response capacity of the future incentive demand response operating cycle 1 h before each rolling optimization. Therefore, the scheme achieves the goal that the incentive demand response capacity can increase or decrease with the trend of power deviation. But for the incentive demand response with slower response speed, the contract capacity declare before the day is executed.

1.2 Our contribution

To prevent the occurrence of electricity consumption peaks and valleys in complex network of user electricity consumption, we propose a reputation-based electricity scheduling scheme for complex network of user electricity consumption. In our scheme, we first model a complex network of user electricity consumption, which reflects the relationship and function of each entity in the network. Then in this model, we construct a reputation calculation method for electricity users, and then the users who participate in the incentive demand response of electricity scheduling are selected according to the calculated reputation of users. Further, we use the machine learning

method to train a computation model to calculate the adjustment coefficients, and then the electricity scheduling tasks are adjusted based on the calculated adjustment coefficients. Finally, the corresponding electricity scheduling tasks are assigned to the selected electricity users respectively for realizing electricity peak load shifting. The main contributions of our scheme are summarized as follows.

- We propose a user electricity scheduling task allocation scheme in complex network of user electricity consumption, which can allocate corresponding electricity scheduling tasks for related users according to the calculated reputation of users to improve the reliability of power grid scheduling tasks. A two-layer user selection scheme is constructed to select related users participating in the incentive demand response of electricity scheduling, where these users are filtered based on their historical and real-time electricity consumption data.
- The machine learning method is used to train a computation model to calculate the adjustment coefficients of electricity load, and then the electricity scheduling tasks are adjusted based on the calculated adjustment coefficients.
- Experiment results show our proposed scheme is efficient and effective for adjusting electricity consumption peaks and valleys. Our scheme can effectively calculate the reputation values of users based on their historical data, and the corresponding electricity scheduling tasks are effectively assigned to related users.

1.3 Paper organization

The structure of the rest of this paper is as follows: In Section 2, we describe related works about the adjustment of electricity consumption. In Section 3, a reputation-based electricity scheduling scheme for complex network of user electricity consumption is proposed. In Section 4, we make some experiments to test the efficiency and effectiveness of our scheme. In Section 5, we draw our conclusions.

2 Related works

In order to solve the issues of electricity consumption, many scholars have proposed various schemes for efficient electricity scheduling, including the use of energy storage system in the peak discharge and trough of the power grid charging to alleviate the problem of peak and trough of electricity consumption, a scheme that indirectly affects the consumption habits of users through the change of electricity price, and a scheme that adjusts the consumption behavior of users and other schemes based on incentives.

2.1 Application of complex network in power system

Complex network refers to a network with self-organization, self-similarity, attractor, small world, scale-free, partial or total properties network. Over the years, many researchers have applied complex networks to power systems. In [3], a new methodology for stability assessment of a smart power system was proposed. The key to this assessment was an index called betweenness index which was based on ideas from complex network theory. [4] studied the vulnerability analysis and recognition of key nodes in power grids from a complex network perspective. To effectively analyze the behavior and verify the correctness of node electrical centrality, the net-ability and vulnerability index were introduced to describe the transfer ability and performance under normal operation and assess the vulnerability of the power system under cascading failures, respectively. When a single failure occurs in a vulnerable part of a power system, this may cause a large area cascading event. Therefore, an advanced method that could assess the risks during cascading events is needed. Therefore, [5] proposed an improved complex network model for power system risk assessment. Risk was defined by consequence and probability of the failures in their model, which were affected by both power factors and network structure. Compared with existing risk assessment models, the proposed one can evaluate the risk of the system comprehensively during a cascading event by combining the topological and electrical information. As a promising approach to modeling complex systems, complex networks could provide a sound theoretical framework for developing proper simulation model. [6] proposed a complex network model of the bilateral power transaction market. Unlike other general commodity transactions, both of the financial layer and the physical layer were considered in the model. Through simulation analysis, the feasibility and validity of the model were verified.

2.2 Traditional electricity scheduling schemes

To implement effective electricity scheduling, the traditional schemes use the energy storage method to realize peak load shifting. [7] investigated the potential of peak shaving through battery storage. For each user, they studied the peak load reduction achievable by batteries of varying energy capacities (kWh), ranging from 0.1 to 10 times the mean power (kW). The results showed that for 75% of the users, the peak reduction stayed below 44% when the battery capacity was 10 times the mean power. Furthermore, for 75% of the users the battery remained idle for at least 80% of the time. [8] proposed a decision-tree-based peak shaving algorithm for the islanded micro-grid. The proposed algorithm helped an islanded micro-grid to operate its generation units efficiently; [9] tried to summarize the response modeling for different types of demand-side resources by analyzing the characteristics of different demand-side resources. Then they established a scheduling-response mode, which took optimal peak shaving response as its goal and took the power and electricity quantity as the constraint conditions to solve the model; [10] carried out a long-term distribution system (DS) planning model considering the peak shaving of the Energy station (ES), which was achieved by scheduling the input energy of ES. By regarding DS and ES as different stakeholders, a decentralized framework was devised to shave the electric peak loads in the DS planning, where the coupling relationship between

the time-of-use (TOU) price and exchanged power (e.g., the input power of ES) was clearly expressed. [11] proposed a novel peak load shaving algorithm for peak shaving application in a hybrid photovoltaic (PV) generation system and Battery Energy Storage System connected isolated micro-grid (IMG) system, which helped an IMG system to operate its generation systems optimally and economically along with a PV generation unit.

2.3 Demand-side response

Demand-side response refers to that when the wholesale electricity market price rises or the system reliability is threatened, electricity users change their habitual mode of electricity consumption after receiving the direct compensation notice of induced load reduction or the signal of electricity price rise sent by the electricity supplier. Therefore, the electricity supplier reduces or delays the electricity load in a certain period and responds to the electricity supply. Then it ensures the stability of the power grid system and restrains the short-term behavior of electricity price rise [12]. This scheme is a basic platform for communication between the power grid and users. According to different demand response modes of users, demand-side response technology in the smart power grid can be divided into a price-based and an incentive-based demand side response. According to the different response ways of users, demand response can be divided into two categories: price-based demand response and incentive-based demand response. Price-based demand response can be divided into time-of-use electricity price, real-time electricity price and peak electricity price. Price-based demand response adjusts the consumption time of users through the difference of electricity price in different periods. Compared with the period of high electricity price, users are more inclined to use electricity in the period of low electricity price. Incentive-based demand response refers to the demand response implementer making policies to encourage users to reduce the load when the system burden is large or the electricity price is high. Incentive-based demand response can be divided into direct load control, interruptible load, demand-side bidding, emergency demand response and capacity/auxiliary service plan, etc. The existing incentive-based demand response mainly consists of direct load control and interruptible load.

There are many schemes to implement demand-side response. Some studies indirectly regulate users' electricity consumption behavior through TOU, so as to achieve scheduling goals. [13] derived the optimal capacity investment and pricing decisions for the electricity company. Furthermore, they used real data from a case study to validate the results and derived insights for implementing the TOU tariff. [14] proposed an analytical method that incorporated the TOU strategy into the reliability evaluation of the electricity system. A price-based demand response was modeled to calculate the hourly load using an apportionment method. By using the particle swarm optimization algorithm, a TOU-based optimization model with a penalty was constructed to find the optimal electricity price in their scheme. [15] proposed models of costs to utility companies arising from user demand fluctuations, and models of user satisfaction with the difference between the nominal demand and the actual consumption. They designed utility functions for the company and the users, and obtained a Nash equilibrium using backward induction. Some studies have implemented different measures for different users based on their behaviors, so as to better develop demand response schemes. [16] based on system dynamics, constructed a demand response incentive signal strength analysis model, analyzed the demand response behavior of different users, and calculated demand response incentive subsidies standards for different types of users. [17] decomposed the users' electricity consumption situation by using the additive model, extracted various influencing factors, and constructed the modulus one vector to describe and analyzed the electricity consumption behavior of different users. [18] based on the research of feature optimization method for the behavior analysis of intelligent electricity users in the early stage proposed a cluster number optimization method for user behavior analysis to effectively improve the data clustering effect of electrical behavior analysis. [19] proposed a Python-based K-means plus clustering algorithm to classify Taiyuan electricity data. The K-means plus clustering algorithm classified the data of electricity consumption and finally got five different kinds of users. [20] discussed the cluster analysis of electricity consumption behavior and the selection of demand target users, and proved that the selection could guide the implementation of demand response projects. A reasonable assessment of demand response potential was of great significance for effectively gathering demand-side resources. [21] took Jiangxi Province as an example to predict the load after the analysis of adjustable load, analyzed the adjustable load of users, and evaluated the demand response potential of different users. There are also some other demand-side response schemes. [22] proposed a new economic scheduling model combined with wind power, which considered incentive-based DR And reliability measures. Compared with the traditional model, this model considered the response of the users' electricity consumption to the incentive price. The expected cost of unsupplied energy was added to the target in order to strike an optimal balance between the economy and the reliability of electricity system operation. [23] proposed a safe optimal scheduling model of the electricity system considering the demand response of electricity price under the incentive mechanism of the electricity market aiming at the traditional day-ahead scheduling scheme. Based on the peak valley TOU price, the incentive compensation mechanism should be established to encourage users to actively use it. Participating in demand-side resource scheduling, which made the effect of peak load shifting more obvious.

3 Proposed scheme

In this section, we propose a reputation-based electricity scheduling scheme for complex network of user electricity consumption. In our scheme, we first construct a complex network model of the user's electricity consumption, which reflects the relationship and function of each entity in the network. Then in this model, we construct a reputation calculation method for electricity users, and then the users who participate in the incentive demand response of electricity scheduling are selected according to the calculated reputation of users. Further, we use the machine learning method to train a

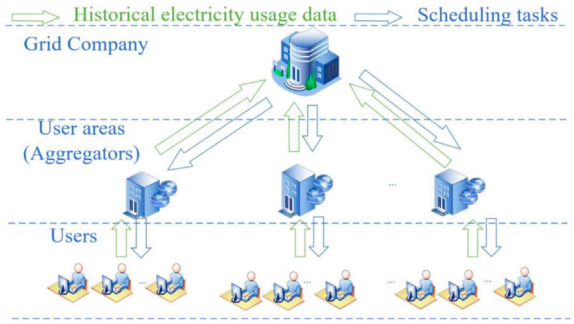


FIGURE 2
System framework.

computation model to calculate the adjustment coefficients, and then the electricity scheduling tasks are adjusted based on the calculated adjustment coefficients. Finally, the corresponding electricity scheduling tasks are assigned to the selected electricity users respectively for realizing electricity peak load shifting.

3.1 System framework

In this section, we show a system framework of user electricity scheduling, which consists of users, user areas (aggregators) and grid company. In our framework, Due to the interconnection of various entities and various interrelationships, the whole power grid system can be regarded as a complex network. In Section 3.2, we introduce the construction of a complex network model for user electricity consumption in detail, which demonstrates the distribution of individual entities in the system and the relationships between individual entities, such as users, aggregators and grid companies.

In the system, various entities cooperate with each other to realize the electricity scheduling. Among them, the electricity data of users is the basis of the whole scheduling scheme. At the same time, the realization of scheduling task also needs the cooperation of electricity users. Due to the big size of the power grid system, interacting directly with area users through the power grid side will incur relatively large resource consumption. Aggregator plays the role of an intermediate agent in the system, the electricity data of users are aggregated and sent to the power grid side through the aggregator. The tasks assigned by the users and the incentives obtained by the users for completing the scheduling tasks are also distributed to the electricity users in the region through the aggregator. Power grid company is the initiator of the data processing center and scheduling tasks. Due to the emergence of the peak will increase power grid company of power circuit operations costs, even cause damage to the power circuit, so the grid company interacts with the user response by publishing power scheduling task to users. In order to mobilize users to complete the scheduling task, the grid side distributes the corresponding subsidies for the users who complete the scheduling task. The framework is shown in Figure 2. The functions of each entity are summarized as follows.

- Users: The users are the object that participate in the incentive demand response. By responding to the task assigned by the power grid, the users can assist the power grid to complete the scheduling tasks. Meanwhile, the users can also obtain the corresponding price subsidies through their own reputation values and the scheduling tasks completed by users.
- User areas (Aggregators): Each area will firstly collect the electricity consumption data of the users in the area, send data to the server, and accept the scheduling tasks assigned by the power grid company for the area. Then the aggregators will assign the corresponding scheduling tasks to users according to the reputation values and adjustable load of users.
- Grid company: The server of grid company calculates the adjustable load of users after receiving the user electricity consumption data sent by various aggregators. In this paper we assume that electricity consumption directly interacts with users, and the reputation value of each user is stored in the server. By calculating the adjustable load of users and the weight of the reputation values of users, the corresponding scheduling tasks are assigned to users.

3.2 Modeling complex network of user electricity consumption

Because the load control of power grid users in different areas is a complex network system, the current user load control method is not accurate and effective for electricity consumption. Based on the above system framework, we further model the complex network of user load regulation. Since the users in each region have different habits of using electricity, the task allocation for the users in each region is different, even though the users' electricity consumption data at the same time is the same. Therefore, groups of electricity users are typically complex systems. We regard a single user in each region as a node, and each node will be linked to the aggregation server in each region, where the aggregation server is the intermediate node. When the node interacts with the intermediate node, that is, when the user participates in the incentive demand response, the user's reputation value will change with the completion of the scheduling task. Therefore, the weight of the connection between this node and the intermediate node is increased, and the user's reputation value is taken as the weight. Similarly, the aggregation servers in each region and the grid company's servers can be represented by the network relationship described above. For the convenience of the experiment, in this paper we assume that the power grid directly interacts with users in a certain area. In actual application, aggregators are still used to complete the overall implementation of this scheme. The complex network model of user electricity consumption is shown in Figure 3.

3.3 User electricity scheduling scheme

In this section, we show the details of our electricity scheduling scheme for the complex network of user electricity consumption in the model. In our scheme, for users in a certain area, their real-time adjustable loads are calculated based on the users' historical

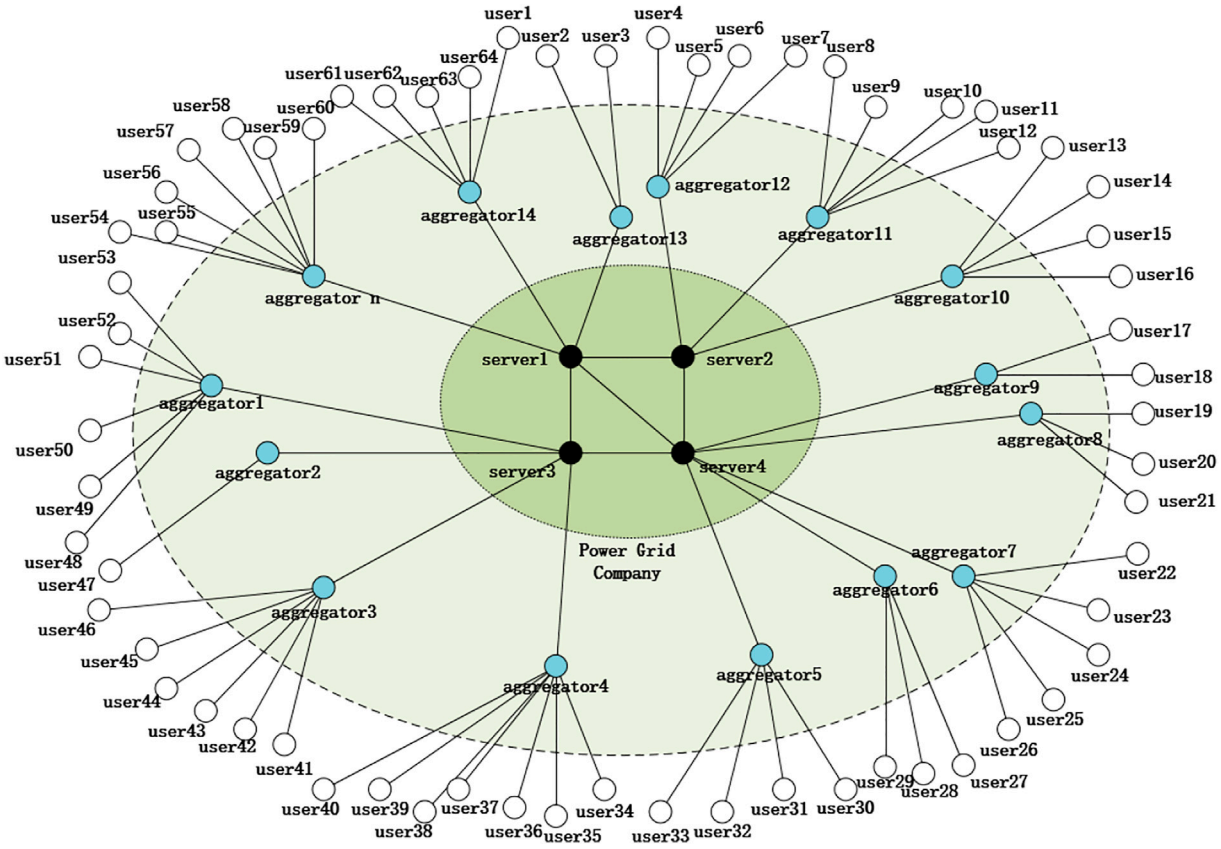


FIGURE 3
Complex network model of user electricity consumption.

electricity consumption data and real-time electricity consumption data. It is convenient for users that they don't need to upload their adjustable load before participating in incentive demand response every time. Assigning corresponding scheduling tasks to users based on the users' reputation values is conducive to improve the reliability of users participating in completing scheduling tasks. While the power grid allocates scheduling tasks to users participating in incentive demand response by combining users' reputation values with real-time adjustable load of users. We firstly outline the five steps of this scheme, and then introduce the implementation details of each step. The process of implementation of this scheme is shown in Figure 4.

- Initialization: The server of grid company receives the data of users. Subsequently, the power grid company adjusts coefficient s according to total load scheduling announced price subsidies;
- Calculation of Reputation Values: The server calculates the reputation values of users based on the number of times that users accept their scheduling tasks and the number of times they complete their scheduling tasks;
- Filtering of Participating Users: According to the users' historical electricity consumption data and real-time electricity consumption data, the users' load elasticity coefficients and adjustable load coefficients are calculated respectively. These coefficients are used as reference for filtering of users;
- Assignment of Tasks: The server sends incentive demand response request to filter users, users can choose whether to participate in the incentive demand response. Then the server assigns the scheduling tasks to the users who finally participate in the incentive demand response according to the users' reputation values and real-time adjustable load;
- Calculation of Price Subsidies: The server assigns corresponding subsidies to users based on the reduced load and the reputation values of users who completed the scheduling tasks.

3.3.1 Initialization

The historical data of electricity consumption of users are stored in the grid company. The server can directly obtain the historical data of users and the situation of users' participation in incentive demand response and completion of scheduling tasks. However, the power grid company does not have the data of the users' adjustable load, so the users need to upload the real-time adjustable load l_{it} to the server. In this paper, we take each week as a cycle. For the convenience of users, we set the users to upload their data once every four cycles. What's more, the power grid company needs to determine the budget for each interaction response. Then the price subsidies coefficient $\boldsymbol{s}$ is calculated according to the total

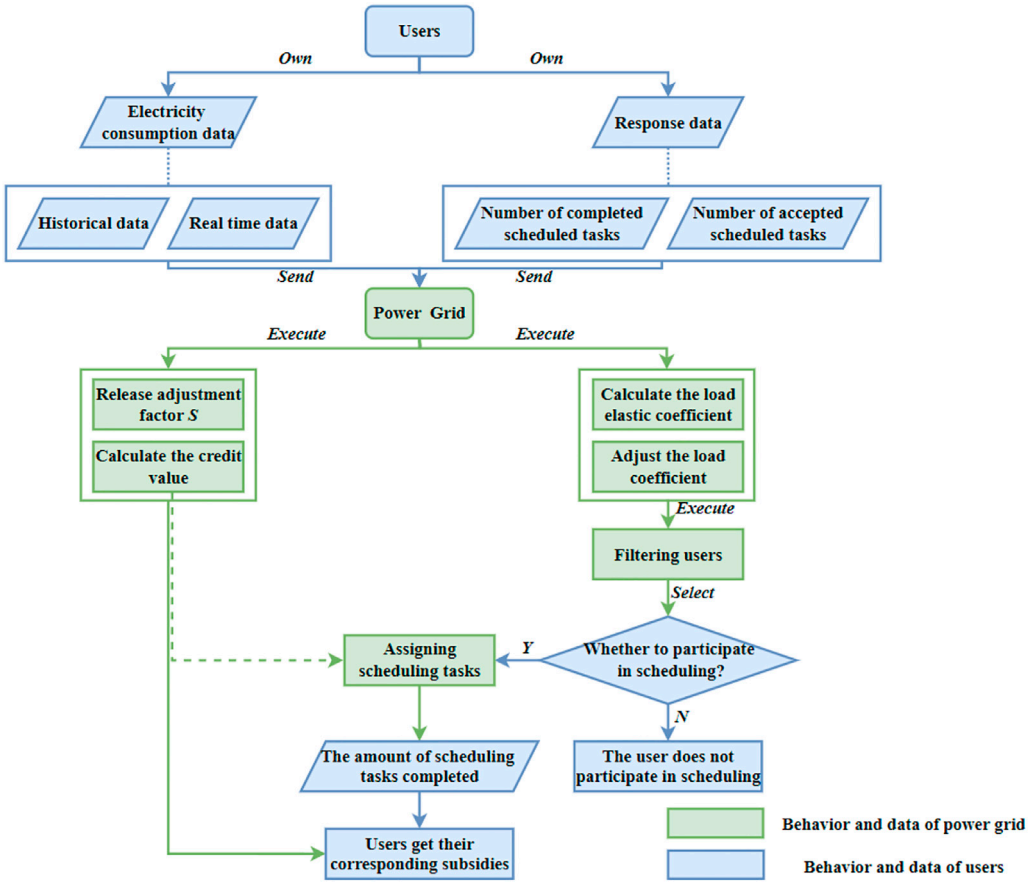


FIGURE 4
The process of user electricity scheduling.

load to be scheduled and the budget for completing the scheduling task:

$$s = \frac{B}{E} \times \delta \tag{1}$$

Where δ is the floating coefficient and $\delta \in (0, 1)$, E is the total load needed to be reduced this time, B is the budget for this scheduling task. The price subsidy coefficient is a reference for the calculation of the users' price subsidies.

3.3.2 Calculation of reputation values

In this section, we propose a method to calculate the reputation values of users. The reputation value is a measurement that reflects the reliability of the users to complete their scheduling tasks. It is calculated by the number of scheduling tasks completed by users and the number of scheduling tasks accepted by users in the scheduling task assigned by the server. Besides, the reputation values of users will be used as the reference for users to obtain the price subsidies after completing their scheduling tasks. However, the reputation value of the users participating in scheduling task in the first time cannot be calculated, so we need to initialize the credit value of the users who have not participated in the scheduling task to ensure the amount of scheduling tasks for users who participate in the scheduling task for the first time. We set the reputation value of the users who participate in the scheduling task for the first time as 0.5.

The calculation process of reputation value is as follows. Firstly, users upload their data to the server, and then the users' reputation values are calculated according to the incentive demand response data of users. The user's reputation value is:

$$R_i = \begin{cases} 0.5, \textit{The user i accepts a scheduled task first time} \\ \gamma \times \dfrac{D_i}{G_i} \end{cases} \tag{2}$$

where γ is a hyper-parameter and $\gamma = 10$, D_i is the number of scheduled tasks completed by the user i, G_i is the number of scheduled tasks accepted by the user i. The user's reputation value is related to two factors, one is the number of scheduling tasks assigned when the user participates in electricity scheduling, the other is the price subsidies the user gets after completing the scheduling task. It is used for the subsequent calculation of the user's reputation level.

3.3.3 Filtering of participating users

In this section, we show how users are assigned related electricity scheduling tasks. Due to some users have no adjustable load at a

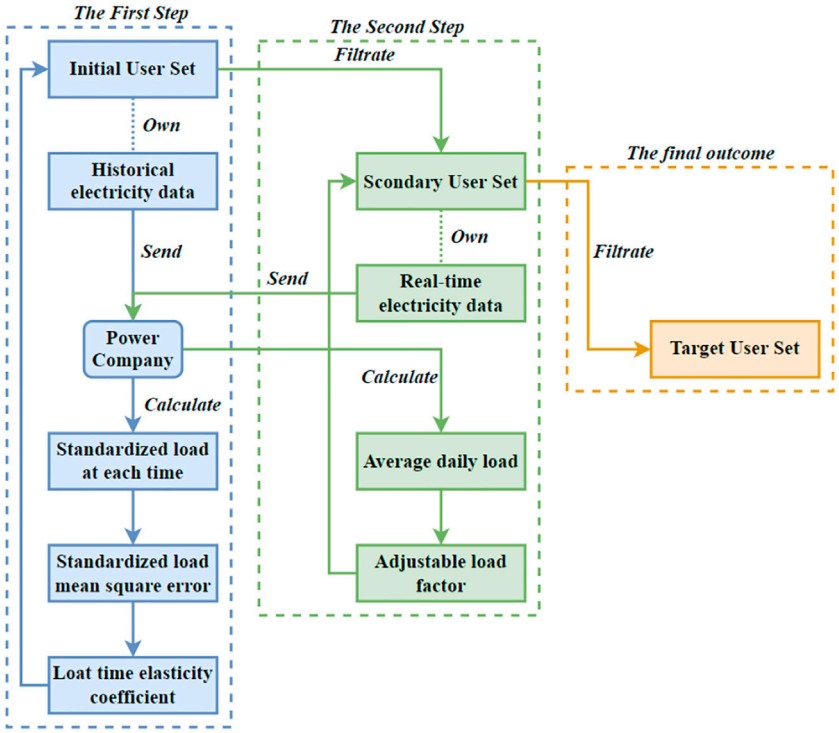


FIGURE 5
Selection frame of participating users.

certain period, users involved in the scheduling task need to be filtered. We define the load used by the user in a certain period as follows: There are 60 min in each hour. We take the maximum load value generated by the power user per minute as the load within 1 minute, and the load within each hour is the average of the load within 60 min during that period. Then based on the historical and real-time electricity consumption data of users, users are selected to participate in scheduling. We use a two-layer scheme to select participating users. Firstly, the load elasticity coefficient is calculated according to the users' historical data, and the users whose load elasticity coefficient meeting the requirements are selected preliminarily. Then, users meeting the requirements are further filtered according to the real-time electricity load of users, and the process of users filtering is shown in Figure 5.

3.3.3.1 The first layer

First of all, the user set meeting the requirements of this scheduling task is preliminarily obtained by processing the users' electricity usage habits and historical data of electricity consumption. In this scheme, each week is divided into 7 days as a cycle. In each cycle, the electricity consumption of user i in each period of 24 h and the maximum electricity load of users in each week are standardized to obtain the standardized load of users in each period:

$$P_{widt} = \frac{N_{widt}}{N_{widm}} \tag{3}$$

Where w represents the statistical period number and it is a natural number greater than or equal to 1. d represents the number of dates within a period and $d = 1,2,3, \ldots, 7$. t represents t periods of day and $t = 0,1,2, \ldots, 23$. N_{widt} is the electricity load of the user i at the t period of the d day in the w cycle. N_{widm} is the maximum daily electricity load of the user i on day d of the w cycle.

In order to analyze the data better, we initially process the data. Firstly, after standardizing the load in each period for user i, the server calculates the mean square deviation of the standardized load in the same period of every day in a cycle. Then the average standardized load of user i in each period is calculated:

$$u_{idt} = \frac{1}{C}\sum_{l=1}^{C} P_{lidt} \tag{4}$$

Where C is the total number of cycles, and C is taken as 5 in the experimental section of this paper. Secondly, the mean square deviation of the standardized load σ_{idt} in each period of the day in a cycle is calculated according to the average standardized load:

$$\sigma_{idt} = \sqrt{\frac{1}{C}\sum_{l=1}^{C}(P_{lidt} - u_{idt})^2} \tag{5}$$

Firstly, the server calculates the mean square deviation of the standardized load of user on a certain day according to the mean square deviation of the standardized load:

$$u_{idw} = \frac{1}{24}\sum_{t=0}^{23}\sigma_{idt} \tag{6}$$

Secondly, according to the mean square deviation of the user's average standardized load, the user's load elastic coefficient on that day is obtained:

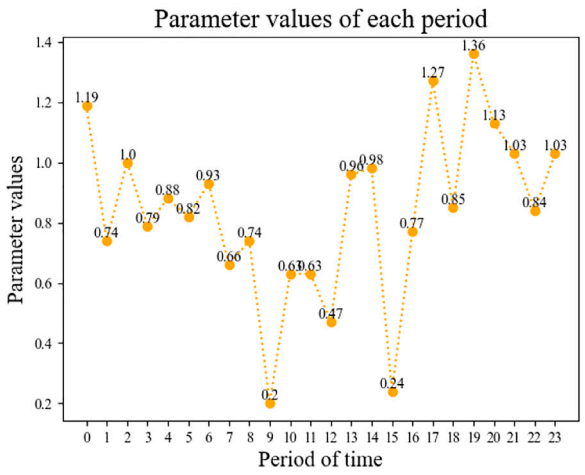


FIGURE 6
Values of H_t in each period.

$$\varphi_{idw} = \sqrt{\frac{1}{24}\sum_{t=0}^{23}(\sigma_{idt} - u_{idw})^2} \tag{7}$$

Then, the server selects the user whose load elastic coefficient is greater than the threshold as the candidate.

3.3.3.2 The second layer

Users need to be filtered more accurately after the first layer of filtering for them. According to the user's real-time electricity load, the real-time adjustable load coefficient is obtained:

$$\tau_{it} = \frac{P_{it} - \overline{P_l}}{P_{im} - \overline{P_l}} \tag{8}$$

where P_{it} is the user's real-time electricity load at time t, $\overline{P_l}$ represents the average electricity load of the user on the day. We take the maximum load used by the user within a certain minute as the load within that minute, and then calculate the average load of the user within an hour according to this method, and then calculate the average load of the user within a day. P_{im} represents the maximum electricity load of the user on the day. According to the above load determination method, we can determine the maximum average load used by users in 1 hour of the day, and use it as P_{im}. If τ_{it} is greater than 0, the user i is selected as a participant in the incentive demand response.

3.3.4 Assignment of tasks

In this section, we describe how to perform the task assignment in detail. After the server selects the users who meet the requirements, it sends to them an invitation message to participate in the incentive demand response. The users can choose whether to participate in the response or not, and then the server assigns tasks to the users who finally participate in the incentive demand response. Reasonable allocation of scheduling tasks is conducive to solving the problem of electricity consumption in the grid company. The details of the task assignment are as follows:

Firstly, users participating in the response are selected by server according to the period when the power grid publishes scheduling tasks, the reputation values of users and the adjustable load coefficients of users. Then the users choose whether to participate in the incentive demand response. The users with a high reputation value are assigned more scheduling tasks. Finally, the user set *su* which includes users who participate in this incentive demand response are obtained, and *su* is $\{u_i|\theta_{it} > 0 \; and \; u_i \; selects \; to \; participate \; in \; the \; scheduling \; task\}$. According to the number of users in the user set *su* and the reputation value of each user, the corresponding scheduling tasks are generated and sent to users.

3.3.4.1 Calculation of users' adjustable load

To select high-quality users, users' adjustable load needs to be ensured. The user's adjustable load is calculated according to the user's real-time adjustable load coefficient and the user's real-time electricity load:

$$P_i = H_t \times \tau_{it} \times P_{it} \tag{9}$$

Where the H_t is the adjustment coefficient when the total electricity load is too high in time period *t*.

3.3.4.2 Determination of parameter H_t

To determine the parameter H_t, we use the machine learning to train a computation model. The first step is to build a training set, which includes the real-time electricity consumption data and the real-time adjustable load of users. We adjusted the division of the data set in the model training, and divided the data set containing 50 users into three sub-data sets: training set, validation set and test set, and the number of users in each sub-data set was 30,10,10 respectively. The adjustable load coefficient can be calculated according to the real-time adjustable load coefficient and the real-time adjustable load of users. Since users upload data every four cycles, the parameters are updated every four cycles to ensure the accuracy of the scheme. Then through the formula (9), each user's H_{it} is calculated to construct a data set $\{(\tau_{it}, l_{it}; p_{it})| H_{it}\}, i \in [0, 50], t \in [0, 23]$. And then we use the training set $\{(\tau_{it}, l_{it}; p_{it})| H_{it}\}, i \in [0, 30], t \in [0, 23]$ to train a linear regression model. The objective function is:

$$J(\theta_{it}) = \underset{\theta_{it}}{minimize} \frac{1}{2m}\sum_{n=1}^{m}\left(h_{\theta_{it}}(x^n) - H_{it}{}^n\right)^2 \tag{10}$$

The optimal parameters of the model are obtained by minimizing the objective function through the gradient descent algorithm. Since the parameters of each period are different, we need to train the models for 24 periods. After that, the real-time electricity load of the user and the period are taken as the input of the model to obtain the parameters H_{it}. After the parameters are trained according to the data of each user, the average value of the parameters of each user in the same period is taken as the final parameter value H_t.

3.3.4.3 Assignment of tasks

After the server calculates the users' adjustable load, it assigns the corresponding tasks to users according to the users' reputation values. But the scheduling task should not exceed the range of the user's

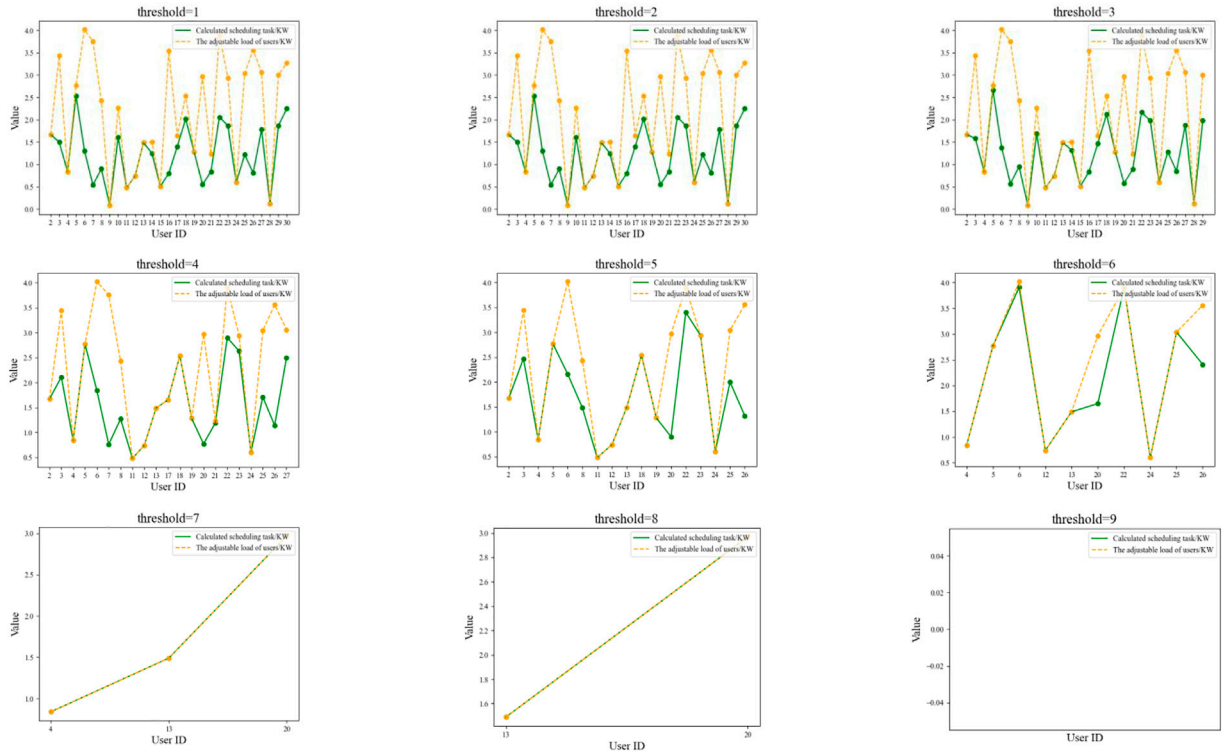


FIGURE 7
Comparison between scheduling tasks and adjustable load of users with different load elastic coefficient threshold (*threshold* $\in [1, 9]$).

adjustable load. Otherwise, most users may fail to complete their scheduling tasks. The following is a detailed description of our task allocation scheme. First of all, according to the users' reputation values, the server calculates the reputation weight of all users in the *su*:

$$W_i = \frac{R_i}{\sum_{l=1}^{M} R_l} \tag{11}$$

Where M is the total number of user and $M = |su|$. When the reputation value of the user is 0, that is, when the reputation weight of the user is 0, the minimum scheduling task is assigned to the user according to the total load of the scheduling task:

$$P_0 = \frac{E}{\sum_{l=1}^{M} R_l}, R_l \in [1, 10] \tag{12}$$

If P_0 is greater than P_i, the user's scheduling task is P_i. When the user's reputation value is greater than 0, the scheduling task of the user can be calculated as follows according to the user's reputation weight:

$$P_{ib} = \left(W_i - \frac{M \times W_i - 1}{\sum_{l=1}^{M} R_l} \right) \times E \tag{13}$$

If P_{ib} is greater than P_i, the user's scheduling task is P_i. Above all, the scheduling task of user i is as follows:

$$P_{id} = \begin{cases} P_{ib}, R_i \neq 0 \text{ and } P_i > P_{ib} \\ P_0, R_i = 0 \text{ and } P_i > P_0 \\ P_i, \qquad Others \end{cases} \tag{14}$$

3.3.5 Calculation of price subsidies

In this section, we propose a method to calculate the incentive. In order to mobilize the enthusiasm of users to participate in incentive demand response, we need to give certain subsidies to users who have completed scheduling tasks. In this way, users will be more willing to receive scheduling tasks, and help the grid company to achieve peak load shifting and other scheduling tasks. In our scheme, users' corresponding subsidies are generated according to their reputation values and the values of the scheduled load completed by users. The incentive obtained by the user is directly proportional to the user's reputation value and the scheduling task completed by the user. If users receive and complete the scheduling task, their reputation values will increase. If users receive a task, but they don't complete it, then their reputation values will decrease. If users don't receive the scheduling task, their reputation values remain unchanged. The users' original reputation value before updating are used as the basis to calculate the subsidies for the completion of the scheduling task. We divide the reputation value levels into ten levels, from 1 to 10 respectively. The users' reputation ratings are rounded up to their reputation values, denoted as $\lceil R_i \rceil$, where $R_i \in [1, 10]$. If the user's reputation values are greater than the threshold value r after participating in the incentive demand response, the user's price subsidies (incentive) can be calculated by the users' own reputation values and the schedule load of the users participating in the schedule:

$$S_i = \lceil R_i \rceil \times s \times P_{id}, \lceil R_i \rceil > r \tag{15}$$

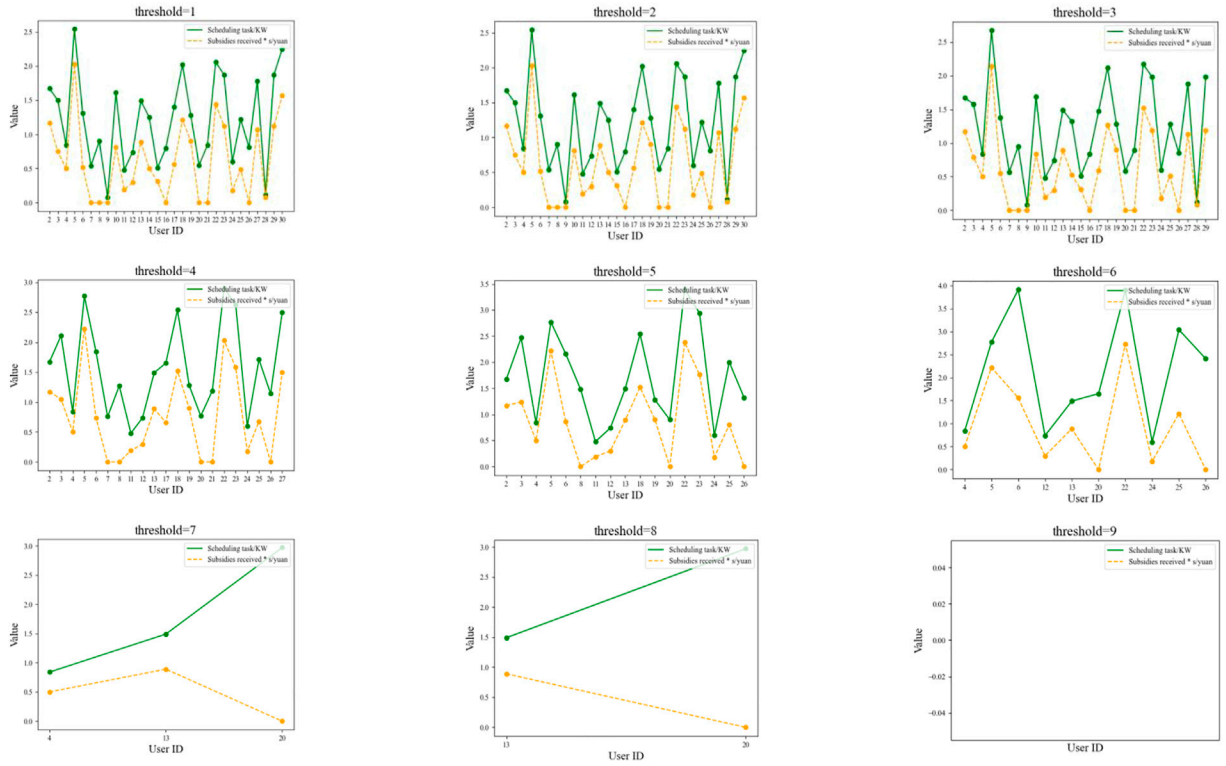


FIGURE 8
Changes of users' scheduling tasks and subsidies with different load elastic coefficient threshold (*threshold* $\in [1, 9]$).

where r is the threshold of the price subsidies users can get after completing the scheduling tasks, and s is the price subsidy coefficient used by the power grid company to adjust the subsidies to the users. Users are more active in participating in demand response and completing scheduling tasks by providing incentives to users in the form of price subsidies. When the users pay the electricity bills, the corresponding amount can be offset with the price subsidies obtained by participating in the incentive demand response.

4 Experiment

In this section, the experimental process and results are introduced in detail. The experimental part of this paper is described below. Due to the difficulty in obtaining power grid data and the innovation of the method adopted in this paper, the data set adopted in this experiment is generated based on our daily consumption habits. And then we reasonably generate the data of electricity load of 50 users for 24 h a day within 5 weeks, as well as the times of each user accepting and completing scheduling tasks, so as to obtain the reputation values of each user. In this paper, every week is taken as a cycle, and the electricity consumption of users is different every day in each cycle. Therefore, in the experiment, a certain period from Monday to Sunday should be selected as the target regulation period, and the total regulation load should be input. Then, we calculate the parameter H_t by machine learning method, and show the influence of load elasticity coefficient threshold and reputation threshold on scheduling task allocation and the incentives users can get. In the end, this experiment proves the effectiveness of this scheme by simulating the assignment of tasks and the allocation of price subsidies to users.

4.1 Calculation of parameter H_t

In this section, we show the calculation results of the parameter. The server uses the electricity data of every moment in one specified day provided by users and adjustable load for the users each moment to combine the adjustable load coefficients calculated at this moment. And the server takes the users' adjustable load coefficients, real-time electricity loads, real-time adjustable load and parameter H_{it} calculated by each user data to constitute training set $\{(\theta_{it}, l_{it}, p_{it}) \mid H_{it}\}$, $t \in [0, 23]$. Since the simulation data of 50 users are used in this experiment, the data of 30 users are taken as the training data. After the parameters are trained according to the data of each user, the average value of the parameters of each user in the same period is calculated as the final parameter value H_t. Figure 6 shows the value of H_t for each period.

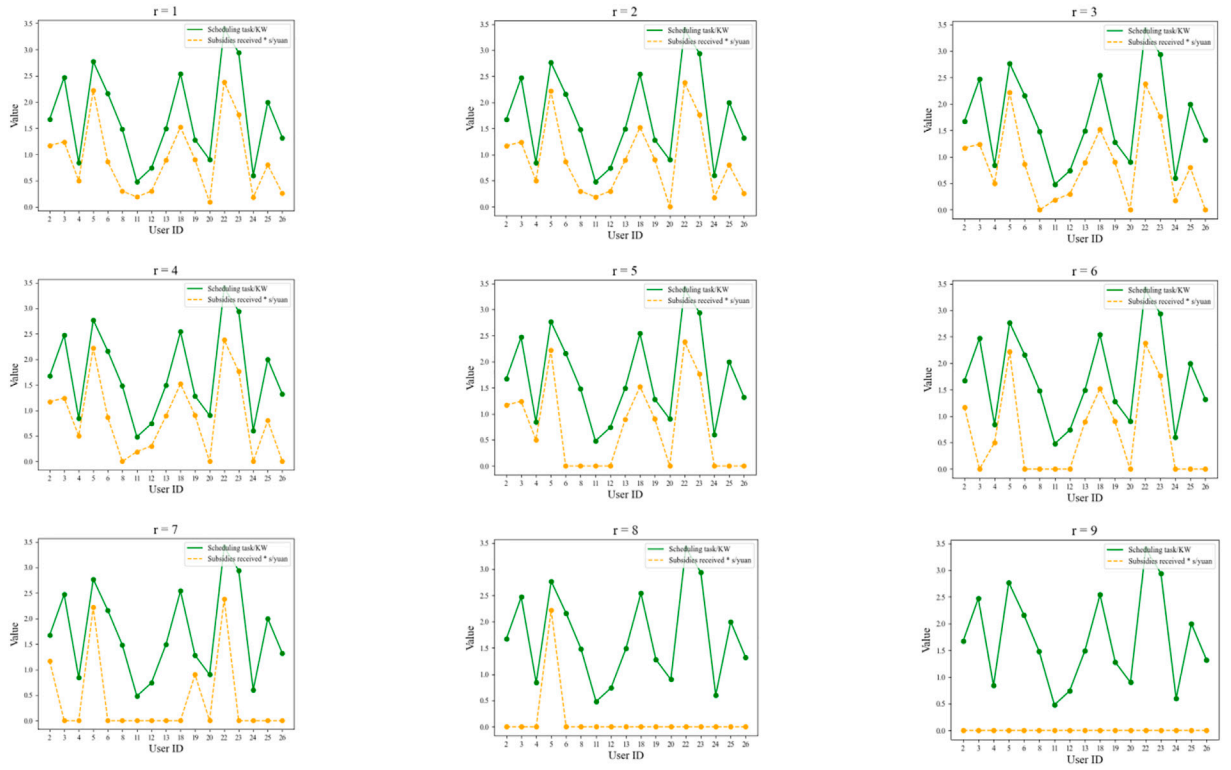


FIGURE 9
Differences of subsidies obtained by users under different reputation thresholds ($r \in [1, 9]$).

4.2 Assignment of scheduling tasks

In this section, we show the influence of load elasticity coefficient threshold and reputation threshold on scheduling task allocation and the incentives users can get. The assignment of users' scheduling tasks is a relatively important link in this scheme. The price subsidies obtained by users participating in incentive demand response are not only related to their own electricity consumption data, but also related to some parameters selected at the grid company side, such as load elastic coefficient threshold and reputation value threshold. The load elasticity coefficient threshold and reputation value threshold are determined according to the range of load elasticity coefficient and reputation value. Because the load elasticity coefficient is small, the load elasticity coefficient threshold is the amplified value for the sake of observation. In practical application, the most appropriate threshold is selected by the grid end to screen users and assign scheduling tasks.

4.2.1 Influence of load elastic coefficient threshold

In this section, we discuss the impact of the load elastic coefficient threshold on the scheme performance. After the daily load elastic coefficient within a cycle is calculated according to the historical electricity consumption data of the user, the daily load elastic coefficient of the user compares with the threshold value. The users, whose load elastic coefficient is larger than the threshold value, are preliminarily selected. The selection of different load elastic coefficients will affect the selection of users participating in scheduling tasks. Thus, different scheduling tasks are assigned to users, and the price subsidies users get after completing the scheduling tasks are different. The adjustable load coefficient of the user is obtained through multiple calculations based on the historical electricity consumption data and real-time electricity consumption data of each user. The parameters H_t are obtained through a linear regression model, and then the adjustable load of the user is calculated based on the real-time electricity consumption data of the user and parameters H_t. There is a certain difference between the users' adjustable load and the tasks assigned to the users by the power grid according to the users' reputation values. Figure 7 shows the difference between the calculated scheduling tasks that should be assigned and the users' adjustable load when the load elastic coefficient threshold is different.

As can be seen from Figure 7, the load elastic coefficient threshold should not be too large or too small. If the threshold of load elasticity coefficient is too small, many low-quality users will also be selected. Although more users are selected to participate in scheduling tasks, the quality of users participating in scheduling tasks is not high, which will affect the completion of the overall tasks of power grid scheduling. When the load elastic coefficient threshold is too large, users participating in scheduling tasks are over-filtered. When the load elastic coefficient threshold is equal to 9, no users even meet the requirement to participate in scheduling tasks. In the selection of the final scheduling tasks of users, if the values of the calculated scheduling task assigned to the user is smaller than the adjustable load of the user, the calculated scheduling

tasks are selected as the final scheduling tasks. If not, the user's adjustable load is selected as the final scheduling task. The load elastic coefficient threshold will also affect the subsidies obtained by users when they complete scheduling tasks by affecting the assignment of their scheduling task. Figure 8 shows the change of subsidies obtained by users when the load elastic coefficient threshold is different.

As we can see from Figure 8, the number of users participating in scheduling tasks changes with the change of the load elastic coefficient threshold. When the load elastic coefficient threshold is small, the number of users participating in scheduling tasks is large, and the average scheduling tasks assigned to users are small, so the subsidies obtained by users also decrease. In this case, although there are more users participating in scheduling tasks, there are more users who are not highly likely to complete scheduling tasks. Because the load elastic coefficient is small while the time limit of participating in scheduling tasks is large, then the complete degree of overall scheduling tasks may be low. As the load elastic coefficient threshold increases, the number of users participating in scheduling tasks also decreases. Accordingly, users participating in scheduling tasks are assigned more scheduling tasks and get more price subsidies. However, due to the limitation of users' adjustable load, the scheduling tasks that users can complete will not exceed their adjustable load. Therefore, too few people participating in the overall scheduling task will lead to a low completion degree of the overall scheduling task. Therefore, the threshold of load elastic coefficient should not be too small or too large.

4.2.2 Influence of reputation threshold

In this section, we discuss the impact of the reputation threshold on the scheme performance. Firstly, the server calculates the weight of the users participating in this scheduling task among all participating users based on their reputation values. And the server considers them as the weight of the scheduling tasks assigned to users in the total scheduling tasks. Then, the server compares the user's adjustable load with the scheduling tasks assigned to the user. If the scheduling task of the user is smaller than the adjustable load, the scheduling task is taken as the final scheduling value. Otherwise, the calculated adjustable load of the user is taken as the final scheduling value. After completing the scheduling task, users will get the corresponding subsidies according to their reputation levels and the size of the scheduling tasks completed by users, but the users whose reputation value is less than the reputation value threshold will not get the price subsidies. The difference of the reputation value threshold will affect the users who participate in the incentive demand response to get their subsidies. Figure 9 shows the differences of the users' subsidies under different reputation value thresholds.

As can be seen from Figure 9, when the reputation threshold increases, the number of users who can get price subsidies decreases. When the reputation threshold reaches nine, no user can get price subsidies. If the reputation threshold is too large, it will reduce the enthusiasm of users to participate in scheduling tasks. If the reputation threshold is too small, it will increase the cost of the power grid. Because users with small reputation value are less likely to complete scheduling tasks, it is necessary to set an appropriate reputation threshold, and users with low reputation can get subsidies after reaching the threshold. Then the enthusiasm of low-reputation users to complete their scheduling tasks can be improved.

5 Conclusion

To solve the issues of user electricity consumption in complex network, we propose a reputation-based electricity scheduling scheme for complex network of user electricity consumption in this paper. We first construct a complex network model of the user's electricity consumption. Then we construct a reputation calculation method for electricity users, and then we use the machine learning method to train a computation model to calculate the adjustment coefficients of electricity load, which is the basis for adjusting the electricity scheduling tasks. Further, the corresponding electricity scheduling tasks are assigned to the selected electricity users respectively for realize electricity peak load shifting when the total electricity load is too high. Finally, the incentives of users who participate in the incentive demand response of electricity scheduling are calculated according to the calculated reputation and the completed scheduling tasks of users. Experiment results show our scheme can effectively calculate the reputation values of users based on their historical data, and the corresponding electricity scheduling tasks are effectively and efficiently assigned to related users according to the users' reputation values and the real-time adjustable load.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

WT, XL, YZ, and MZ contributed to conception and design of the study. ZW organized the database. YZ and JW performed the statistical analysis. WT wrote the first draft of the manuscript. XL, YZ, MZ, and JW wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Conflict of interest

WT, YZ, ZW, and JW were employed by Shenzhen Power Supply Bureau Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Zheng R, Li Z, Tang Y, Ni C-Y, Li G, Han B. Incentive demand response model and evaluation considering the uncertainty of resident user participation [J]. *Automation Electric Power Syst* (2022) 46(08):154–62. doi:10.7500/AEPS20210404001

2. Zhenbo W, Zhang H, Pingeucalyptus W, Zheng L, Ma X, Sun Z. Considering dynamic incentive model two phase micro power grid operation optimization of demand response [J]. *Power Syst Prot Control* (2021) 49(19):1–10. doi:10.19783/j.carolcarrollnkiPSPC.201605

3. Nasiruzzaman ABM, Pota HR. Transient stability assessment of smart power system using complex networks framework. In: 2011 IEEE Power and Energy Society General Meeting; 24-28 July 2011; Detroit, MI, USA (2011). p. 1–7. doi:10.1109/PES.2011.6038970

4. Liu B, Li Z, Chen X, Huang Y, Liu X. Recognition and vulnerability analysis of key nodes in power grid based on complex network centrality. *IEEE Trans Circuits Syst Express Briefs* (2018) 65(3):346–50. doi:10.1109/TCSII.2017.2705482

5. Wang Z, Hill DJ, Chen G, Dong ZY. Power system cascading risk assessment based on complex network theory. *Physica A: Stat Mech its Appl* (2017) 482:532–43. doi:10.1016/j.physa.2017.04.031

6. Wu Y, Liu J, Li F, Yan Z, Zhang L. Network model of bilateral power markets based on complex networks. *Int J Mod Phys B* (2014) 28(22):1450144. doi:10.1142/s0217979214501446

7. Papadopoulos V, Knockaert J, Develder C, Desmet J. Peak shaving through battery storage for low-voltage enterprises with peak demand pricing. *Energies* (2020) 13:1183. doi:10.3390/en13051183

8. Uddin M, Romlie MF, Abdullah MF, Tan C, Shafiullah GM, Bakar AHA. A novel peak shaving algorithm for islanded microgrid using battery energy storage system. *Energy* (2020) 196:117084. ISSN0360-5442. doi:10.1016/j.energy.2020.11784

9. Guo W, Liu D, Liu M, Zhang H, Wang G, Fan B, et al. Demand-side resource response characteristics and peak shaving scheduling response model. *EnergyReports,* (2022) 8(6):586–93. doi:10.1016/j.egyr.2022.03.149

10. He S, Gao H, Liu J, Zhang X, Chen Z. Distribution system planning considering peak shaving of energy station. *Appl Energ* (2022) 312:118692. ISSN 0306-2619. doi:10.1016/j.apenergy.2022.118692

11. Rana MM, Romlie MF, Abdullah MF, Uddin M, Sarkar MR. A novel peak load shaving algorithm for isolated microgrid using hybrid PV-BESS system. *Energy* (2021) 234:121157. ISSN0360-5442. doi:10.1016/j.energy.2021.121157

12. Jin H, Qin L, Qian C, Wang Y, Wang L. *Discussion on demand side response technology in smart grid [J].* Beijing,China: Electricity Technology Edition (2011). 17–22.

13. Dong C, Ng CT, Cheng TCE. Electricity time-of-use tariff with stochastic demand. *Prod Oper Manag* (2017) 26:64–79. doi:10.1111/poms.12610

14. Yang H, Wang L, Zhang Y, Tai HM, Ma Y, Zhou M. Reliability evaluation of power system considering time of use electricity pricing. *IEEE Trans Power Syst* (2019) 34(3):1991–2002. doi:10.1109/TPWRS.2018.2879953

15. Yang P, Tang G, Nehorai A. A game-theoretic approach for optimal time-of-use electricity pricing. *IEEE Trans Power Syst* (2013) 28(2):884–92. doi:10.1109/TPWRS.2012.2207134

16. Lin J, Sheng M, Wang L, Li Y, Zeng M, Zhang X. Research on incentive subsidy mechanism of demand response based on system dynamics. In: 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2); 22-24 October 2021; Taiyuan, China (2021). p. 2027–31. doi:10.1109/EI252483.2021.9712902

17. Wu Y, Li H. Additive model for user electricity consumption behavior analysis. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC); 20-22 December 2019; Chengdu, China (2019). p. 1703–7. doi:10.1109/IAEAC47372.2019.8997882

18. Wang Y, Chen Z, Xu Z, Gang G, Lu J. User electricity consumption pattern optimal clustering method for smart gird. In: 2018 14th IEEE International Conference on Signal Processing (ICSP); 12-16 August 2018; Beijing, China (2018). p. 567–70. doi:10.1109/ICSP.2018.8652346

19. Zhao Z, Wang J, Liu Y. User electricity behavior analysis based on K-means plus clustering algorithm. In: 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC); 19-21 December 2017; Dalian, China (2017). p. 484–7. doi:10.1109/ICCTEC.2017.00111

20. Cheng Q, Min C, Ciwei G, Huixing L, Tugang S. Research on the analysis of user's electricity behavior and the application of demand response based on global energy interconnection. In: 2016 China International Conference on Electricity Distribution (CICED); 10-13 August 2016; Xi'an, China (2016). p. 1–7. doi:10.1109/CICED.2016.7576206

21. Chen H, Xiong N, Zhang H, Wang W, Chen J. Demand response potential measurement based on adjustable load scale. In: 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2); 22-24 October 2021; Taiyuan, China (2021). 831–4. doi:10.1109/EI252483.2021.9713011

22. Xu Q, Ding Y, Zheng A. An optimal dispatch model of wind-integrated power system considering demand response and reliability. *Sustainability* (2017) 9:758. doi:10.3390/su9050758

23. Fu X, Zhou H, Jiang QJ, Fan K. Research on demand side response system of electricity price under electricity market incentive mechanism[J]. *Distributed Generation Altern Energ J* (2021) 36:23–42. doi:10.13052/dgaej2156-3306.3612

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 09 May 2023
DOI 10.3389/fphy.2023.1196294

OPEN ACCESS

EDITED BY
Fei Yu,
Changsha University of Science and Technology, China

REVIEWED BY
Dongfan Chao,
Hainan University, China
Xiong Li,
East China Jiaotong University, China

*CORRESPONDENCE
Lei Chen,
chenlei@hnust.edu.cn

RECEIVED 29 March 2023
ACCEPTED 24 April 2023
PUBLISHED 09 May 2023

CITATION
Lei Y, Chen L, Li Y, Xiao R and Liu Z (2023), Robust and fast representation learning for heterogeneous information networks. *Front. Phys.* 11:1196294. doi: 10.3389/fphy.2023.1196294

Robust and fast representation learning for heterogeneous information networks

Yong Lei[1,2], Lei Chen[3]*, Yuan Li[3], Ruifeng Xiao[4] and Zhaohua Liu[3]

[1]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China, [2]Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China, [3]School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China, [4]School of Computer Science and Railway Engineering, Hunan Technical College of Railway High-speed, Hengyang, China

Network representation learning is an important tool that can be used to optimize the speed and performance of downstream analysis tasks by extracting latent features of heterogeneous networks. However, in the face of new challenges of increasing network size, diverse latent features, and unseen network noise, existing representation models need to be further optimized. In this paper, a robust and fast representation learning model is proposed for heterogeneous networks, called RFRL. First, the global features of a heterogeneous network are divided into multiple intra-type local features and inter-type local features, and a type-aware biased sampling is designed to generate training samples for each local feature. Second, a node-type-aware and a link-type-aware shallow representation strategy are used to learn intra-type features and inter-type features respectively. This enables the model to achieve good performance while having high speed through the divide-and-conquer learning process and shallow learning model, thus coping with increasing network size and latent feature diversity. Finally, adversarial learning is used to integrate the above two representation strategies to address unseen network noise and enhance the robustness of representation learning. Extensive experiments on three network analysis tasks and three public datasets demonstrate the good performance of our RFRL model.

KEYWORDS

heterogeneous information network (HIN), robust representation learning, adversarial learning, intra-type feature, inter-type feature

1 Introduction

In the real world, many systems (such as traffic systems and social systems) can be abstracted into heterogeneous information networks (HINs) with different node types and link types [1]. However, as HINs grow in size, complex coupled network data cannot cope with the real-time demands of downstream network analysis tasks [2]. For this reason, heterogeneous network representation learning has been proposed and is developing rapidly [3]. Heterogeneous network representation learning is the process of converting high-dimensional complex HINs into low-dimensional simple discrete vectors that retain as much of the underlying features of the network as possible [4]. After heterogeneous network representation learning, the resulting low-dimensional vectors can be used as feature input for downstream network analysis tasks to improve speed and performance [5].

Heterogeneous network representation learning has proved to be very useful in many downstream tasks [6] such as link prediction, node classification, node clustering, etc.

Depending on the depth of the model structure, existing heterogeneous network representation models can be simply classified into shallow and deep models [7]. Shallow models learn the feature representation of nodes by neural networks with fewer layers [8, 9]. In the metapath2vec [10] model, a meta-path guided walk strategy is used to sample node sequences. These node sequences are then used to generate node embedding via a word vector model skip-gram, which maps the feature information of HINs into low-dimensional vectors. The ASPEM [11] model proposes a multi-aspect-based approach to capture the semantic information of HINs by learning low-dimensional discrete vectors of nodes in multiple semantic spaces. Shallow models have fewer model layers and parameters and are therefore faster to train and rarely suffer from overfitting problems [12]. Deep models generally extract feature information in HINs through multiple nonlinear transformations between multiple hidden layers [13]. The HAN model first splits the graph into multiple sub-graphs with the same type guided by meta-paths, and then uses a node-level attention strategy to learn the features of nodes in each sub-graph. Finally, these sub-graphs with different types are mapped into the same feature space by the semantic-level attention strategy. In HetGNN [14], both long short-term memory (LSTM) and Multilayer Perceptron (MLP [15]) are used to extract and understand the feature information of HINs and convert it into low-dimensional discrete vectors. With more complex structures, deeper models can learn higher dimensional features and be applied to more tasks such as image processing, speech recognition, etc. In summary, shallow models are faster than deep models but perform worse.

In the new era of big data, the scale of HINs is getting larger and larger, the network features are getting more and more complex and diverse, and the network noise is getting more and more numerous [16]. Facing the characteristics of HINs in the new era, there are three further challenges to existing models.

- *Speed and performance are difficult to balance.* With the rapid development of IoT [17] and cloud computing technologies, the size of HINs is increasing [18]. Meanwhile, downstream network analysis tasks are increasingly time-sensitive. Existing deep models perform well, but their timeliness cannot meet the demands of the big data era. Existing shallow models have good speed and scalability, but they struggle to accurately capture network features and have yet to improve their performance.
- *Accurate network features are difficult to extract.* In the era of big data, the number of node types and link types in a HIN is also increasing, and the latent features it represents are becoming increasingly complex and diverse. Moreover, multiple features are increasingly coupled and mixed. As a result, feature extraction becomes increasingly difficult. Most traditional models try to extract all features directly and fail to separate multiple features better, thus facing the dilemma of insufficient feature extraction accuracy.
- *The effect of network noise is neglected.* There is often some noise in HINs, such that some nodes are lost and some nonexistent links are constructed. These noises can cause local features of nodes or links to be lost or inaccurate. During model training, this noise increases the generalization error of the model and degrades its performance when dealing with unseen data. Overall, this noise can make the learned heterogeneous network representation vectors less accurate, which can affect the performance of downstream tasks.

To solve the above problems, the motivation of this paper is to design a heterogeneous network representation model that can adapt to network noise with high speed and performance. For one thing, to maintain the high-speed of representation learning, shallow models rather than deep models are chosen to cope with large-scale HINs. And for another, to maintain the performance of representation learning, the features of the heterogeneous network are further decomposed into intra-type and inter-type features. Intra-type features refer to the proximity of multiple nodes under the same node type. Inter-type features refer to the semantic similarity of nodes between two different node types. The accuracy of feature extraction is further improved by converting the original one learning process of global features into multiple learning processes of different sub-features. And thirdly, to attenuate the effect of network noise, generative adversarial networks (GANs [19]) are introduced into the representation learning process to enhance the generalization ability. Based on the above ideas, we propose a robust and fast representation learning model for HINs, called RFRL. The main contributions of this paper are as follows.

- A type-aware bias sampling strategy is proposed to treat each node type and each link type as independent subspaces, and generate both intra-type training samples for each node type and inter-type training samples for each link type using a random walk strategy.
- A node type-aware adversarial learning strategy is designed to learn intra-type features in each node type space using a shallow network, and generate more unseen samples using GAN to enhance the robustness of feature extraction and attenuate the effects of noise.
- A link-type-aware adversarial learning strategy is designed to learn inter-type features in each link type space using another one shallow network, and also to enhance the robustness and generalization of feature extraction using adversarial learning as well.
- The RFRL model is designed to achieve a balance between speed and performance by combining the above strategies. Extensive experiments on three analysis tasks and three public datasets demonstrate the excellent performance of our RFRL model.

The rest of the paper is organized as follows. Related work and definitions are presented in Section 2 and Section 3, respectively. Section 4 shows our RFRL model in detail. The experimental analysis is described in Section 5. Finally, Section 6 concludes the paper.

2 Related work

From the technical perspective, existing models or methods can be simply divided into two categories:

(1) Shallow model-based algorithms. The first shallow representation models are random walk-based models. These models first use a random walk strategy to obtain training samples with both intra-type and inter-type features, and then use a shallow skip-gram model to learn both features simultaneously [10, 20–22]. For example, in metapath2vec [10], the setting of meta-paths guides the model to sample intra-type features and inter-type features. The Spacey [20] model proposes a meta-path-based random walk method for heterogeneous personalized space to collect samples on a meta-graph collapsed from a given meta-path. The HHNE [21] model uses meta-path guided random walk to generate heterogeneous neighborhoods for each node to obtain intra-type features and inter-type features. The MARU [22] model uses a meta-context-aware skip-gram based model to learn dynamic meta-contextual relationships to collect samples. Such algorithms have high speed and performance, but most rely on supervised information given by external experts to guide the learning patterns of intra-type features and inter-type features. The second shallow representation models are decomposition-based models. These models decompose the original network into multiple subnetworks and perform shallow learning for each sub-network [23–25]. For example, the EOE [23] model incorporates a harmonious embedding matrix to further embed the embedding that only encode intra-network links. In RHINE [24], pairs of network links are used to distinguish relations into affiliation relations (ARs) and point-to-point structured interaction relations (IRs) to capture the unique structure of the relations. The MIFHNE [25] method models structural proximity, attribute information, and label information in the framework of non-negative matrix decomposition (NMF). The PME [26] model propose to build object and relation embedding in separate object space and relation spaces. Such algorithms do not rely on external supervised information and are fast in time. However, the integration and fusion of multiple subgraph features is difficult, resulting in the performance of this type of algorithm being weak and stable.

(2) Deep model-based algorithms. To better capture intra-type features and inter-type features, multiple deep models are used to enhance the feature learning capability of the models [27]. For example, in MAGNN [28], inner and outer aggregation of meta-paths are designed to collect samples containing inter-type features and intra-type features. The HAN [29] model proposes a node-level attention mechanism and semantic-level attention mechanism to learn intra-type features and inter-type features of HINs, respectively. Both models are designed to consider intra-type and inter-type features of HINs, but both rely on the setting of meta-paths. The HetSANN [34] model and the HGT model take the same type of node as the center and calculate the importance of other types of nodes around it. These two methods can capture the interactions between different types of nodes well, but do not do specialized learning of intra-type node features. The HetGNN [14] model uses a restarted random walk strategy instead of meta-path-based walk strategy, using multiple artificial neural networks to learn the attributes and structures of the nodes, respectively. The model considers and explicitly uses both artificial neural networks to learn intra-type features and inter-type features. However, due to the complexity of the model structure, the training learning of the model is slow and the generalization ability is not strong [30]. The MV-ACM [31] model is a GAN-based model of multiple views divided by link relations, using a game of generators and discriminators to robustly learn the relations between views.

In summary, shallow models have high speed but relatively weak learning ability for network features; deep models can better capture the nonlinear features of complex networks, but their time complexity is higher. Moreover, existing models rarely consider the effect of noise in the network. With the rapid development of new technologies (such as IoT and cloud computing), HINs are getting more large, heterogeneous, and noisy, and their features are getting more complex. The existing models need to be further improved in the face of new features of HINs. To this end, this paper tries to decompose complex and diverse feature learning into intra-type feature learning for node-type subspaces and inter-type feature learning for link-type subspaces to reduce the learning difficulty and enhance the learning accuracy. Meanwhile, in this paper, we try to design a novel shallow model that guarantees the speed and performance of learning. To reduce the effect of network noise, adversarial learning is incorporated into the shallow model to generate more unseen training samples using adversarial learning, which results in more generalized and robust network features.

3 Definition

In this section, several definitions in this paper are first introduced.

Definition 1: Heterogeneous Information Networks (HINs). The heterogeneous network $G(N, E, T, R)$ is composed of a node set $N=\{n_1, n_2, \ldots, n_n\}$ with node type $T=\{T_1, T_2, \ldots, T_n\}$. and a link set $E=\{e_1, e_2, \ldots, e_n\}$ with link type $R=\{R_1, R_2, \ldots, R_n\}$. The mapping relations between node types T with nodes V and between link types R with links E are φ. Specifically, if nodes v_i and v_j belong to the same node type T_1, then there exists $\varphi(v_i)=\varphi(v_j)=T_1$. The links e_i and e_j belong to the same link type R_1, then there exists $\varphi(e_i)=\varphi(e_j)=R_1$.

Definition 2: Heterogeneous Network Representation Learning. Heterogeneous network representation learning is the process of mapping node v_i to low-dimensional vectors $x_i \in R^{1\times d}$ by learning from a HIN, that is $f(v_i)\rightarrow x_i \in R^{1\times d}$. A feature matrix $X \in R^{|V|\times d}$ is formed with the low-dimensional vectors of all nodes, where $|V|$ is the number of nodes. The feature matrix X of nodes can be used in the analysis of downstream tasks of the network.

4 The proposed model

4.1 Overview

With the advent of the era of big data, complex systems in the real world are getting larger and noisier, and their internal heterogeneity is getting stronger. That is to say, in the network,

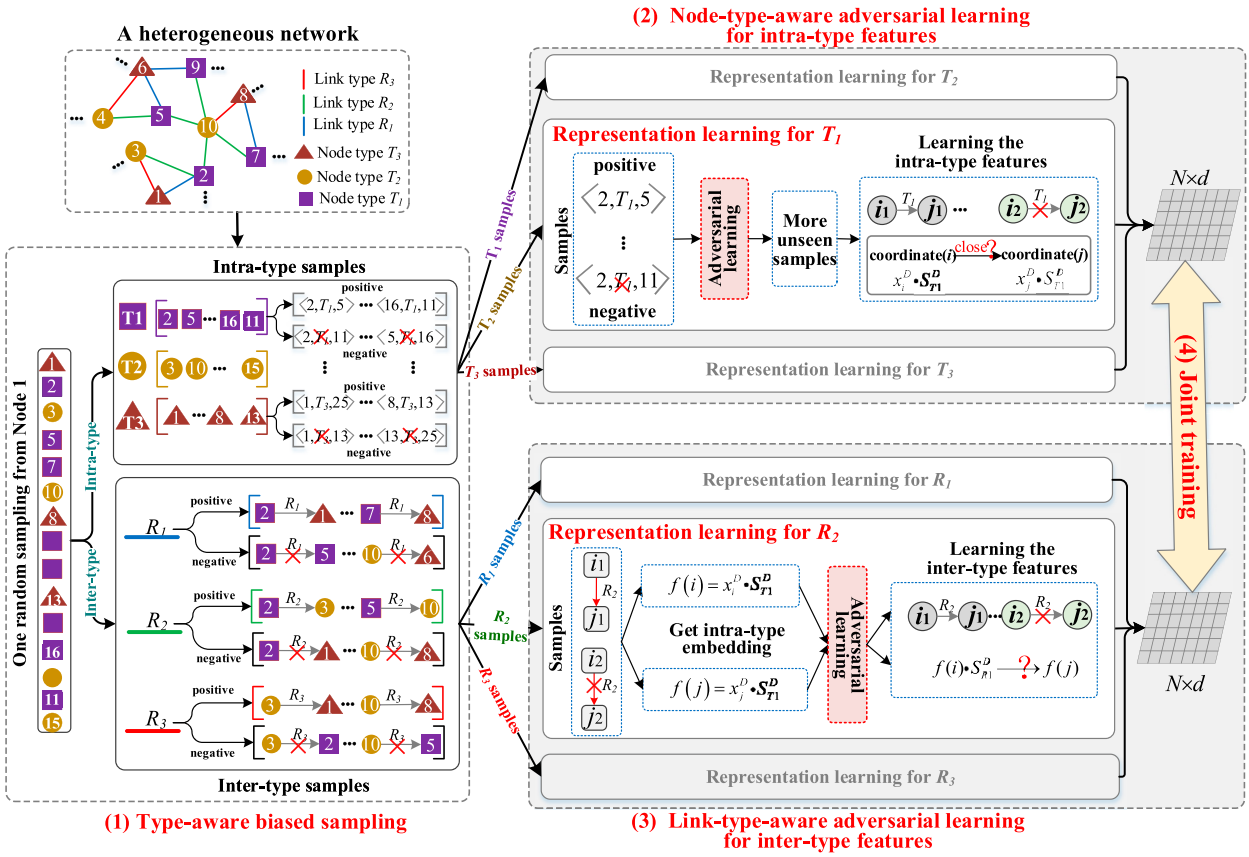


FIGURE 1
The overview of the RFRL model.

the types of nodes and links are increasingly numerous, the features are increasingly complex and diverse, and the correlations between features are increasingly strong. To cope with the change of HINs, the paper constructs a robust and fast heterogeneous network representation model, called RFRL. Based on the idea of "divide and conquer", the model decomposes the heterogeneous network global features into intra-type local features and inter-type local features. Specifically, the model treats each node type and each link type as a feature subspace. The intra-type features refer to node proximity under each node type feature subspace, and inter-type features refer to semantic similarity between each link type feature subspace. Through multiple learning of intra-type and inter-type features instead of one learning of global features, the accuracy of feature learning increased. Moreover, the model uses shallow models instead of deep models to ensure the high-speed of feature learning, and use adversarial learning to enhance generalization of learned features and compatibility with network noise.

The overview of the RFRL model is shown in Figure 1. The whole model contains four parts.

- *Type-aware biased sampling* is the first part. In this part, a type-aware random walk strategy is designed to simultaneously generate intra-type training samples for each node type and inter-type training samples for each link type in one sampling process. Furthermore, the global information of node types is used as weights to generate the final biased intra-type samples of different node types. Meanwhile, the global information of link types is used as weights to generate the final biased inter-type samples of different link types.

- *Node-type-aware adversarial learning* is the second part. In this part, each node type is first viewed as an intra-type feature subspace. Then, based on the idea that "if two nodes are near neighbors, the coordinates of the two nodes in the subspace should be close", a shallow network is designed as a discriminator to accurately learn intra-type features of the subspace. Next, a noisy version of the same shallow model is used as a generator to generate fake intra-type features. The fake features are disguised as more unseen fake samples to cheat the discriminator. Finally, the discriminator identifies real and fake features from real and fake samples to generate more robust features and reduce the influence of network noise.

- *Link-type-aware adversarial learning* is the third part. Similar to above part, each link type is viewed as an inter-type feature subspace. Then, based on the idea that "if a link exists between two nodes with different types, the embedding vector of one node can reach the embedding vector of another node by

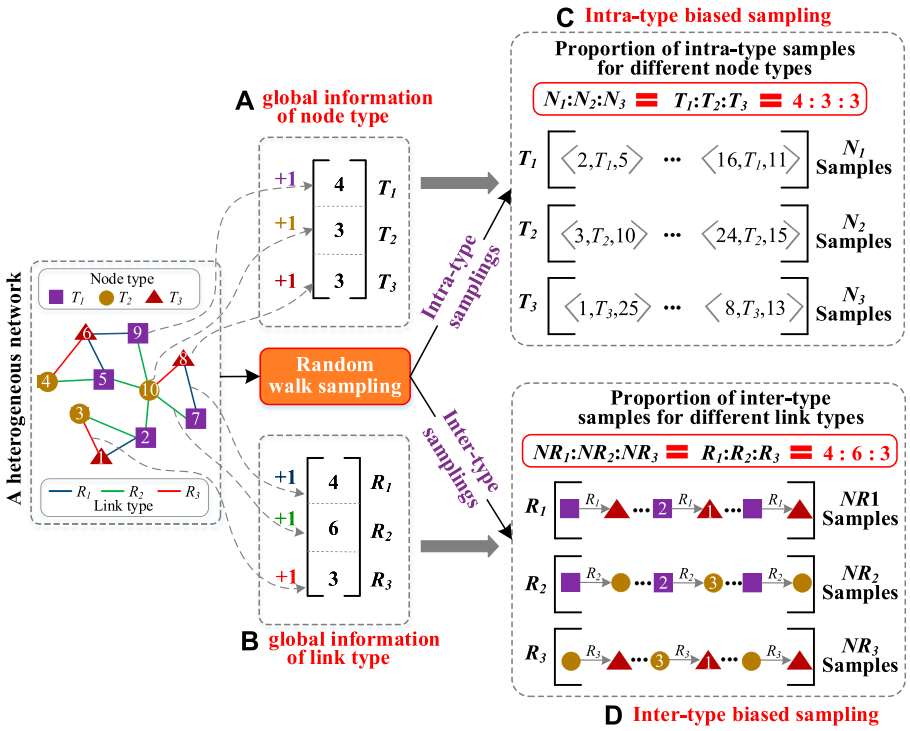


FIGURE 2
The process of type-aware biased sampling. **(A)** calculate global information of node types. **(B)** calculate global information of link types. **(C)** generate intra-type biased samples. **(D)** generate inter-type biased samples.

transforming the semantic features of the link", another shallow network is designed as a discriminator to accurately learn inter-type features of the subspace. Next, a noisy version of the same shallow model is used as a generator to generate fake inter-type features as unseen fake samples to cheat the discriminator. Finally, adversarial learning between generator and discriminator can capture more robust and generalized inter-type features and weaken the impact of network noise.

- *Joint training* is the fourth part. This part uses the intra-type features of each node as shared parameters. By the alternate execution of intra-type feature learning and inter-type feature learning, the global node representation is further improved.

4.2 Type-aware biased sampling

To accurately learn intra-type and inter-type features in HINs, it is essential to generate suitable training samples for these features. This involves generating intra-type training samples for each node type and inter-type training samples for each link type. However, in HINs, different node types have different global distributions (in terms of the number of nodes with different types), and different link types also have different global distributions (in terms of the number of links with different types). This global information determines the importance of different intra-type features or different inter-type features in the global features. Therefore, the intra-type training samples should satisfy the global distribution of node types, and inter-type training samples should satisfy the global distribution of link types. Moreover, due to intra-type features and inter-type features are coupled to each other, it is important to use a same sampling process to generate both intra-type and inter-type training samples, so as to preserve the coupling between them.

Based on the above ideas, supervised by the global distribution information, a type-aware biased sampling strategy is designed to simultaneously generate intra-type samples for each node type and inter-type samples for each link type. The detailed procedure of this strategy consists of three steps, as shown in Figure 2.

Step 1, *calculate global information of node types and link types.* First, grouping the network topology by node type, the proportion of nodes in each group is the weight W_N of all intra-type features in global feature learning, as shown in Figure 2A. Second, grouping the network topology by link type, the proportion of links in each group is the weight W_L of all inter-type features in global feature learning, as shown in Figure 2B.

Step 2, *type-aware sampling.* First, a sample queue $Q_N(T_k)$ is assigned to each node type T_k for storing intra-type feature samples and a sample queue $Q_L(R_s)$ is assigned to each link type R_s for storing inter-type feature samples. After that, a node n_i as n_{pre} is randomly selected from the HIN as the starting point. The node n_{pre} walks randomly from its neighbors to the next node n_{pre} through a link. Following this, node n_{next} as n_{pre} walks to the next node n_{next} at random. Each walked node n_{next} is dropped into the queue Q_N of its type. And the link (n_{pre}, n_{next}) is dropped into the queue Q_L of its type. A walking ends when the number of walking nodes exceeds the preset random walk length. When all nodes in the network are used

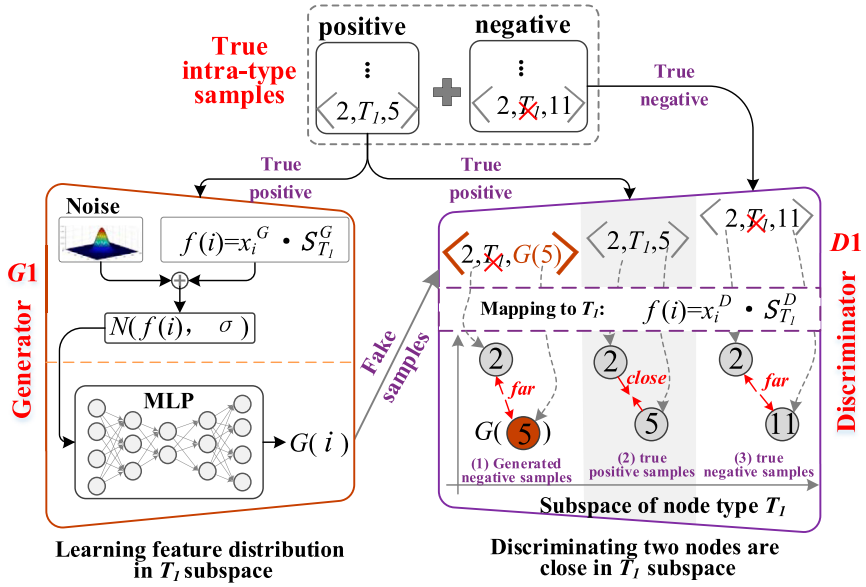


FIGURE 3
The process of node-type-aware adversarial learning.

as the starting point for a walking, the random walk process ends. Finally, in each queue $Q_N(T_k)$, the node sequence is further divided into multiple subsequences with window size L. In each subsequence, any two nodes n_i and n_j can form a positive sample $< n_i, T_k, n_j >$ to preserve 1-order and high-order intra-type features. And, two nodes n_i and n_m in different subsequences can generate a negative sample$< n_i$, no $T_k, n_j >$. Moreover, in each queue $Q_L(R_s)$, a link and its two vertices can generate a positive sample$< n_i, R_s, n_j >$ for the inter-type feature learning of type R_s. And, a positive sample $< n_x, R_l, n_h >$ of type R_l can be transformed to a negative sample $< n_x, R_s, n_h >$of type R_s by replacing R_l with R_s. In the two training samples, the ratio of positive samples to negative samples is 1:3.

Step 3, *generate biased samples*. After sampling, the intra-type training samples (or inter-type training samples) must be proportionally consistent with the weights W_N (or W_L) calculated in step 1. Therefore, the expected number of each intra-type training samples (or inter-type training samples) is first calculated based on the weight W_N (or W_L). Then, for a node type (or link type), if the number of generated samples is larger than the expected number, the redundant part is randomly removed. Finally, if the number of generated samples is smaller than the expected number, the missing parts are randomly copied from the existing samples.

4.3 Node-type-aware adversarial learning for intra-type features

The first important feature of a HIN is the intra-type feature, which refers to the proximity among multiple nodes with the same type. In this paper, each node type corresponds to a feature subspace with a special intra-type feature. Compared to global features, each intra-type feature will be purer, and its learning process will be simpler and accurate. Moreover, the difference in the number of training samples for different node types helps more accurate intra-type features to meet global distribution. In addition, the learning process of multiple intra-type features can be executed in parallel to improve the speed of the representation model. In order to further resist the noise in the network, we try to employ the generative adversarial network (GAN) to learn the distribution of each intra-type feature. This allows us to generate more unseen training samples and reduce the impact of noise on model training. By using the GAN, we are able to capture more robust and generalized intra-type features for nodes.

Based on the above ideas, a node type-aware adversarial learning strategy is designed to use the shallow network instead of deep network to learn more robust and accurate intra-type features of nodes, as shown in Figure 3. The whole strategy consists of two components: the generator and the discriminator, which engage in a game to learn more robust and accurate the intra-type features of nodes. Then, a noised version of the same shallow model is used as a generator to generate fake intra-type features. The fake features are disguised as more unseen fake samples to cheat the discriminator. Next, the discriminator identifies real and fake features from real and fake samples. Finally, adversarial learning between generator and discriminator can capture more robust and generalized inter-type features and weaken the impact of network noise. As an example, the adversarial learning process of the intra-type feature learning for node type T_l is as follows.

4.3.1 Intra-type feature learning

Based on the idea that "if two nodes are neighbors, the coordinates of the two nodes in the subspace should be close", the intra-type feature learning of node type T_l is to project two T_l-type nodes into the same feature subspace, and then pull the

coordinates of the two nodes as close together as possible when the two nodes are neighbors, or push the coordinates of the two nodes as far away as possible when the two nodes are not neighbors. Taking two T_1-type nodes i and j as examples, the intra-type feature learning process is as follows.

First, the two nodes are mapped to the feature subspace of node type T_1, and the position coordinates of the two nodes are obtained respectively, as follows.

$$\begin{aligned} f(i) &= x_i \cdot S^{T_1} \\ f(j) &= x_j \cdot S^{T_1} \end{aligned} \tag{1}$$

Where x_i (or x_j)$\in \mathbb{R}^{1\times d}$ is the global representation vectors of node i (or node j). The $S^{T1}\in \mathbb{R}^{d\times d}$ is the projection matrix of node type T_1, which represents the feature subspace of node type T_1.

Second, since the coordinates of two nodes are vectors, the inner product of the two vectors is used to calculate the proximity of the coordinates of the two nodes in the feature subspace, as follows.

$$dis(i,j) = f(i) \cdot [f(j)]^T \tag{2}$$

Then, the *Sigmoid* function is used to regularize the distance between nodes for easy comparison, as follows.

$$Sim(i,j) = sigmoid(dis(i,j)) = \frac{1}{1+\exp(-dis(i,j))} \tag{3}$$

After regularization, the distance between two nodes is quantified into the range of [0, 1]. If two T_1-type nodes are neighbors, then the distance between the two nodes in the subspace of node type T_1 should converge to 1, otherwise the distance between the two nodes should converge to 0.

4.3.2 Intra-type generator

Based on the above pattern of feature learning, we try to design a noisy feature extractor as a generator **G1** to generate fake intra-type features of node type T_1, so as to help the model tolerate noise and extract more robust and general intra-type features. Specifically, the fake intra-type feature generation process of node type T_1 is as follows.

Fake intra-type feature generation. First, another projection matrix $S_G^{T_1} \in \mathbb{R}^{d\times d}$ of node type T_1 is generated and randomly initialized, which represents the feature subspace of node type T_1 in the generator **G1** and competes with the projection matrix $S^{T1}$of the T_1-type real feature subspace.

Second, one node i is mapped to the T_1-type fake feature subspace of **G1**, and the position coordinate is calculated respectively, as follows.

$$f_G(i) = x_i \cdot S_G^{T_1} \tag{4}$$

Where $f_G(i)\in \mathbb{R}^{1\times d}$ is the coordinate of node i in T_1-type real feature subspace.

Then, to generate noisy fake intra-type features, we use Gaussian noise to disturb the coordinate of node i in the T_1-type fake feature space of **G1**, defined as follows.

$$f_G^{noise}(i) = N(f_G(i), \sigma^2 I) \tag{5}$$

Where the σ is the variance of Gaussian noise, which is a preset hyper-parameter. The I is the unit vector, $N(*,\sigma)$ is a function to generate a Gaussian distribution with mean * and variance σ.

Finally, we use a multi-layer perceptron to enhance the nonlinearity of the noise coordinate of node i, and defined as.

$$f_G^{noise}(i) = g(\cdots g(f_G^{noise}(i)\cdot W_1 + b_1)\cdots W_k + b_k) \tag{6}$$

Where $W\in \mathbb{R}^{d\times d}$ is the parameter matrix of the MLP and $b\in R^{1\times d}$ is the bias of the MLP. And the MLP is set to one layer in this paper. The $g()$ is the nonlinear activation function(*LeakyReLU* is used in this paper).

Loss function. To ensure the effectiveness of the generated fake feature distribution, we hope that the generated fake feature distribution is as close as possible to the real distribution. To achieve this goal, we use the positive intra-type samples <i, T_1, j> of node type T_1 to train the generator **G1**.

Therefore, the loss function of generator **G1** consists of two parts. The first part is that the coordinates of two nodes of one positive sample in **G1** should be as close as possible, defined as follows.

$$Loss_1 = -\sum_{<i,j>\in \left\{ \begin{matrix} postive \\ samples \end{matrix} \right\}} \log(Sim(f_G^{noise}(i), f_G^{noise}(j))) \tag{7}$$

The second part is that the fake coordinate in **G1** of any node in one positive sample should be as close as possible to the coordinate in the real subspace, defined as follows.

$$\begin{aligned} Loss_2 = -\sum_{<i,j>\in \left\{ \begin{matrix} postive \\ samples \end{matrix} \right\}} & \log(Sim(f_G^{noise}(i), f(i))) \\ & + \log(Sim(f_G^{noise}(j), f(j))) \end{aligned} \tag{8}$$

Where $f(j)$ and $f(i)$ is the representation vector (the coordinate) in the real feature subspace of node type T_1.

The final loss of generator **G1** is defined as follows.

$$Loss_{G1} = Loss_1 + Loss_2 \tag{9}$$

4.3.3 Intra-type discriminator

As a game competitor of the generator **G1**, a discriminator **D1** needs to be constructed. In this way, the generator **G1** and the discriminator **D1** form an adversarial generative network GAN. In this paper, to simplify the structure of GAN, the **D1** needs to have two capabilities. The first capability is to learn the true feature distribution in the feature subspace of node type T_1. The second capability is to identify the real feature distribution and the fake feature distribution.

Real intra-type feature learning and discrimination. The fake feature distribution f_G^{noise} generated by **G1**, positive intra-type samples <i, T_1, j> and negative intra-type samples<i, *not* T_1, j> generated by the type-aware biased sampling strategy are inputs of the discriminator **D1**. To achieve the two capabilities of **D1**, three inputs are transformed into three types of training samples.

The first type of training samples are real positive samples <i, T_1, j>$\in RPS$, and they are generated by the type-aware biased sampling strategy. For a real positive sample<i, T_1, j>, the proximity of the coordinates of two nodes i and j in the T_1-type true feature subspace of should be 1. That is to say, $Sim(i,j) = 1$ according to Equation 3.

The second type of training samples are real negative samples <i, *not* T_1, j>$\in RNS$, and they are also generated by the type-aware biased

sampling strategy. For a real negative sample<*i, not* T_1, *j*>, the proximity of the coordinates of two nodes *i* and *j* in the T_1-type true feature subspace of should be 0. That is to say, $Sim(i,j) = 0$ according to Equation 3.

The third type of training samples are fake positive samples <*i*, T_1, $G(j)$>(or <$G(i)$, T_1, *j*>)$\in FPS$ where the $G(j)$ represents the fake feature vector of node *j* in **G1**. For a fake positive sample <*i*, T_1, $G(j)$ >, the proximity of the coordinates of two nodes *i* and *j* in the T_1-type true feature subspace should be 0. That is to say, $Sim(i,j) = 0$ according to Equation 3.

When the discriminator **D1** is trained only with real positive and real negative samples, the **D1** can accurately capture the true feature distribution in the T_1-type feature subspace. When the discriminator **D1** is trained with real positive and fake positive samples, the **D1** can accurately identify the real and fake feature distribution. Therefore, through the combined training of three training samples, the **D1** can not only learn the real feature distribution, but also distinguish the real and fake feature distributions. Moreover, driven by the fake feature distribution generated by generator **G1**, a large number of unseen training samples are generated. These fake samples can further help **D1** to learn more robust and generalized intra-type features, and reduce the influence of network noise.

Loss function. According to three types of training samples, the loss function of discriminator **D1** consists of three parts, defined as.

$$\begin{aligned} Loss_{D1} = & \sum_{<i,j>\in RPS} -\log\left(Sim\left(f(i), f(j)\right)\right) \\ & + \sum_{<i,j>\in RNS} -\log\left(1 - Sim\left(f(i), f(j)\right)\right) \\ & + \sum_{<i,j>\in FPS} -\log\left(1 - Sim\left(f(i), f_G^{noise}(j)\right)\right) \\ & - \log\left(1 - Sim\left(f_G^{noise}(i), f(j)\right)\right) \end{aligned} \tag{10}$$

4.4 Link-type-aware adversarial learning for inter-type features

The second important feature of a HIN is the inter-type feature, which refers to the semantic similarity between two node types. That is to say, if there is a link of type R_1 between node *i* of type T_1 and node *j* of type T_2, then the node *i* can reach the node *j* through the semantic relation R_1. In this paper, each link type is also regarded as a feature subspace possessing a unique inter-type feature that enables the semantic transformation of two heterogeneous nodes. More specifically, if a link of type R_1 exists between node *i* of type T_1 and node *j* of type T_2, then the T_1-type intra-type feature of node *i* can be similar to the T_2-type intra-type feature of node *j* through the semantic transformation of relation R_1, abbreviated as <*i*, T_1, R_1, T_2, *j*>. Similar to the intra-type feature learning, the difference in the number of training samples for different link types helps more accurate inter-type features to meet global distribution. To reduce the impact of network noise, we also employ the generative adversarial network (GAN) to learn the more robust and generated inter-type feature distribution.

Based on the above ideas, a link-type-aware adversarial learning strategy is designed to learn more robust and accurate inter-type features of nodes, as shown in Figure 4. The whole strategy consists of generators and discriminators, each of which uses a shallow network. It is important to note that the inter-type feature learning aims to capture the relation between two intra-type feature spaces. For instance, in the case of link type R_1, we use a generator and a discriminator to learn inter-type features as follows.

4.4.1 Inter-type feature learning

Based on the idea that "if a link exists between two nodes with different types, the embedding vector of one node can be transformed to the embedding vector of another node by the semantic features of the link", the inter-type feature of link type R_1 is to learn the transform from the intra-type feature of one node to the intra-type feature of another node in the R_1 feature subspace. Then, the two intra-type feature vectors are pushed close when there is an R_1 relation between them, and the two feature vectors are drawn far apart when there is not an R_1 relation between them. Taking T_1-type node *i* and T_2-type node *j* with an edge of type R_1 as example<*i*, T_1, R_1, T_2, *j*>, the inter-type feature leaning process is as follows.

First, the intra-type feature $f(i)$ of node *i* in T_1-type feature space and the intra-type feature $f(j)$ of node *j* in T_2-type feature space are first obtained from the Section 4.3. Meanwhile, the link type R_1 is regarded as a semantic feature subspace, and a transformation matrix S_{R1} is defined to represent the semantic transformation process in the subspace.

Second, the intra-type feature $f(i)$ of T_1-type node *i* is mapped to the R_1 feature subspace, and defined as.

$$Tran(i) = f(i) \cdot S_{R_1} \tag{11}$$

Where $S_{R1} \in \mathbb{R}^{d\times d}$ is the semantic transformation matrix of link type $_{R1}$, and the $\cdot$ is the function of matrix multiplication.

Then, the similarity between the transformation feature $Tran(i)$ of node i and the intra-type features $f(j)$ of node *j* is calculated as.

$$Sim(i,j) = Tran(i) \cdot [f(j)]^T \tag{12}$$

Finally, the *Sigmoid* function is used to regularize the similarity to the range of [0,1], as follows.

$$Sim(i,j) = \frac{1}{1+\exp\left(-Sim(i,j)\right)} = \frac{1}{1+\exp\left(-Tran(i)\cdot[f(j)]^T\right)} \tag{13}$$

The similarity is adjusted according to the correctness of the semantic transformation of R_1. Specifically, the value of similarity should be increased to 1 when $Tran(i)$ can correctly reach the T_2 features space. Conversely, the value of similarity should be decreased to 0.

4.4.2 Inter-type feature generator

Same as the intra-type generator, the generator **G2** is used to generate the noisy fake inter-type features and apply them to the inter-type discriminator **D2**, so as to learn more robust features and reduce the impact of the network noise. The generating process for the fake inter-type features of R_1 is as follows.

Fake inter-type feature generation. First, another projection matrix $S_G^{R_1} \in \mathbb{R}^{d\times d}$ of link type R_1 is generated and randomly initialized, which represents the fake feature subspace of R_1 in

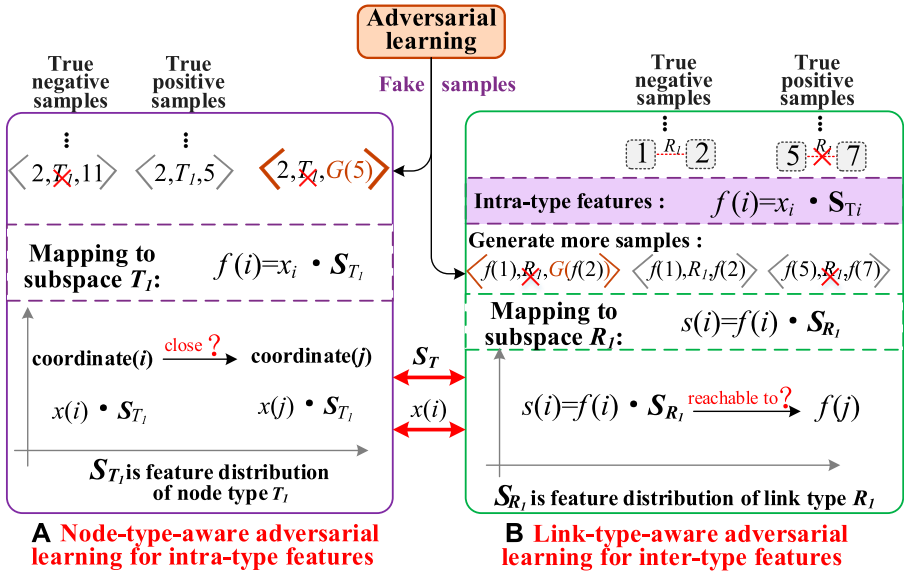


FIGURE 4
Comparison of intra-type feature learning and inter-type feature learning. **(A)** Node-type-aware adversarial learning for intra-type features. **(B)** Link-type-aware adversarial learning for intra-type features.

G2. And, the intra-type feature of node *i* is transformed by the fake feature subspace in **G2** to obtain the transformation feature $Tran_G(i)$.

$$Tran_G(i) = f_G(i) \cdot S_{R1}^G \tag{14}$$

Second, to better obtain the fake inter-type feature, we add some Gaussian noise to the inter-type feature $Tran_G(i)$, defined as follows.

$$Tran_G^{noise}(i) = N(Tran_G(i), \sigma^2 I) \tag{15}$$

Where the size of the variance σ can be adjusted to control the degree of disturbance of Gaussian noise.

Finally, to enhance the expressiveness of $Tran_G^{noise}(i)$, the model continues using the MLP model with two layers to obtain the final fake inter-type features in the following way.

$$Tran_G^{noise}(i) = g(\cdots g(Tran_G^{noise}(i) \cdot W_1 + b_1) \cdots W_k + b_k) \tag{16}$$

Where $W \in \mathbb{R}^{d \times d}$ is the parameter matrix of the MLP and $b \in R^{1 \times d}$ is the bias of the MLP. And the $g()$ is the nonlinear activation function (*LeakyReLU* is used in this paper).

Loss function: For inter-type features, the concern is whether the node intra-type features can reach the correct feature subspace after the semantic transformation. Therefore, we use the *cross-entropy* loss to make the fake inter-type features *Trannoise* $G(i)$ similar to the true intra-type features $f(j)$, so that the fake inter-type features of type T_1 can arrive correctly in the feature subspace of T_2. To achieve this goal, we use the positive inter-type samples $<i, T_1, R_1, T_2, j>$ to train the generator **G2**. The details are as follows.

$$Loss_{G2} = -\sum_{<i,j> \in \left\{ \begin{array}{c} postive \\ samples \end{array} \right\}} \log(Sim(Tran_G^{noise}(i), f(j))) \tag{17}$$

4.4.3 Inter-type feature discriminator

Similarly, we need an inter-type discriminator **D2** as a competitor to the inter-type generator **G2**. In this way, the generator **G2** and the discriminator **D2** form an adversarial generative network GAN. Similar to **D1**, **D2** also needs to have two capabilities. The first capability is to learn real inter-type features. The second capability is to achieve discrimination between true inter-type features and fake inter-type features.

Real inter-type feature learning and discrimination. The fake feature distribution *Trannoise* $G(i)$ generated by **G2**, positive inter-type samples $<i, T_1, R_1, T_2, j>$ and negative intra-type samples $<i, T_1,$ not $R_1, T_2, j>$ generated by the sampling strategy are inputs of the discriminator ***D2***. To achieve the two capabilities of *D2*, three inputs are transformed into three types of training samples.

The first type of training samples are real positive samples. For the real positive samples $<i, T_1, R_1, T_2, j> \in RPS$, the similarity of the inter-type features $Tran(i)$ and $f(j)$ should be 1. That is to say, $Sim(i,j) = 1$ according to Equation 13.

The second type of training samples are real negative samples. For a real negative sample $<i, T_1,$ not $R_1, T_2, j> \in RNS$, the similarity of the inter-type features $Tran(i)$ and $f(j)$ should be 0. That is to say, $Sim(i,j) = 0$ according to Equation 13.

The third type of training samples are fake positive samples $<G2(i), T_1, R_1, T_2, j>$ (or $<i, T_1, R_1, T_2, G2(j)> \in FPS$. For a fake positive sample $<G2(i), T_1, R_1, T_2, j>$, the similarity of the inter-type features $Tran_G^{noise}(i)$ and $f(j)$ should be 0. That is to say, $Sim(G2(i), j) = 0$ according to Equation 13.

When discriminator **D2** is trained with only true positive samples and true negative samples, **D2** can accurately capture the true inter-type feature distribution. When discriminator **D2** is trained with true positive and fake positive samples,

D2 can accurately identify the true and fake feature distributions. Therefore, through the joint training of three training samples, **D2** can not only incrementally learn the true inter-type features, but also achieve the discrimination of true inter-type features and fake inter-type features. In addition, the large number of unseen training samples generated by generator **G2** can further help *D*2 to achieve robust and accurate learning of inter-type features and thus reduce the effect of network noise.

Loss function. Based on the above idea, the *cross-entropy* loss function is used as the loss of *D*2, and defined as follows.

$$\begin{aligned} Loss_{D2} = & \sum_{<i,j>\in RPS} -\log\left(Sim\left(Tran(i), f(j)\right)\right) \\ & + \sum_{<i,j>\in RNS} -\log\left(1 - Sim\left(Tran(i), f(j)\right)\right) \\ & + \sum_{<i,j>\in FPS} -\log\left(1 - Sim\left(Tran_G^{noise}(i), Tran(i)\right)\right) \end{aligned} \quad (18)$$

4.5 Joint learning

Comparing intra-type feature learning with inter-type feature learning, it is not difficult to get the following finding, as shown in Figure 4. First, the two learning processes use different shallow learning strategies. Intra-type feature learning attempts to learn the positional proximity of the projected coordinates of two nodes in the same subspace. Inter-type feature learning attempts to learn the semantic similarity between two nodes by the semantic transformation of one link type. Second, the two learning processes are closely related to each other. The global representation vector of each node is shared in the two learning processes. Moreover, the learned intra-type features of node types are used as input in the inter-type feature learning process. In addition, an adversarial learning strategy is used in two learning processes to capture more robust and generalized network features and reduce the impact of network noise.

Therefore, node-type-aware intra-type feature learning and link-type-aware inter-type feature learning need to be jointly trained for better performance. In our experiment, the intra-type feature learning process is performed for 5 consecutive epochs with separate sampling, where the sampled intra-type samples are used and the inter-type samples are kept. After the intra-type features are learned, the inter-type feature learning process are performed without sampling for 5 epochs, where each epoch uses the inter-type samples reserved by the intra-type feature learning process. This can help the two learning processes capture the coupling between the intra-type feature and inter-type feature. All training procedures use the stochastic gradient descent algorithm (SGD) for parameter updates. The ratio of the iterations of generator and discriminator is adjusted during training to balance the learning rates of both, resulting in a steady improvement in the performance of both. In addition, adjusting the ratio of two learning rates also controls the learning speed of both. However, it is important to note that adjusting the learning rate may cause the performance of the generator and the discriminator to degrade. In this paper, the learning rate of the generator and discriminator is set to 1e-4.

5 Experiments

5.1 Experimental setup

Datasets. We select three datasets with different sparsity levels, which are related to the literature citation network, the shopping network, and the business network. (1) *DBLP network* is a citation network with journal and conference bibliographic information (https://dblp.uni-trier.de/xml/). (2) *Amazon network* comes from the user and product information of the Amazon platform (http://jmcauley.ucsd.edu/data/amazon/). (3) *Yelp network* contains information on merchants and users in multiple cities in the United States (https://www.yelp.com/dataset). The details of each dataset are shown in Table 1.

Baselines. In the comparison experiments, we selected five models as baselines: two shallow models (RHINE [24] and Metapath2vec [10]), two deep models (HAN [29] and HGT [32]), and one GAN-based representation model (HeGAN [33]). In addition, recommended meta-paths and default parameter settings are used for all models. The details of the baselines are shown in Table 2.

Tasks and metrics. We chose the following three tasks and five metrics to comprehensively evaluate the performance of our model. (1) *Node classification*. Based on the learned node representation vector, the classifier predicts the labels of the nodes. In this task, we use Macro_F1 and Micro_F1 metrics to evaluate the performance of the node classification task. (2) *Node clustering*. Based on the learned node representation vector, the nodes are divided into multiple clusters, where each cluster represents a category. We evaluate the performance of the node clustering task using the NMI metric. (3) *Link prediction*. Based on the learned node representation vector, we predict whether there is a link between two nodes. We use AUC and ACC metrics to evaluate the performance of link prediction tasks.

Setting. The experimental platform is a PC server equipped with an NVIDIA T4 card. The server is outfitted with a 32-core Intel Xeon Cascade Lake (2.5 Hz) processor, 64 GB of RAM, and the Ubuntu 18.04 operating system. The RFRL algorithm was programmed using the PyCharm IDE.

5.2 Node classification

In this section, we perform the node classification task on three datasets Amazon, DBLP, and Yelp, compared to five baselines to test our performance. In this experiment, to verify the stability, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% of the real labeled samples as 9 training datasets are used to train all models, and 10% of other real labeled samples are used as the test set. For the classifier, we use *SoftMax* to calculate the probability of each category and evaluate the classification result with Macro_F1 and Micro_F1 metrics.

Figure 5 shows the accuracy of node classification on 3 different networks, where Figures 5A–C are the Macro_F1 metrics and Figures 5D–F are the Micro_F1 metrics. In

TABLE 1 Datasets.

Dataset	Nodes	Number of nodes	Relation	Number of relations	Avg.Degree
DBLP	Author(A)	14,475			9.04
	Paper(P)	14,376	P-A	41,794	
	Conference(C)	20	P-C	14,376	
	Type(T)	8,920	P-T	114,624	
Amazon	User(U)	344			10.29
	Item(I)	95	I-U	365	
	View(V)	3,773	I-V	195,791	
	Brand(B)	5	I-B	95	
Yelp	User(U)	1,286			19.77
	Business(B)	2,614	B-U	30,838	
	Service(S)	2	B-S	2,614	
	Star(St)	9	B-St	2,614	
	Reservation(R)	2	B-R	2,614	

TABLE 2 Baselines.

Algorithms	Full name	Implement
RHINE [24]	Relation Structure-Aware Heterogeneous Information Network Embedding	Python
Metapath2vec [10]	metapath2vec: Scalable Representation Learning for Heterogeneous Networks	Python
HAN [29]	Heterogeneous Graph Attention Network	Python
HGT [32]	Heterogeneous Graph Transformer	Python
HeGAN [33]	Adversarial Learning on Heterogeneous Information Networks	Python

each graph, we used different colored lines to indicate the performance scores of different algorithms. By observing, we can obtain the following results. (1) *Comparing all models*, the best average performance in the three datasets is RFRL and HGT, followed by HeGAN, HAN, and finally RHINE, metapath2vec. For example, in the Macro_F1 metric, RFRL outperforms the second-best HGT by 0.76% on average across the three datasets; In the Micro_F1 metric, the average performance of RFRL is 1.01% higher than that of the second-best HGT. (2) *Considering the stability*, the stability of the RFRL model is better than that of the shallow model and better than that of the deep model on three networks. For example, on Macro_F1 of the DBLP dataset, the performance drift range of 3.75% for RFRL is lower than the ranges of 4.33% for metapath2vec and 5.21% for HGT. Meanwhile, for the training set with fewer labels, the RFRL model outperforms the other models. For example, on Amazon network, the Micro_F1 score of our RFRL model is 6.22% higher than the second-best model in the training set with 20% labels. (3) *Comparing the structure of the models*, the deep models (HGT and HAN) generally outperforms the shallow models (metapath2vec and RHINE). For example, on the Amazon dataset, the average performance of Macro_F1 of HGT is 3.01% higher than that of RHINE. (4) *Considering the potential of capturing information*, the RFRL model has a better potential to capture information than other models. That is to say, it can obtain good performance on fewer training sets. For example, on the Amazon training sets with 10%–30% labels, RFRL is on average about 3% better than the second-best Macro_F1 metric. (5) *In summary*, the RFRL model has high generalization ability and performance, especially when most of the node information is unknown (few label samples).

5.3 Node clustering

In this section, we test our performance on the node clustering task by choosing the same datasets and baselines as in the above experiments. In this experiment, we use the k-means algorithm to divide the nodes into multiple clusters based on the learned node representation vectors, each of which is a category. Finally, the NMI metric is used to evaluate the consistency of these categories with the true labeled categories.

Table 3 shows the comparison of the best clustering results of the six algorithms on the three datasets. The following

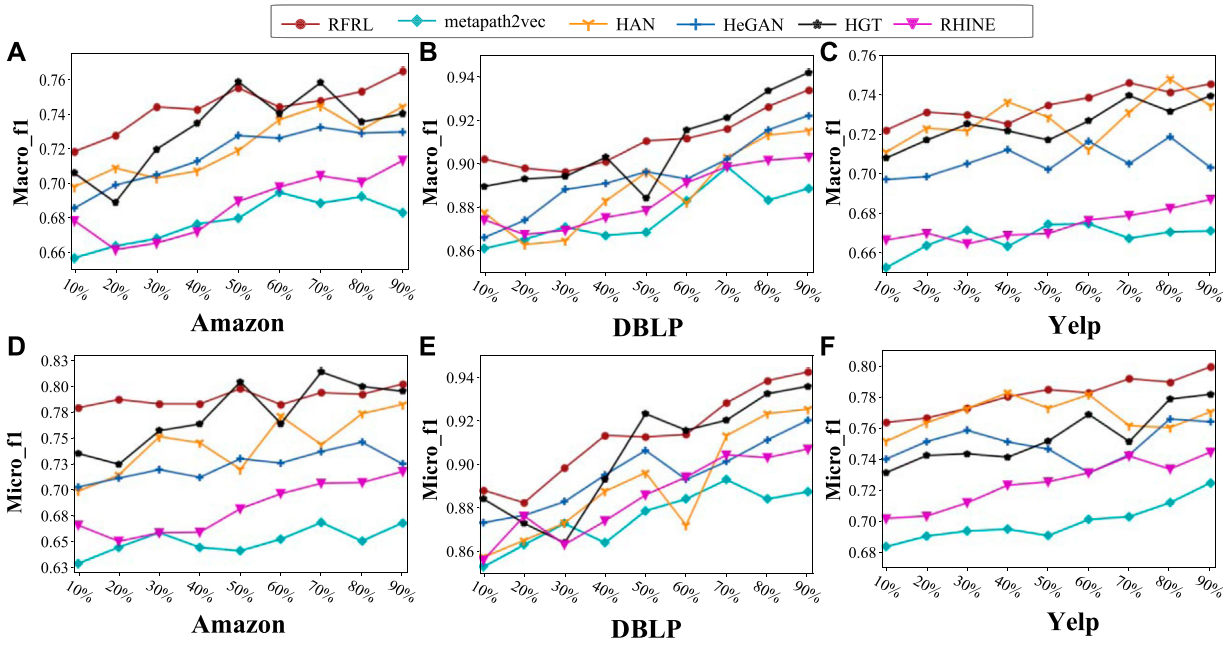


FIGURE 5
Performance of multiple algorithms in node classification. **(A)** The Macro-f1 scores of multiple algorithms on the Amazon dataset; **(B)** The Macro-f1 scores of multiple algorithms on the DBLP dataset; **(C)** The Macro-f1 scores of multiple algorithms on the Yelp dataset; **(D)** The Micro-f1 scores of multiple algorithms on the Amazon dataset; **(E)** The Micro-f1 scores of multiple algorithms on the DBLP dataset; **(F)** The Micro-f1 scores of multiple algorithms on the Yelp dataset.

TABLE 3 Performance of multiple algorithms in node clustering.

	Amazon	Yelp	DBLP
	NMI	NMI	NMI
metapath2vec	0.2989	0.3069	0.6738
RHINE	0.3479	0.3739	0.7352
HAN	0.3893	0.3631	0.7831
HGT	0.3981	0.3871	0.7438
HeGAN	0.367	0.3965	0.7689
RFRL	0.4375	0.4253	0.7576

conclusions are drawn from Table 3. (1) *Comparing all models*, the clustering performance of RFRL is higher than baseline algorithms in the three datasets, which proves the effectiveness and accuracy of RFRL. (2) *Considering the stability*, RFRL is more stable relative to the other models. For example, the smallest differences from the peak in the three datasets are 0.802% for RFRL, 2.75% for HeGAN, and 3.68% for HAN. (3) *Comparing the structure of the models*, GAN-based models (HeGAN, RFRL) generally outperform the other models. For example, in the Yelp dataset, the NMI of HeGAN is 0.3965 and that of RFRL is 0.4253; while the NMI of the other models is below 0.3872. In addition, the deep models (HAN, HGT) generally outperform the shallow models (metapath2vec, RHINE). For example, the NMI of HAN and HGT are roughly 1%–9% higher than metapath2vec and RHINE in all three datasets. (4) *Comparing GAN-based models*, RFRL outperforms HeGAN overall. For example, in the Amazon dataset, HeGAN has an NMI of 0.397 and RFRL has an NMI of 0.4375. (5) *In summary*, the RFRL model is robust and effective in the node clustering task.

5.4 Link prediction

In this section, we test our model using the link prediction task. In this task, we concatenate the node representation vector at the two ends of the link as a low-dimensional feature vector of the link, and then use logistic regression to implement a binary classification to determine the existence of a link. As with node classification, to demonstrate the stability of the model, we still use 5 baselines as competitors and 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% of the real labeled samples as 9 training datasets for comparison experiments.

Figure 6 shows the link prediction performance of the six algorithms on three datasets. The following conclusions can be drawn from the observations. (1) *Comparing the ACC of all models*, the best average performance is achieved by RFRL and HeGAN, followed by HGT and HAN, and finally RHINE and metapath2vec. For example, in the ACC metrics, the average ACC performance of RFRL is 66.53%, and the average ACC performance of HAN is 64.97%. (2) *Comparing the AUC of all models*, the performance of HeGAN has decreased. The best performances on average were RFRL and HGT, followed by

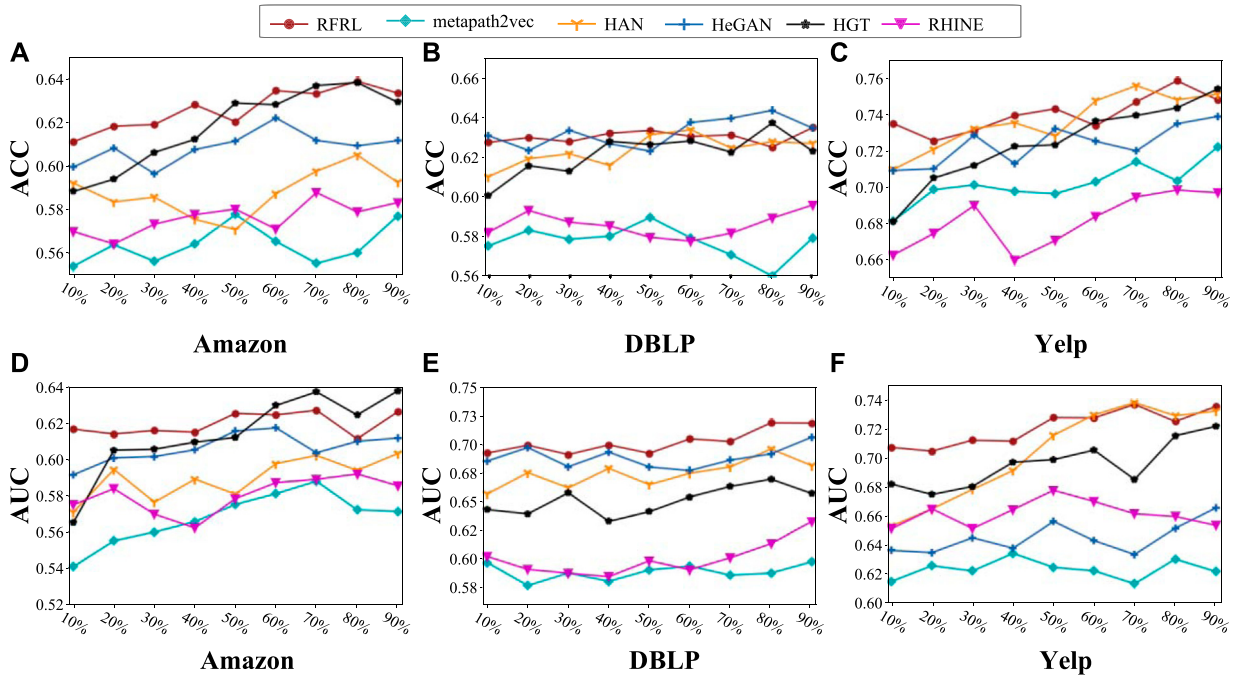


FIGURE 6
Performance of multiple algorithms in link prediction. **(A)** The ACC scores of multiple algorithms on the Amazon dataset; **(B)** The ACC scores of multiple algorithms on the DBLP dataset; **(C)** The ACC scores of multiple algorithms on the Yelp dataset; **(D)** The AUC scores of multiple algorithms on the Amazon dataset; **(E)** The AUC scores of multiple algorithms on the DBLP dataset; **(F)** The AUC scores of multiple algorithms on the Yelp dataset.

HeGAN and HAN, and finally RHINE and metapath2vec. For example, on the Yelp dataset, the average performance of HeGAN is 64.50% and that of RFRL is 72.10%, a difference of 7.60%. (3) *Considering the stability*, the GAN-based models RFRL and HeGAN are the best. For example, in the DBLP dataset, the extreme differences of RFRL and HeGAN are 2.76% and 2.89%. Meanwhile, the performance of these two GAN-based models is relatively better in the training set with 30% labels. For example, on the ACC of Amazon dataset, the average performance of RFRL and HeGAN is 61.62% and 60.15%. (4) *Comparing the structure of the models*, the GAN-based models (RFRL and HeGAN) outperform the other models on average, and the deep models (HGT and HAN) outperform the shallow models (RHINE and Metapath2vec) overall. For example, in DBLP dataset, the average AUC value of HGT is 4.41% higher than that of metapath2vec and 3.57% higher than that of RHINE. (5) *In summary*, RFRL outperforms the other baselines by 1%–13% on the link prediction task. The good performance of RFRL on the datasets with less real samples demonstrates the effectiveness and robustness of the model.

5.5 Additional experiments

To further demonstrate the advantages of the biased sampling strategy, the robustness and Scalability of the model, we perform the following additional comparison experiments based on the link prediction task.

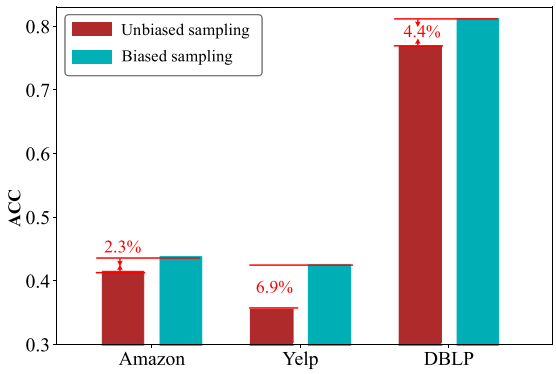


FIGURE 7
Performance of biased and unbiased sampling strategy in link prediction.

5.5.1 The biased sampling strategy

In this experiment, the samples collected by biased and unbiased sampling strategy are fed into our RFFL model as two different models for training, and finally, the performance of the two models is evaluated on the link prediction task using the ACC metric.

Figure 7 shows the performance on the ACC metric for link prediction with two different sampling strategies. In the figure, red bars represent the results of the unbiased sampling strategy and blue bars represent the results of the biased sampling strategy. We can get the following findings. (1) *Comparing*

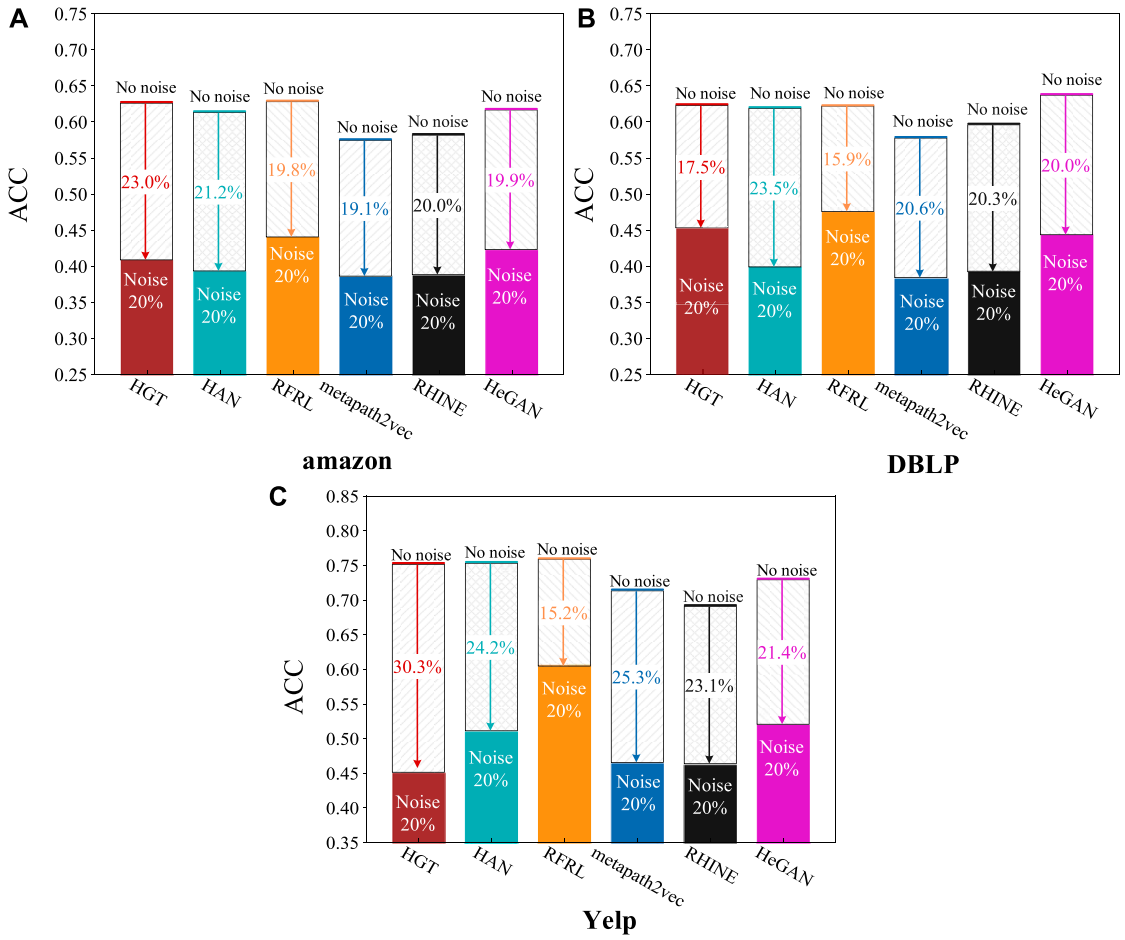


FIGURE 8
Performance of multiple algorithms in the noise networks. **(A)** The ACC scores of multiple algorithms on the Amazon dataset with noise; **(B)** The ACC scores of multiple algorithms on the DBLP dataset with noise; **(C)** The ACC scores of multiple algorithms on the Yelp dataset with noise.

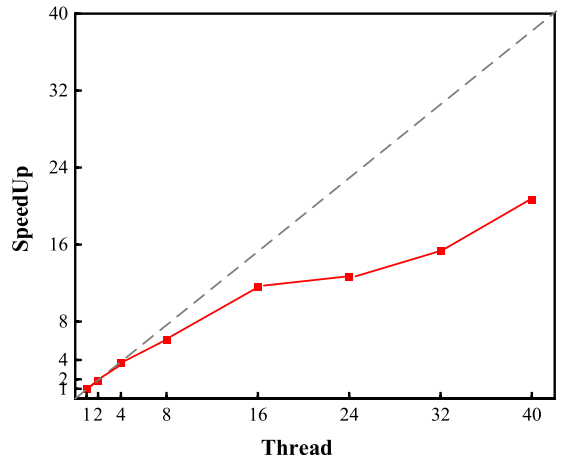


FIGURE 9
The scalability performance of the RFRL model.

the two strategies, the overall performance of biased sampling is better than random sampling. For example, on the Amazon, Yelp and DBLP datasets, the performance improves by 2.25%, 6.93% and 4.4%, respectively. This demonstrates that the biased sampling can indeed better preserve the feature information of heterogeneous networks. (2) *Comparing different datasets*, the performance optimizations of the biased sampling strategy on different datasets are different. For example, the optimization on the Yelp dataset is 6.93%, while on the DBLP it is only 2.25%. This may be related to the sparsity of the data, as well as the number of types. Specifically, the Yelp dataset has a larger number of types and degrees than the other two datasets. (3) *In summary*, the biased sampling strategy captures the intra-type and inter-type features in the network better and performs better.

5.5.2 Robustness

In this experiment, a noise network is obtained by randomly removing 20% of the links from the original heterogeneous network. On the noise network, RFRL is compared with other baseline algorithms in terms of ACC metric for link prediction. The ratio of the training set to the test set is 9:1.

Figure 8 shows the link prediction performance of different algorithms on the noise networks. From the figure, we can

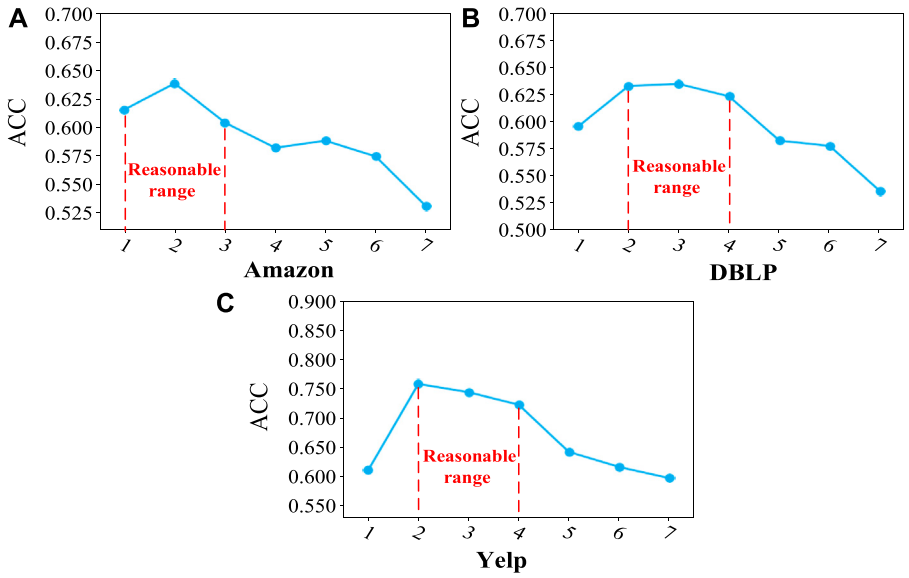


FIGURE 10
Sensitivity analysis of the hyper-parameter *L*. **(A)** The ACC scores of the different parameters L on the Amazon dataset; **(B)** The ACC scores of the different parameters L on the DBLP dataset; **(C)** The ACC scores of the different parameters L on the Yelp dataset.

observe the following information. (1) *Comparing all the models*, the overall performance of all models decreased. RFRL showed the smallest decrease in performance with an average decrease of 16.96%; followed by HeGAN, RHINE, metapath2vec with average decreases of 20.43%, 21.13%, 23.6%, and HAN, HGT with average decreases of 22.97% and 23.60%, respectively. (2) *Comparing different datasets*, the model performance degradation varies, which is related to the structure and semantics of the network. Although the degradation of RFRL in the Amazon dataset is slightly higher at 19.8% than that of metapath2vec at 19.1%, good noise control performance is achieved in all other datasets. For example, RFRL performance decreases by only 15.2% on Yelp. **(3)*In summary***, RFRL is more robust and more compatible with noise.

5.5.3 Scalability

In the era of big data, the scalability of the model is very important, so we further validate the scalability of the RFRL model. In this experiment, intra-type feature learning process is parallelized using multi-threading techniques, and the ratio of parallelized threads to accelerated multipliers is recorded. The experimental results are shown in Figure 9.

Figure 9 plots the number of threads *versus* the speedup multiplier. In the figure, the vertical axis is the acceleration multiplier, the horizontal axis is the number of concurrent threads, and the curve is the ratio between the two. By analyzing Figure 9, we can see that the RFRL model has a significant speedup with fewer threads. Specifically, a speedup of 10–12 times is achieved when using 16 threads for concurrent execution. As the number of threads increases, the speedup increases more slowly. For example, when 40 threads are executed concurrently, the speedup is only 18 to 19 times. In addition, the speedup almost stops growing when the experiment is performed concurrently with 78 threads.

5.6 Parameter sensitivity

The context window length L is an important hyper-parameter for our RFRL model, which determines the range of node intra-type proximity features needs to learn. In the type-aware biased sampling strategy, any two nodes in a context window constitute a positive sample. That is to say, if the window length L is larger, then the two nodes are farther away in the original network, and the order of proximity between them is higher. On the contrary, if the window length L is smaller, the closer the two nodes are in the original network, the lower the order of proximity between them is. In this experiment, the performance of our RFRL model with different window lengths is evaluated on the link prediction task using the ACC metric.

Figure 10 shows the ACC accuracy of our RFRL model with different window lengths L. From the figure, we can get the following observations. (1) *Focusing on a network*, the link prediction accuracy rises and then falls as the L value increases. For example, for the DBLP dataset, the ACC value increases rapidly in the range [1,2]. Then, in the range [2,4], the ACC values stabilize. Finally, in the range [4,7], the ACC value decreases slowly. This indicates that too small L values cannot fully capture the intra-type features of nodes in the network, and too large L leads to capturing imprecise intra-type features. Specifically, the relative stability range of the parameter L is [1,3] on Amazon, [2,4] on DBLP and Yelp. (2) *Comparing different datasets*, the window length of the Yelp and DBLP datasets should be larger than that of the Amazon dataset. This is because neighboring nodes in a walk sequence are more closely related in a dense network. It experimentally demonstrates that the first- and

second-order neighbor relations of nodes are the most worth learning, which can help the model to capture the proximity feature information between nodes well. (3) *In summary*, with context window length L in the range of [2,3], the model has the most stable performance for different datasets. In all our experiments, the value of the context window length is 2 as default.

6 Conclusion

In this paper, we propose a robust and fast representation learning model for heterogeneous networks, called RFRL. The RFRL model is well adapted to the following characteristics of future heterogeneous networks: larger scale, more diverse features, and stronger noise. *To better cope with large-scale networks*, two novel shallow learning strategies are designed to replace the traditional deep learning network to quickly generate the low-dimensional feature vectors of nodes. *To better learn complex and diverse features*, each node type and link type is treated as a feature subspace to perform representation learning separately. The RFRL model uses multiple learning processes for partial features instead of a single learning process for all features to achieve high speed and performance. *To reduce the impact of network noise*, GANs are further used to generate fake training samples in each subspace, and the adversarial learning between generator and discriminator can help the RFRL model to capture more robust and generalized node features. Extensive experimental results on multiple networks and multiple tasks demonstrate the performance of our model.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

LC: Conceptualization, Writing- Original draft preparation, Formal analysis, Supervision. YoL: Conceptualization, Validation, Writing- Reviewing and Editing, Investigation. YuL: Conceptualization, Validation, Visualization, Investigation. RX: Conceptualization, Supervision. Resources, Visualization. ZL: Methodology, Data curation, Validation. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the National Key Research and Development Program (No. 2019YFE0105300); the National Natural Science Foundation of China (No. 62103143), the Hunan Province Key Research and Development Program (No. 2022WK2006), the Special Project for the Construction of Innovative Provinces in Hunan (Nos. 2020TP2018 and 2019GK4030), the Young Backbone Teacher of Hunan Province (No. 2022101), and the Scientific Research Fund of Hunan Provincial Education Department (Nos. 22B0471 and 22C0829).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. He D, Liang C, Huo C, Feng Z, Jin D, Yang L, et al. Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning. *IEEE Trans Neural Networks Learn Syst* (2022) 2022:1–13. doi:10.1109/tnnls.2022.3149997

2. Huang F, Yi P, Wang J, Li M, Peng J, Xiong X. A dynamical spatial-temporal graph neural network for traffic demand prediction. *Inf Sci* (2022) 594:286–304. doi:10.1016/j.ins.2022.02.031

3. Amara A, Taieb MAH, Aouicha MB. Cross-network representation learning for anchor users on multiplex heterogeneous social network. *Appl Soft Comput* (2022) 118:108461. doi:10.1016/j.asoc.2022.108461

4. Bing R, Yuan G, Zhu M, Meng F, Ma H, Qiao S. Heterogeneous graph neural networks analysis: A survey of techniques, evaluations and applications. *Artif Intelligence Rev* (2022) 2022:1–40. doi:10.1007/s10462-022-10375-2

5. Pham P, Nguyen LT, Nguyen NT, Kozma R, Vo B. A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. *Inf Sci* (2023) 620:105–24. doi:10.1016/j.ins.2022.11.085

6. Fu Y, Yu X, Wu Y, Ding X, Zhao S. Robust representation learning for heterogeneous attributed networks. *Inf Sci* (2023) 628:22–49. doi:10.1016/j.ins.2023.01.038

7. Han K, Wang Y, Xu C, Guo J, Xu C, Wu E, et al. GhostNets on heterogeneous devices via cheap operations. *Int J Comp Vis* (2022) 130(4):1050–69. doi:10.1007/s11263-022-01575-y

8. Chen C, Li K, Wei W, Zhou JT, Zeng Z. Hierarchical graph neural networks for few-shot learning. *IEEE Trans Circuits Syst Video Tech* (2021) 32(1):240–52. doi:10.1109/tcsvt.2021.3058098

9. Chen L, Li Y, Deng X, Liu Z, Lv M, He T. Semantic-aware network embedding via optimized random walk and paragaraph2vec. *J Comput Sci* (2022) 63:101825. doi:10.1016/j.jocs.2022.101825

10. Dong Y, Chawla NV, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; August 13-17, 2017; Halifax, NS, Canada (2017).

11. Shi Y, Gui H, Zhu Q, Kaplan L, Han J. Aspem: Embedding learning by aspects in heterogeneous information networks. In: Proceedings of the 2018 SIAM International Conference on Data Mining; May 3-5, 2018; San Diego (2018).

12. Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. *Artif Intelligence Rev* (2021) 54:6391–438. doi:10.1007/s10462-021-09975-1

13. Liu K, Zheng M, Liu Y, Yang J, Yao Y. Deep autoencoder thermography for defect detection of carbon fiber composites. *IEEE Trans Ind Inform* (2022) 2022:1. doi:10.1109/tii.2022.3172902

14. Zhang C, Song D, Huang C, Swami A, Chawla NV. Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining; August 4-8, 2019; Anchorage, AK, USA (2019).

15. Deng Z, Zhu Q, He P, Zhang D, Luo Y. A saliency detection and gram matrix transform-based convolutional neural network for image emotion classification. *Security Commun Networks* (2021) 2021:1–12. doi:10.1155/2021/6854586

16. Yu F, Liu L, Xiao L, Li K, Cai S. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* (2019) 350:108–16. doi:10.1016/j.neucom.2019.03.053

17. Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi-scroll memristive hopfield neural network. *IEEE Trans Netw Sci Eng* (2022) 10:845–58. doi:10.1109/tnse.2022.3223930

18. Wang J, Wu Y, He S, Sharma PK, Yu X, Alfarraj O, et al. Lightweight single image super-resolution convolution neural network in portable device. *KSII Trans Internet Inf Syst* (2021) 15(11).

19. Chen L, Li Y, Deng X, Liu Z, Lv M, Zhang H. Dual auto-encoder GAN-based anomaly detection for industrial control system. *Appl Sci* (2022) 12(10):4986. doi:10.3390/app12104986

20. He Y, Song Y, Li J, Ji C, Peng J, Peng H. Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management; November 3 - 7, 2019; Beijing China (2019).

21. Wang X, Zhang Y, Shi C. Hyperbolic heterogeneous information network embedding. In: Proceedings of the AAAI conference on artificial intelligence; 27 January 2019- 1 February 2019; Hawaii USA (2019).

22. Jiang J-Y, Li Z, Ju CJ-T, Wang W. Maru: Meta-context aware random walks for heterogeneous network representation learning. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management; 19th to the 23rd of October 2020 (2020).

23. Xu L, Wei X, Cao J, Yu PS. Embedding of embedding (EOE) joint embedding for coupled heterogeneous networks. In: Proceedings of the tenth ACM international conference on web search and data mining; Feb 06, 2017-Feb 10, 2017; Cambridge, United Kingdom (2017).

24. Lu Y, Shi C, Hu L, Liu Z. Relation structure-aware heterogeneous information network embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence; February 22-March 1, 2022 (2019).

25. Li B, Pi D, Lin Y, Khan IA, Cui L. Multi-source information fusion based heterogeneous network embedding. *Inf Sci* (2020) 534:53–71. doi:10.1016/j.ins.2020.05.012

26. Chen H, Yin H, Wang W, Wang H, Nguyen QVH, Li X. Pme: Projected metric embedding on heterogeneous networks for link prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining; August 19 - 23, 2018; London United Kingdom (2018).

27. Wang J, Zou Y, Lei P, Sherratt RS, Wang L, Hu F, et al. Changes in colistin resistance and mcr-1 abundance in *Escherichia coli* of animal and human origins following the ban of colistin-positive additives in China: An epidemiological comparative study. *J Internet Tech* (2020) 21(4):1161–71. doi:10.1016/S1473-3099(20)30149-3

28. Fu X, Zhang J, Meng Z, King I. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of The Web Conference; April 20 - 24, 2020; Taipei Taiwan (2020).

29. Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, et al. Heterogeneous graph attention network. In: The world wide web conference; May 13 to May 17, 2019; San Francisco, CA, USA (2019).

30. Long M, Zeng Y. Detecting iris liveness with batch normalized convolutional neural network. *Comput Mater Contin* (2019) 58(2):493–504. doi:10.32604/cmc.2019.04378

31. Zhao K, Bai T, Wu B, Wang B, Zhang Y, Yang Y, et al. Deep adversarial completion for sparse heterogeneous information network embedding. In: Proceedings of The Web Conference; April 20 - 24, 2020; Taipei Taiwan (2020).

32. Hu Z, Dong Y, Wang K, Sun Y. Heterogeneous graph transformer. In: Proceedings of the web conference; April 20 - 24, 2020; Taipei Taiwan (2020).

33. Hu B, Fang Y, Shi C. Adversarial learning on heterogeneous information networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; August 4-8, 2019; Anchorage, AK, USA (2019).

34. Hong H, Guo H, Lin Y, Yang X, Li Z, Ye J. An attention-based graph neural network for heterogeneous structural learning. In: Proceedings of the AAAI conference on artificial intelligence (2020).

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 30 May 2023
DOI 10.3389/fphy.2023.1202398

OPEN ACCESS

EDITED BY
Chunbiao Li,
Nanjing University of Information Science and Technology, China

REVIEWED BY
Jingru Sun,
Hunan University, China
Jie Jin,
Hunan University of Science and Engineering, China
Fuhong Min,
Nanjing Normal University, China

*CORRESPONDENCE
Yuan Lin,
153190675@qq.com
Fei Yu,
yufeiyfyf@csust.edu.cn

RECEIVED 08 April 2023
ACCEPTED 16 May 2023
PUBLISHED 30 May 2023

CITATION
Lin Y, Gong J, Yu F and Huang Y (2023), Current mode multi scroll chaotic oscillator based on CDTA.
Front. Phys. 11:1202398.
doi: 10.3389/fphy.2023.1202398

Current mode multi scroll chaotic oscillator based on CDTA

Yuan Lin[1]*, Junhui Gong[1], Fei Yu[2]* and Yuanyuan Huang[2]

[1]College of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan, China, [2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China

Compared to voltage mode circuits, current mode circuits have advantages such as large dynamic range, fast speed, wide frequency band, and good linearity. In recent years, the development of call flow modeling technology has been rapid and has become an important foundation for analog integrated circuits. In this paper, a current mode chaotic oscillation circuit based on current differential transconductance amplifier (CDTA) is proposed. This proposed circuit fully utilizes the advantages of current differential transconductance amplifier: a current input and output device with a large dynamic range, virtual ground at the input, extremely low input impedance, and high output impedance. The linear and non-linear parts of the proposed circuit operate in current mode, enabling a true current mode multi scroll chaotic circuit. Pspice simulation results show that the current mode chaotic circuit proposed can generate multi scroll chaotic attractors.

KEYWORDS

current mode, multi scroll, chaotic oscillation, CDTA, chaotic attractors

1 Introduction

In the past 40 years, due to the unique advantages of chaotic systems such as extreme sensitivity to initial values and parameters, ergodicity, and pseudorandomness, chaos has paid more attention to the combination of theory and practical applications, and has been widely used in fields such as secure communication [1–3], image encryption [4–8], random number generators [9–11], memristors [12–17], neural networks [18–25], and chaotic synchronization control [26–30].

The chaotic signal generated by chaotic oscillation circuits or chaotic systems is the core part of the entire chaotic communication system and has always been a research hotspot in the field of chaos [31–38]. How to generate multi scroll chaotic attractors with more complex topological structures has been widely concerned [39–41]. At present, most of the multi scroll chaotic oscillators are implemented by operational amplifiers [42–44], transconductance operational amplifiers (OTA) [45], current feedback operational amplifiers (CFOA) [46, 47] and second-generation current conveyers (CCII) [48–50]. The principles and methods for designing multi scroll chaotic oscillators based on operational amplifiers, CFOAs, and OTAs are summarized in [45]. In [46], a multi scroll chaotic oscillator was implemented using CFOA, and 3–10 scrolls from 1 kHz to 100 kHz were generated in experiments. In [49], the authors proposed a simple multi scroll chaotic oscillator implemented using a positive CCII and a negative CCII-. Circuit simulation shows that the chaotic electronic oscillator can generate more scroll chaotic attractors with higher frequencies. Because the operational amplifier belongs to the traditional voltage mode (voltage input, voltage output) circuit, the chaotic oscillator based on the operational amplifier has the problem of narrow bandwidth; OTA, CFOA

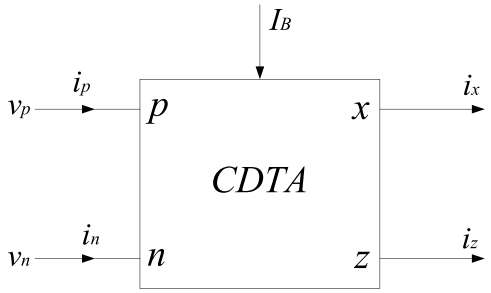


FIGURE 1
Circuit model of CDTA.

and CCII belong to the voltage and current mixed mode devices, The following problems exist in the chaotic oscillator based on OTA, CFOA and CCII:

1) Since the chaotic oscillator composed of OTA, CFOA and CCII still works in voltage mode, the output impedance is very high and changes with frequency.
2) Due to the large parasitic parameters at the input terminals of OTA, CFOA and CCII, the frequency bandwidth of the chaotic oscillator based on OTA, CFOA and CCII is not large.

Currently, analog integrated circuit designs mostly use voltage mode circuit designs [51–53]. With the development and breakthrough of various new technologies represented by the PCB process, traditional voltage mode circuits are no longer suitable for low power supply voltage design requirements due to their high impedance, high voltage gain, and high signal swing characteristics, while current mode circuits have attracted widespread interest due to their low impedance, zero or even negative voltage gain, and broadband characteristics [54–59]. Current differential transconductance amplifier (CDTA) is a current input and output device, characterized by extremely low input impedance, high output impedance, and large dynamic range.

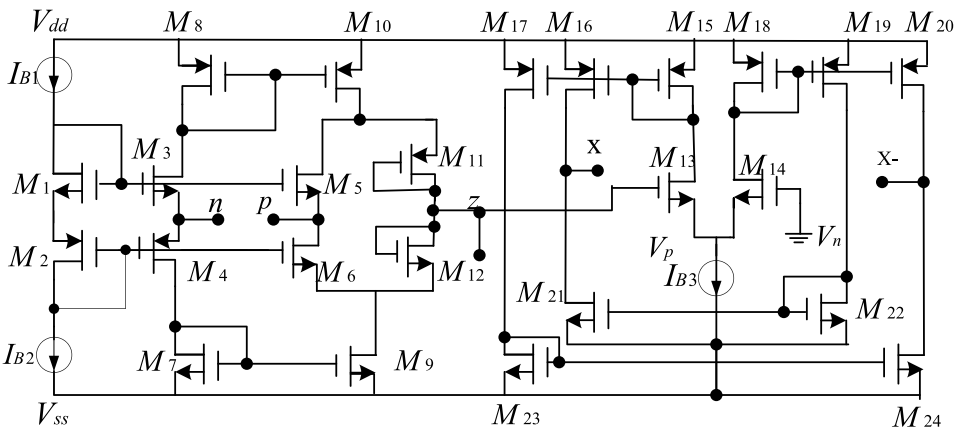


FIGURE 2
Circuit schematic diagram of CDTA.

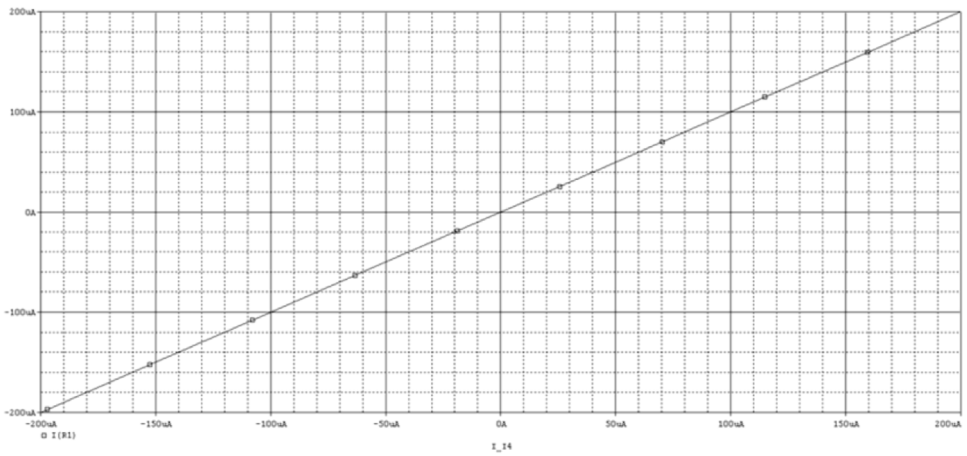


FIGURE 3
Relation curve of Iz and Ip.

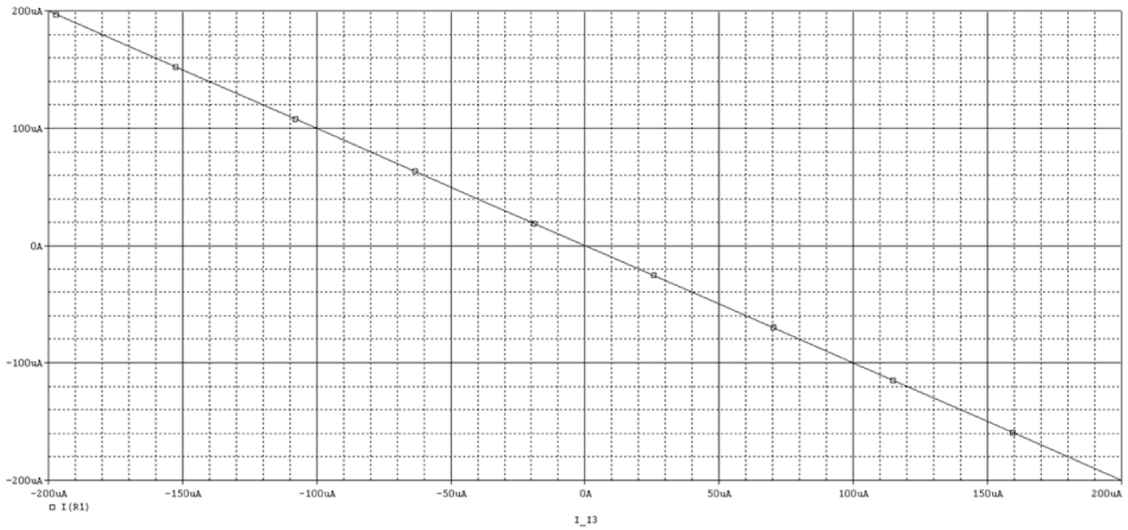


FIGURE 4
Relation curve of I_z and I_n.

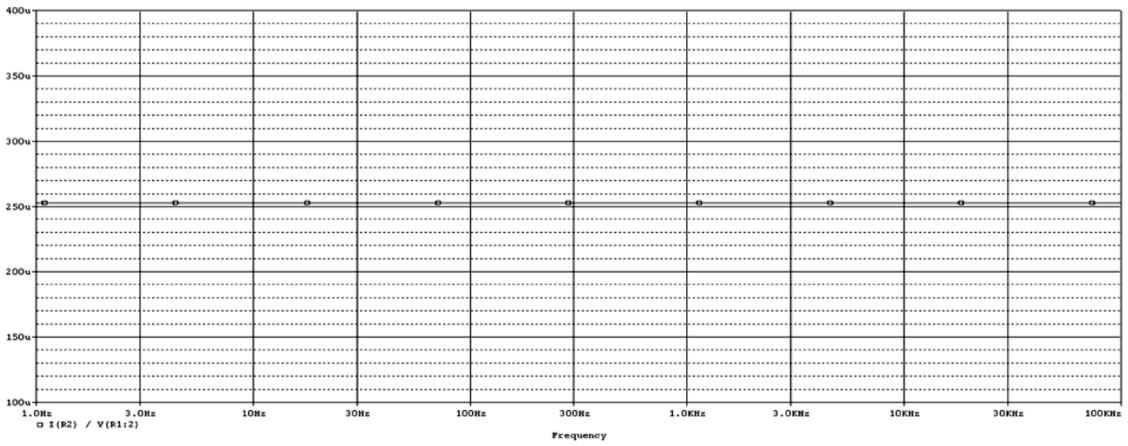


FIGURE 5
The transconductance gain of CDTA.

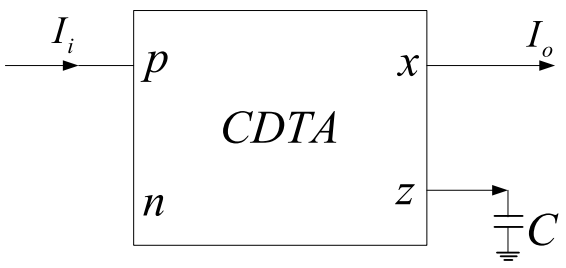


FIGURE 6
CDTA-C current mode integrator.

Compared to OTA, CFOA, and CCII, it is a true current mode device [60–63]. Two new implementations of current mode quadrature oscillators using CDTA as active components are proposed in [64]. The proposed circuit uses two grounded capacitors to achieve current controllability of the oscillation frequency. In [65], a floating decreasing and increasing memristor simulator using OTA, CDTA, and two grounded capacitors is used. Then, the proposed memristor simulator is used in the design of chaotic oscillators and adaptive learning circuits. Simulation results of a chaotic oscillator and an adaptive learning circuit verify the effectiveness of the proposed design. When it is used to form a current mode chaotic circuit, the input

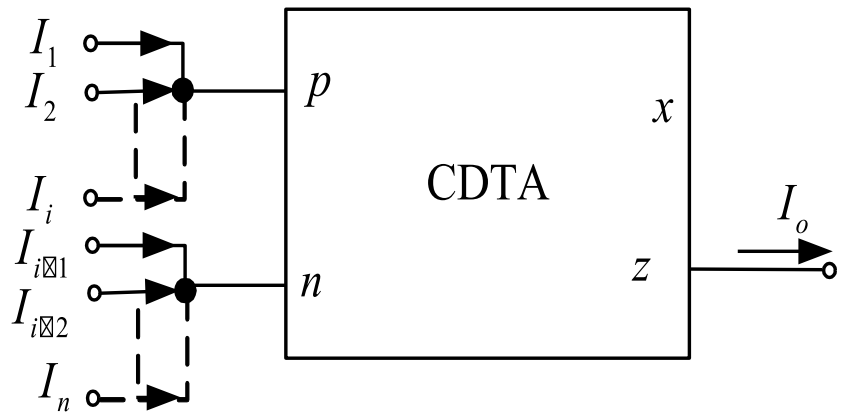


FIGURE 7
CDTA-C CDTA current mode adder and subtracter.

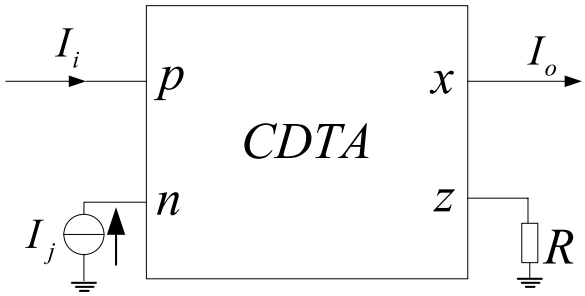


FIGURE 8
Current step function generation circuit composed of CDTA.

and output impedances have nothing to do with frequency, which can generate a larger number of chaotic attractors at high frequencies. In addition, since the input end of the CDTA is a virtual ground, the frequency parasitic parameters are small and the bandwidth is large.

In this paper, a current mode chaotic circuit based on CDTA is proposed, which can generate multi scroll chaotic attractor current signals, promote the practical application of chaos communication, chaotic neural network and other fields.

This paper is organized as follows. In Section 2, the CDTA is studied by theoretical analyses and Pspice simulation. In Section 3, the proposed current mode basic operation modules of chaotic circuit based on CDTA is studied by theoretical analyses and Pspice simulation. In Section 4, we draw our conclusions.

2 The CDTA

The CDTA is a current-input, current-output current mode device with a large dynamic range. When forming a current mode circuit, it has both low input impedance and high output impedance characteristics. Figure 1 is the circuit symbol of CDTA.

Among them, p and n are the differential current input terminals, z is the auxiliary terminal, the current at the z terminal is the difference between the input currents of p and n, the x terminal is the current output terminal, and I_B is the external bias current. The port characteristics of CDTA are as follows:

$$v_p = v_n = 0, i_z = i_p - i_n, i_x = g_m v_z = g_m Z_z i_z \quad (1)$$

where g_m is the function of the external bias current I_B, there is $g_m = f(I_B)$.

A CMOS CDTA circuit is designed, and the circuit schematic diagram is shown in Figure 2. Multiple x+ and x-ports can be expanded as needed. The transistor constitutes the current differential part, so that the z current of the auxiliary terminal is equal to the current difference between the p and n terminals. After an impedance is connected to the auxiliary terminal, the voltage v_z of the z terminal is obtained to realize the transconductance. The amplifying unit converts v_z into the current output of the x terminal.

The Pspice simulation results of CDTA are as follows: Power supply voltage V_{DD} = 2.5V, V_{SS} = −2.5 V, external control current I_b = 200 uA. If only I_p is scanned when I_n = 0A is given, the relationship curve between I_z and I_p can be obtained as shown in Figure 3; If only I_n is scanned when Ip = 0A is given, the relationship curve between I_z and I_n can be obtained as shown in Figure 4. According to these two curves, it is not difficult to see that I_z is a difference relationship with I_p and I_n. When I_p = 1A, I_n = −1A, I_b = 200 uA, the transconductance gain of CDTA gm = $I_x/V_z = I_x/(I_p-I_n)$ can be obtained, and the simulation results are shown in Figure 5.

3 The proposed current mode basic operation modules of chaotic circuit based on CDTA

The basic operation modules of the chaotic system (such as addition and subtraction, integration, etc.) and non-linear function generating circuits (such as step function, saturation function, etc.) can be easily realized by CDTA.

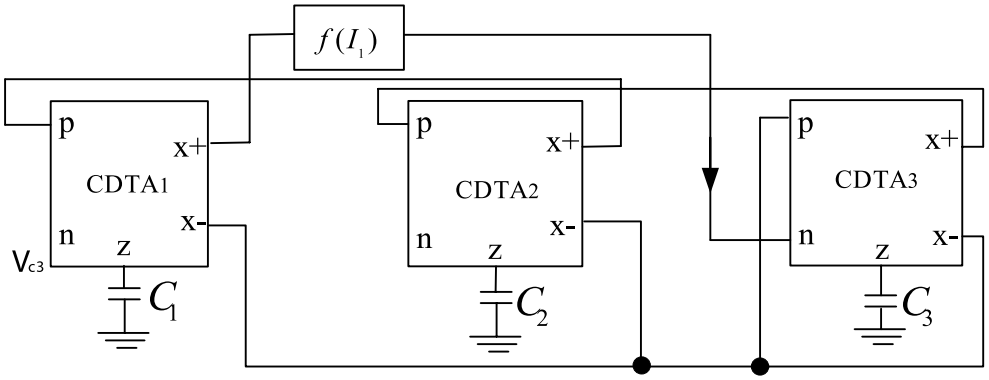


FIGURE 9
The proposed current mode multi scroll Jerk chaotic oscillation circuit.

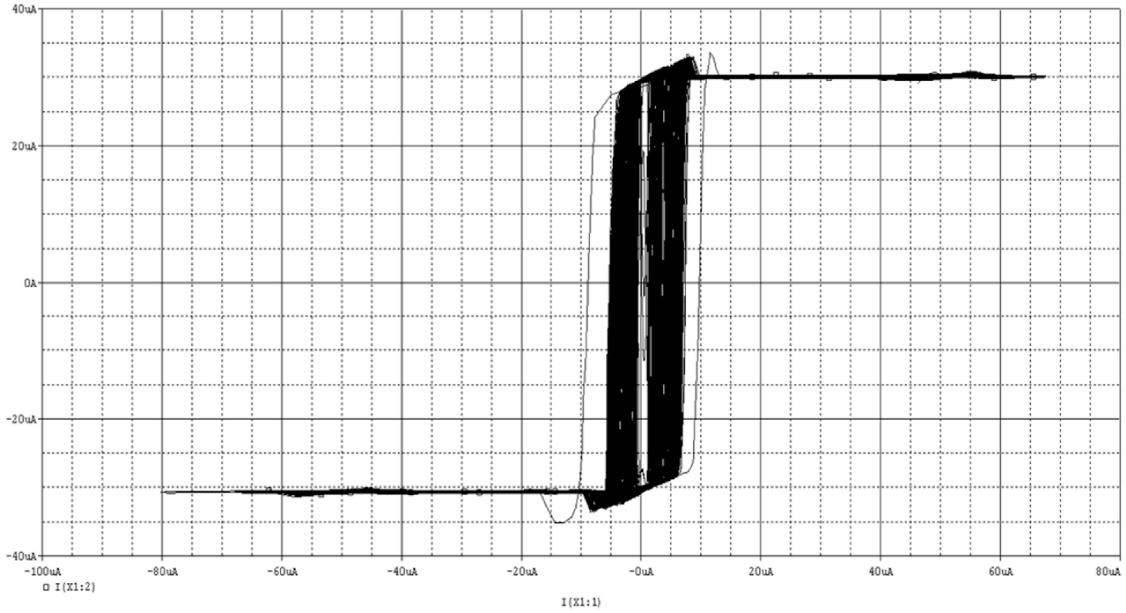


FIGURE 10
2-step wave simulation.

3.1 Integrator module

It can be seen from Equation 1 and Figure 1 that the input voltage of CDTA is zero and the input impedance is zero (the input impedance of the actual circuit is very small). In addition, the output impedance is also very high. When CDTA is used to form a current mode integrator, the input and output impedance characteristics are not destroyed. Figure 6 shows the current mode integrator composed of CDTA and capacitor, the output current expression is: $I_o = \frac{g_m}{C}\int I_i dt$.

Since the capacitor is not connected to the input and output terminals of CDTA, but is connected to the auxiliary terminal z of CDTA, the CDTA-C current mode integrator has very low input impedance and high output impedance, and has nothing to do with frequency, The input impedance of the CDTA-C current mode integrator is ideally 0, and the output impedance is the output impedance of CDTA (usually MΩ level). When implementing a chaotic circuit, the system parameters are independent of frequency, so that it can output chaos signal with a large bandwidth.

3.2 Adder module

Figure 7 shows the current mode addition and subtraction operation module composed of CDTA. Since CDTA has p and n

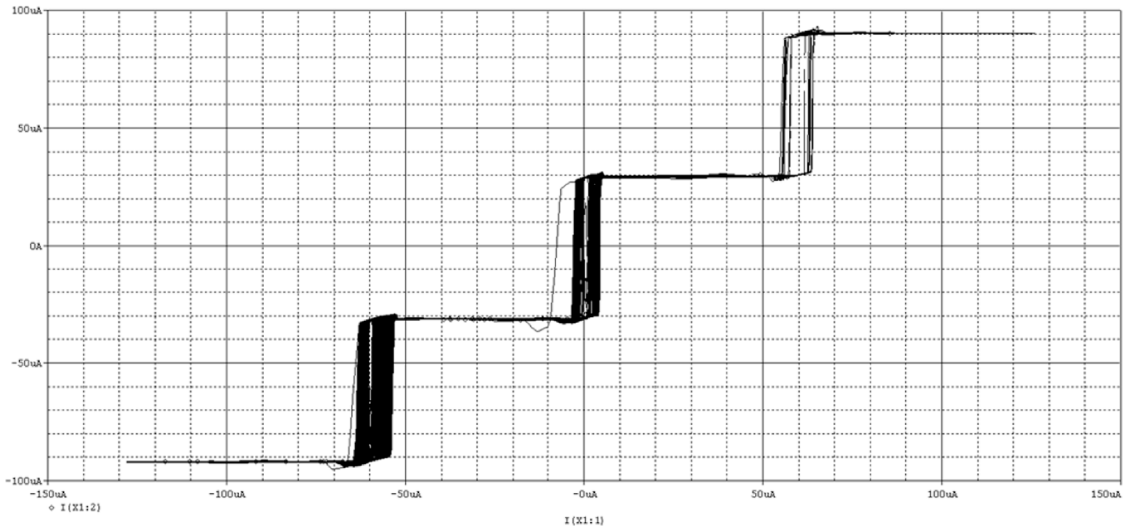


FIGURE 11
4-step wave simulation.

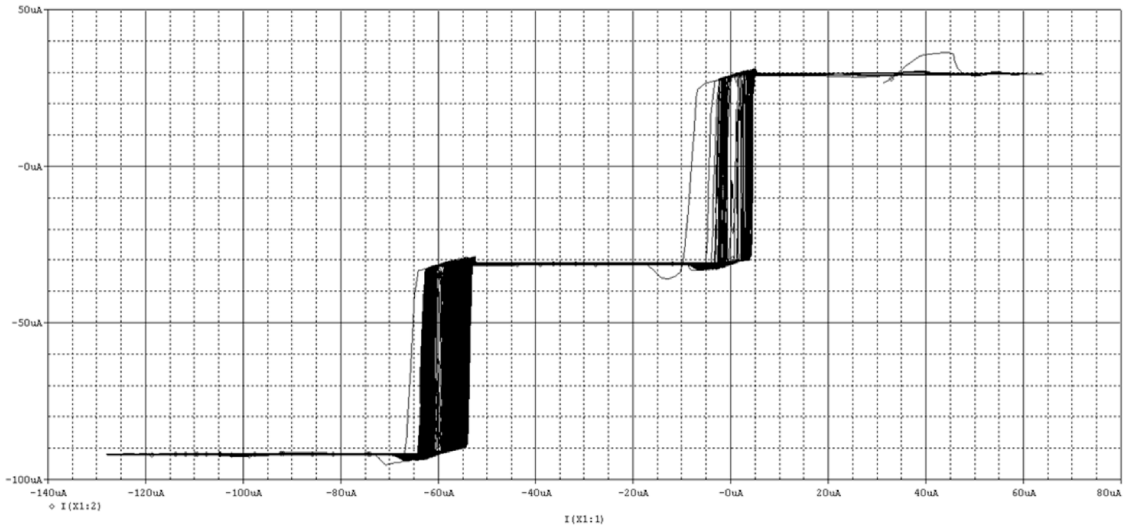


FIGURE 12
3-step wave simulation.

as differential current input terminals, current mode addition and subtraction can be easily realized. The output current expression is: $I_o = \sum_{j=1}^{i} I_j - \sum_{j=i+1}^{n} I_j$.

3.3 Step function module

The step function can be approximated as a saturation function with a sufficiently large slope. The basic unit circuit of the step function using CDTA is shown in Figure 8. The saturation current that the CDTA can achieve for a given supply voltage is denoted by $\pm |I_{sat}|$. Then the output current can be approximately expressed as: $I_o = |I_{sat}|\text{sign}(I_i - I_j)$. By connecting several basic units in parallel, the step function sequence can be obtained, and the expression is: $I_o = \sum_{j=1}^{Q} |I_{sat}| sign(I_i - I_j)$.

It can be seen from the above that the non-linear function generation circuit composed of CDTA compares the state variable current with the comparison current, and outputs the current saturation function. It can be seen that when CDTA is used to form a chaotic circuit, whether it is a linear circuit part or a non-linear circuit, all of them work in the current mode, which can realize the real current mode chaotic circuit.

In addition, due to the grounding of the input terminal (generally virtual grounding when the circuit is implemented),

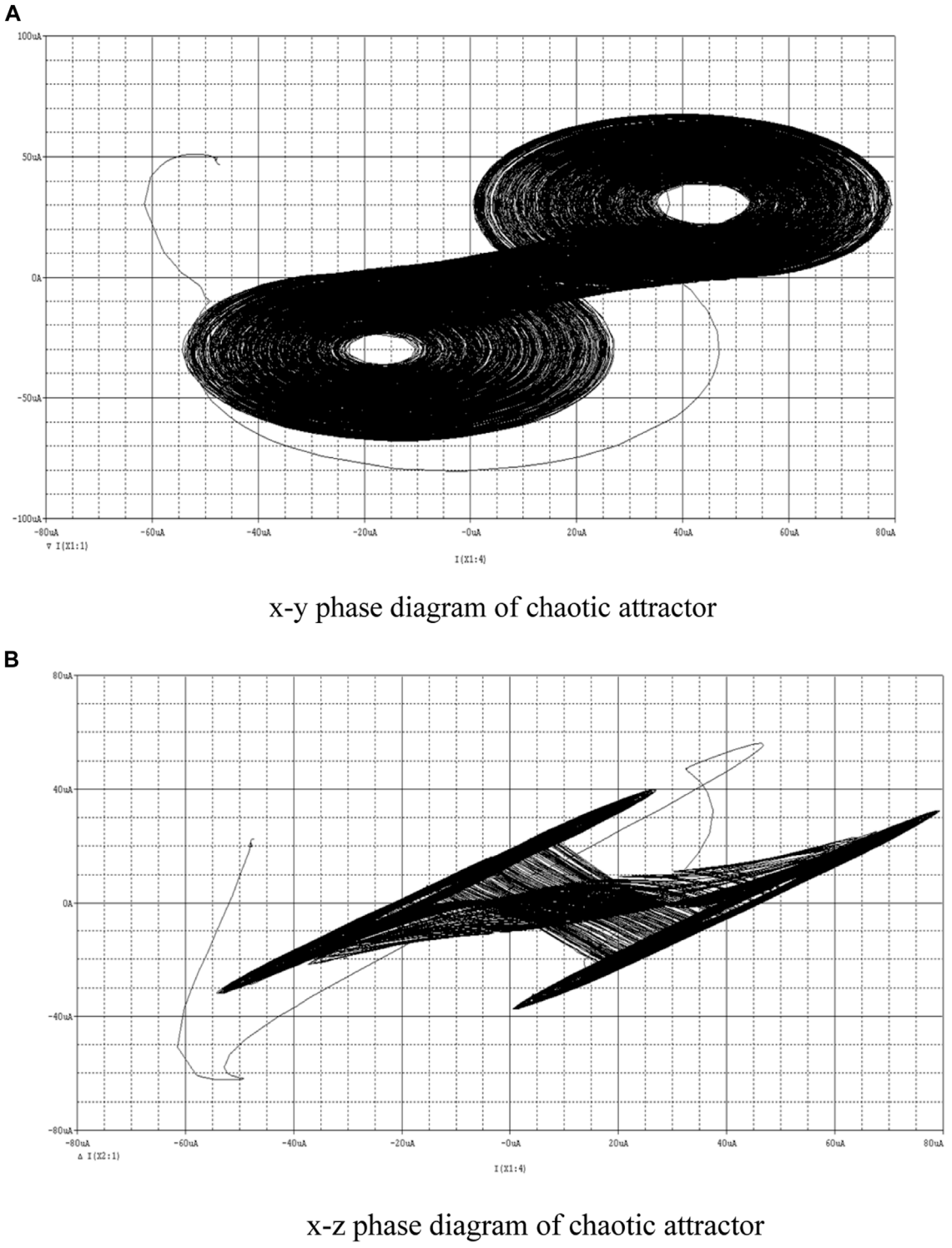


FIGURE 13
Two scroll chaotic attractor. **(A)** x-y phase diagram of chaotic attractor. **(B)** x-z phase diagram of chaotic attractor.

the parasitic parameters are small, the frequency characteristics are good, and the operating frequency bandwidth is wide.

Therefore, compared with operational amplifiers, OTA, CFOA and CCII, CDTA is more suitable for implementing current mode chaotic oscillator circuits. However, so far, there is no report on the use of CDTA to form a chaotic oscillator circuit.

3.4 The proposed CDTA-based current mode multi scroll Jerk chaotic oscillator circuit

Due to the simplicity and good recursion characteristics of the Jerk system, it has become a typical example for the study of multi

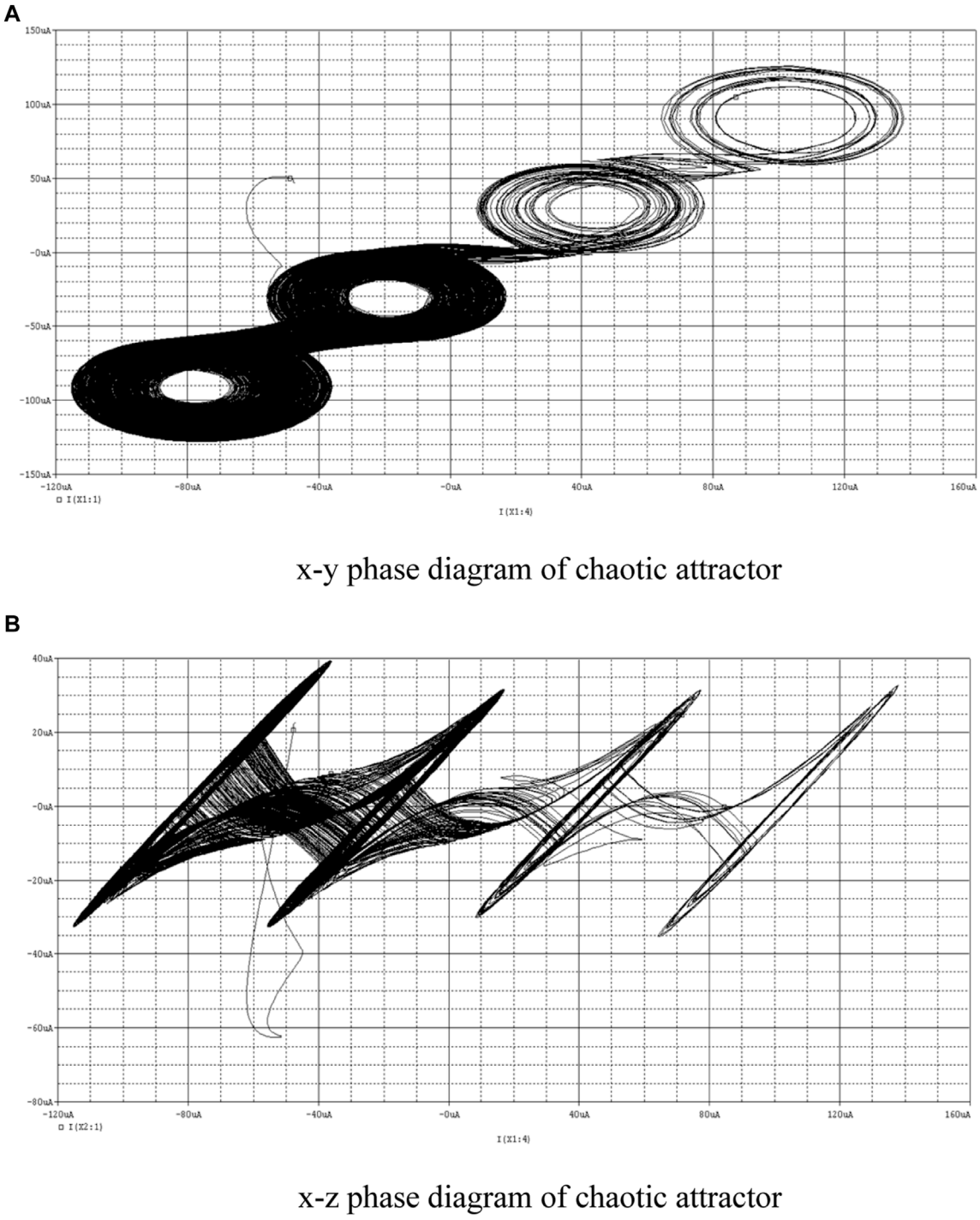


FIGURE 14
Four scroll chaotic attractor. **(A)** x-y phase diagram of chaotic attractor. **(B)** x-z phase diagram of chaotic attractor.

scroll chaotic systems. This design adopts the classic Jerk system, and its dimensionless state equation is:

$$\begin{cases} \dot{x} = y - f(y) \\ \dot{y} = z - f(z) \\ \dot{z} = -a(x + y + z) \end{cases} \tag{2}$$

The proposed CDTA-based current mode multi scroll Jerk chaotic oscillator circuit is shown in Figure 9.

The main circuit of the chaotic oscillation circuit is composed of three CDTAs and three programmable equivalent capacitances CEQ, and the circuit structure is very simple. Its non-linear function adopts a step function, and the step function generating circuit is shown in Figure 8. By

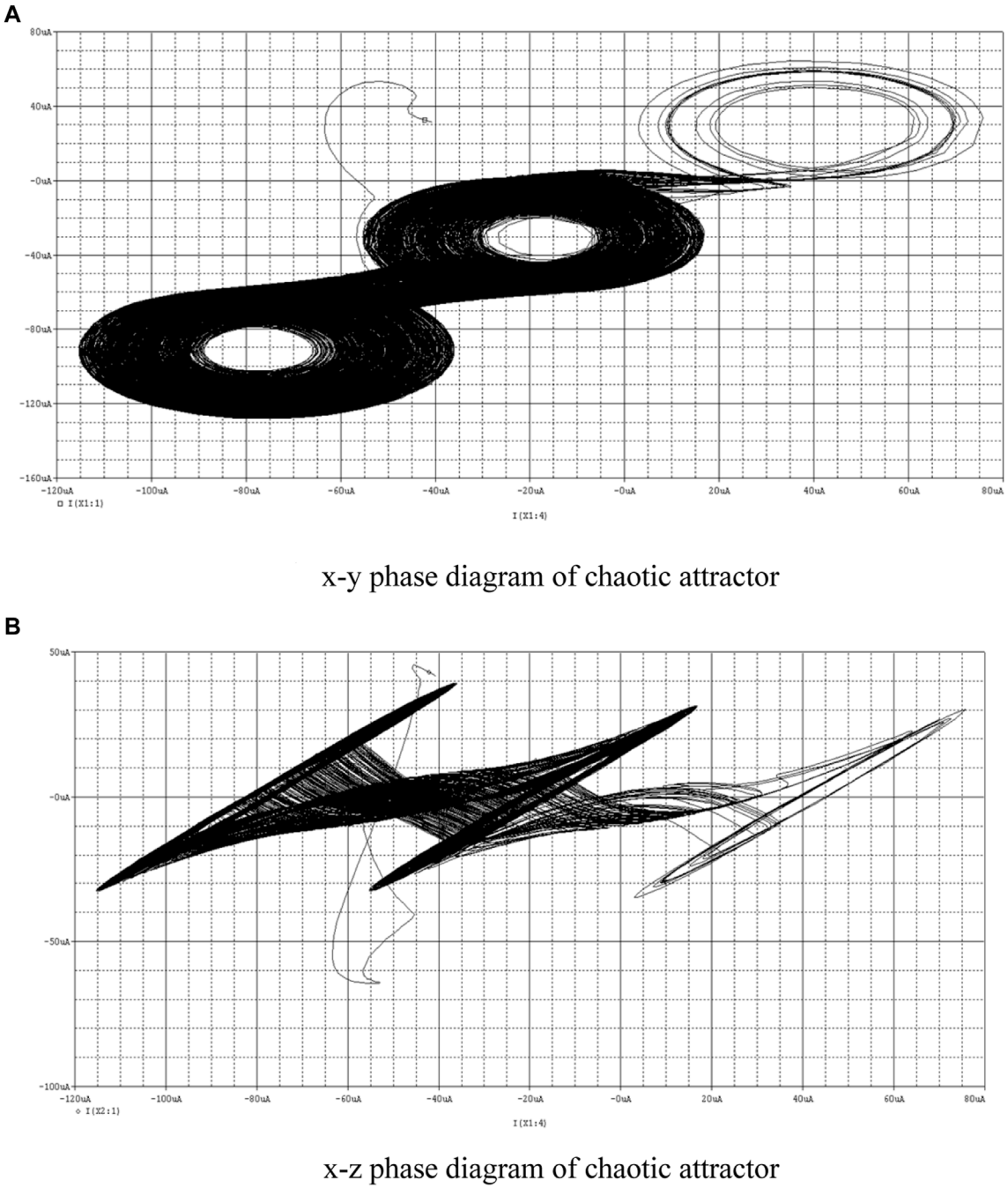


FIGURE 15
Three scroll chaotic attractor. **(A)** x-y phase diagram of chaotic attractor. **(B)** x-z phase diagram of chaotic attractor.

connecting several basic units in parallel, the step function sequence can be obtained.

$$\begin{cases} \dot{I}_1 = \dfrac{g_{m1}}{C_1}(I_2) \\ \dot{I}_2 = \dfrac{g_{m2}}{C_2}(I_3) \\ \dot{I}_3 = -\dfrac{g_{m3}}{C_3}\left(I_1 + I_2 + I_3 - f(I_1)\right) \end{cases} \tag{3}$$

It can be seen from Figures 8, 9 that the non-linear function generating circuit and the main circuit of the chaotic circuit are both current mode circuits implemented by CDTA, with good high frequency characteristics and large dynamic range, and can be designed to generate more scrolls chaotic system. And because the capacitor is connected to the auxiliary z terminal, the input and output impedances are independent of the frequency, so the chaotic system equation will not change with the frequency.

Derive the dynamic equation of the multi scroll chaotic oscillator circuit corresponding to the circuit diagram shown in Figure 9.

This is a set of third-order non-linear autonomous ordinary differential equations with the currents at the x + ports of the output terminals of CDTA1, CDTA2, and CDTA3 respectively, and as the three state variables, and the non-linear functions $f(I_2)$ and $f(I_3)$ as the current step function. Through reasonable design of transconductance and selection of capacitance, the dimensioned current signals I_1, I_2, I_3, time t are converted into signals x, y, z and dimensionless time. It can be seen that the proposed current mode multi scroll chaotic oscillator circuit shown in Figure 9 can realize the multi scroll Jerk system.

3.5 Design of even-numbered scroll Jerk system

The non-linear function adopts step function:

$$f(x) = (N - M)A1 + s(x) + \sum_{n=1}^{N} s(x - 2nA1) + \sum_{m=1}^{M} s(x + 2mA1) \tag{4}$$

Where, N, M = 1,2,3,4, etc. Especially when N = M, there is

$$f(x) = s(x) + \sum_{n=1}^{N} s(x - 2nA1) + \sum_{m=1}^{N} s(x + 2mA1) \tag{5}$$

Where A = A1 is the saturation value of the saturation function, and the number of scrolls can be generated is 2 *N+2. When N = 0, $f(x) = s(x)$, From this, 2-step waves can be obtained, and the simulation is shown in Figure 10. It can be seen that the 2-step function can be achieved, which can generate two saddle focal equilibrium points with two indicators, and can achieve two scrolls.

From the above figures, we can get A = A1 = 30uA, take N = 1, then we can get 4-step waves, and the simulation is shown in Figure 11. It can be seen that the 4-step function can be achieved, which can generate four saddle focal equilibrium points with two indicators, and can achieve four scrolls.

3.6 Design of odd-numbered scroll Jerk system

Using Eq. (4), an odd number of scrolls can be generated, and the number of scrolls is N + M+2. Now take the generation of three scroll numbers as an example. Let N = 1, M = 0, and the scroll number be N + M+2 = 1 + 0+2 = 3. Then Eq. (4) becomes Eq. (6)

$$f(x) = A1 + s(x) + s(x - 2A1) \tag{6}$$

From this, 3-step waves can be obtained, and the simulation is shown in Figure 12. It can be seen that the 3-step function can be achieved, which can generate three saddle focal equilibrium points with two indicators, and can achieve three scrolls.

3.7 Simulation of multi scroll chaotic circuit

According to the expression Eq. (4) of the non-linear function f(x), when N and M take different values, different step waves will be generated, which will affect the number of scroll generated. When N = M and N = 0, two scroll attractors are generated, as shown in Figure 13.

When N = M, N = 1, four scroll attractors are generated, as shown in Figure 14.

When N = 1, M = 0, three scroll attractors are generated, as shown in Figure 15.

As shown in these figures, the proposed CDTA-based current mode multi scroll Jerk chaotic oscillator circuit can display a theoretical number of scrolls in both the x-y and *x-z* directions. The experimental results are consistent with the theoretical results.

4 Conclusion

The circuit structure of the multi scroll chaotic oscillator based on CDTA proposed in this paper is simple, the main circuit does not contain passive resistance elements, has low input impedance, high output impedance, and the input and output impedance are independent of frequency, and the dynamic range is large, so that the chaotic oscillator can generate more scrolls, and the signal is not distorted.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

YL: Conceptualization, Methodology, Visualization, Project administration, Supervision, Writing–Review and Editing. JG: Methodology, Investigation, Formal analysis, Supervision. FY: Methodology, Investigation, Formal analysis, Writing–Original Draft. YH: Investigation, Formal analysis, Supervision. All authors contributed to the article and approved the submitted version.

Funding

This work is supported by the Scientific Research Fund of Hunan Provincial Education Department under Grants 20k036, 21B0345 and 19C0083, and by the Hunan Provincial Natural Science Foundation of China under Grant 2022JJ30624, and by the Postgraduate Training Innovation Base Construction Project of Hunan Province under Grant 2020-172-48.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Yu F, Yu Q, Chen H, Kong X, Mokbel AAM, Cai S, et al. Dynamic analysis and audio encryption application in IoT of a multi scroll fractional-order memristive Hopfield neural network. *Fractal and Fractional* (2022) 6(7):370. doi:10.3390/fractalfract6070370

2. Zhou L, Tan F. A chaotic secure communication scheme based on synchronization of double-layered and multiple complex networks. *Nonlinear Dyn* (2019) 96(2):869–83. doi:10.1007/s11071-019-04828-7

3. Gokyildirim A, Kocamaz UE, Uyaroglu Y, Calgan H. A novel five-term 3D chaotic system with cubic nonlinearity and its microcontroller-based secure communication implementation. *AEU-International J Elect Commun* (2023) 160:154497. doi:10.1016/j.aeue.2022.154497

4. Yu F, Xu S, Xiao X, Yao W, Huang Y, Cai S, et al. Dynamics analysis, FPGA realization and image encryption application of a 5D memristive exponential hyperchaotic system. *Integration* (2023) 90:58–70. doi:10.1016/j.vlsi.2023.01.006

5. Zhu Y, Wang C, Sun J, Yu F. A chaotic image encryption method based on the artificial fish swarms algorithm and the DNA coding. *Mathematics* (2023) 11:767. doi:10.3390/math11030767

6. Yu F, Kong X, Mokbel AAM, Yao W, Cai S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive hopfield neural network with a novel local active memeristor. *IEEE Trans Circuits Syst Express Briefs* (2023) 70(1):326–30. doi:10.1109/tcsii.2022.3218468

7. Xie Z, Sun J, Tang Y, Tang X, Simpson O, Sun Y. A K-svd based compressive sensing method for visual chaotic image encryption. *Mathematics* (2023) 11(7):1658. doi:10.3390/math11071658

8. Gao X, Sun B, Cao Y, Banerjee S, Mou J. A color image encryption algorithm based on hyperchaotic map and DNA mutation. *Chin Phys B* (2023) 32:030501. doi:10.1088/1674-1056/ac8cdf

9. Yu F, Zhang Z, Shen H, Huang Y, Cai S, Jin J, et al. Design and FPGA implementation of a pseudo-random number generator based on a Hopfield neural network under electromagnetic radiation. *Front Phys* (2021) 9:690651. doi:10.3389/fphy.2021.690651

10. Yang Z, Liu Y, Wu Y, Qi Y, Ren F. A high speed pseudo-random bit generator driven by 2D-discrete hyperchaos. *Chaos, Solitons & Fractals* (2023) 167:113039. doi:10.1016/j.chaos.2022.113039

11. Yu F, Zhang Z, Shen H, Huang Y, Cai S, Du S. FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient. *Chin Phys B* (2022) 31(2):020505. doi:10.1088/1674-1056/ac3cb2

12. Min F, Xue L. Routes toward chaos in a memristor-based Shinriki circuit. *Chaos* (2023) 33(2):023122. doi:10.1063/5.0126900

13. Lin H, Wang C, Xu C, Zhang X, Iu HHC. A memristive synapse control method to generate diversified multi-structure chaotic attractors. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2022) 42:942–55. doi:10.1109/TCAD.2022.3186516

14. Min F, Cheng Y, Lu L, Li X. Extreme multistability and antimonotonicity in a Shinriki oscillator with two flux-controlled memristors. *Int J Bifurcation Chaos* (2021) 31(11):2150167. doi:10.1142/s0218127421501674

15. Chen Y, Mou J, Jahanshahi H, Wang Z, Cao Y. A new mix chaotic circuit based on memristor-memcapacitor. *The Eur Phys J Plus* (2023) 138(1):78. doi:10.1140/epjp/s13360-023-03699-7

16. Ma M, Lu Y, Li Z, Sun Y, Wang C. Multistability and phase synchronization of rulkov neurons coupled with a locally active discrete memristor. *Fractal and Fractional* (2023) 7(1):82. doi:10.3390/fractalfract7010082

17. Yu F, Zhang W, Xiao X, Yao W, Cai S, Zhang J, et al. Dynamic analysis and FPGA implementation of a new, simple 5D memristive hyperchaotic Sprott-C system. *Mathematics* (2023) 11(3):701. doi:10.3390/math11030701

18. Lin H, Wang C, Yu F, Sun J, Du S, Deng Z, et al. A review of chaotic systems based on memristive hopfield neural networks. *Mathematics* (2023) 11(6):1369. doi:10.3390/math11061369

19. Xu Q, Chen X, Chen B, Wu H, Li Z, Han B. Dynamical analysis of an improved FitzHugh-Nagumo neuron model with multiplier-free implementation. *Nonlinear Dyn* (2023) 111(9):8737–49. doi:10.1007/s11071-023-08274-4

20. Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi scroll memristive hopfield neural network. *IEEE Trans Netw Sci Eng* (2023) 10(2):845–58. doi:10.1109/tnse.2022.3223930

21. Wan Q, Li F, Chen S, Yang Q. Symmetric multi-scroll attractors in magnetized Hopfield neural network under pulse controlled memristor and pulse current stimulation. *Chaos, Solitons & Fractals* (2023) 169:113259. doi:10.1016/j.chaos.2023.113259

22. Xu Q, Tong L, Ding S, Wu H, Huang L, Chen B. Extreme multistability and phase synchronization in a heterogeneous bi-neuron Rulkov network with memristive electromagnetic induction. *Cogn Neurodynamics* (2022). doi:10.1007/s11571-022-09866-3

23. Yu F, Liu L, Xiao L, Li K, Cai S. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* (2019) 350:108–16. doi:10.1016/j.neucom.2019.03.053

24. Jin J, Zhu J, Zhao L, Chen L, Chen L, Gong J. A robust predefined-time convergence zeroing neural network for dynamic matrix inversion. *IEEE Trans Cybernetics* (2022) 53:3887–900. doi:10.1109/TCYB.2022.3179312

25. Sun J, Wang Y, Liu P, Wen S. Memristor-based neural network circuit with multimode generalization and differentiation on pavlov associative memory. *IEEE Trans Cybernetics* (2023) 53(5):3351–62. doi:10.1109/tcyb.2022.3200751

26. Chen W, Jin J, Chen C, Fei Y, Wang C. A disturbance suppression zeroing neural network for robust synchronization of chaotic systems and its FPGA implementation. *Int J Bifurcation Chaos* (2022) 32(14):2250210. doi:10.1142/s0218127422502108

27. Yao W, Wang CH, Sun YC, Zhou C. Robust multimode function synchronization of memristive neural networks with parameter perturbations and time-varying delays. *IEEE Trans Syst Man, Cybernetics: Syst* (2022) 52(1):260–74. doi:10.1109/tsmc.2020.2997930

28. Ma M, Xie XH, Yang Y, Li ZJ, Sun YC. Synchronization coexistence in a rulkov neural network based on locally active discrete memristor. *Chin Phys. B* (2023) 32:058701. doi:10.1088/1674-1056/acb9f7

29. Tan F, Zhou L, Lu J, Quan H, Liu K. Adaptive quantitative control for finite time synchronization among multiplex switched nonlinear coupling complex networks. *Eur J Control* (2023) 70:100764. doi:10.1016/j.ejcon.2022.100764

30. Yao W, Wang CH, Sun YC, Gong SQ, Lin HR. Event-triggered control for robust exponential synchronization of inertial memristive neural networks under parameter disturbance. *Neural Networks* (2023) 16:467–80. doi:10.1016/j.neunet.2023.04.024

31. Wang N, Xu D, Kuznetsov NV, Han B, Chen M, Xu Q. Experimental observation of hidden Chua's attractor. *Chaos, Solitons and Fractals* (2023) 170:113427. doi:10.1016/j.chaos.2023.113427

32. Sun J, Wang Y, Liu P, Wen S. Memristor-based circuit design of PAD emotional space and its application in mood congruity. *IEEE Internet Things J* (2023) 1. doi:10.1109/JIOT.2023.3267778

33. Deng Z, Wang C, Lin H, Sun Y. A memristive spiking neural network circuit with selective supervised attention algorithm. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2022) 1. doi:10.1109/TCAD.2022.3228896

34. Shen H, Yu F, Kong X, Mokbel AAM, Wang C, Cai S. Dynamics study on the effect of memristive autapse distribution on Hopfield neural network. *Chaos* (2022) 32(8):083133. doi:10.1063/5.0099466

35. Liao M, Wang C, Sun Y, Lin H, Xu C. Memristor-based affective associative memory neural network circuit with emotional gradual processes. *Neural Comput Appl* (2022) 34(16):13667–82. doi:10.1007/s00521-022-07170-z

36. Jin J, Chen W, Chen C, Chen L, Tang Z, Chen L, et al. A predefined fixed-time convergence ZNN and its applications to time-varying quadratic programming solving and dual-arm manipulator cooperative trajectory tracking. *IEEE Trans Ind Inform* (2022) 1–12. doi:10.1109/TII.2022.3220873

37. Shen H, Yu F, Wang C, Sun J, Cai S. Firing mechanism based on single memristive neuron and double memristive coupled neurons. *Nonlinear Dyn* (2022) 110:3807–22. doi:10.1007/s11071-022-07812-w

38. Jin J, Chen W, Ouyang A, Liu H. Towards fuzzy activation function activated zeroing neural network for currents computing. *IEEE Trans Circuits Syst Express Briefs* (2023) 1. doi:10.1109/TCSII.2023.3269060

39. Lin H, Wang C, Sun Y, Wang T. Generating n-scroll chaotic attractors from a memristor-based magnetized hopfield neural network. *IEEE Trans Circuits Syst Express Briefs* (2023) 70(1):311–5. doi:10.1109/tcsii.2022.3212394

40. Ben Slimane N, Bouallegue K, Machhout M. Designing a multi scroll chaotic system by operating Logistic map with fractal process. *Nonlinear Dyn* (2017) 88:1655–75. doi:10.1007/s11071-017-3337-0

41. Yu F, Shen H, Zhang Z, Huang Y, Cai S, Du S. A new multi scroll Chua's circuit with composite hyperbolic tangent-cubic nonlinearity: Complex dynamics, Hardware implementation and Image encryption application. *Integration* (2021) 81:71–83. doi:10.1016/j.vlsi.2021.05.011

42. Chen D, Sun Z, Ma X, Chen L. Circuit implementation and model of a new multi-scroll chaotic system. *Int J Circuit Theor Appl* (2014) 42(4):407–24. doi:10.1002/cta.1860

43. Chao-Xia Z, Si-Min Y. Design and implementation of a novel multi scroll chaotic system. *Chin Phys B* (2009) 18(1):119–29. doi:10.1088/1674-1056/18/1/019

44. Chen L, Pan W, Wu R, Wang K, He Y. Generation and circuit implementation of fractional-order multi scroll attractors. *Chaos, Solitons & Fractals* (2016) 85:22–31. doi:10.1016/j.chaos.2016.01.016

45. Munoz-Pacheco JM, Campos-Lopez W, Tlelo-Cuautle E, Sanchez-Lo C. OpAmp-CFOA-and OTA-based configurations to design multi scroll chaotic oscillators. *Trends Appl Sci Res* (2012) 7(2):168–74. doi:10.3923/tasr.2012.168.174

46. Munoz-Pacheco JM, Tlelo-Cuautle E, Toxqui-Toxqui I, Sánchez-López C, Trejo-Guerra R. Frequency limitations in generating multi scroll chaotic attractors using CFOAs. *Int J Elect* (2014) 101(11):1559–69. doi:10.1080/00207217.2014.880999

47. Wang CH, Yin JW, Lin Y. Design and realization of grid multi scroll chaotic circuit based on current conveyers. *Acta Phys Sin* (2012) 61(21):210507. doi:10.7498/aps.61.210507

48. Zhang X, Wang C. A novel multi-attractor period multi scroll chaotic integrated circuit based on CMOS wide adjustable CCCII. *IEEE Access* (2019) 7:16336–50. doi:10.1109/access.2019.2894853

49. Lin Y, Wang CH, Xu H. Grid multi scroll chaotic attractors in hybrid image encryption algorithm based on current conveyor. *Acta Phys Sin* (2012) 61(24):240503. doi:10.7498/aps.61.240503

50. Zuo T, Sun K-H, Ai X-X, Wang H-H. Grid multi scroll chaotic circuit based on the second generation current conveyers. *Acta Phys Sin* (2014) 63(8):080501. doi:10.7498/aps.63.080501

51. Yu F. A low-voltage and low-power 3-GHz CMOS LC VCO for S-band wireless applications. *Wireless Personal Communications* (2014) 78(2):905–14. doi:10.1007/s11277-014-1791-2

52. Yu F, Tang Q, Wang W, Wu H. A 2.7 GHz low-phase-noise LC-QVCO using the gate-modulated coupling technique. *Wireless Personal Commun* (2016) 86:671–81. doi:10.1007/s11277-015-2951-8

53. Yu F, Gao L, Liu L, Qian S, Cai S, Song Y. A 1 V, 0.53 ns, 59 μW current comparator using standard 0.18 μm CMOS technology. *Wireless Personal Commun* (2020) 111(2):843–51. doi:10.1007/s11277-019-06888-9

54. Chen JH, Chau KT, Chan CC. Analysis of chaos in current-mode-controlled DC drive systems. *IEEE Trans Ind Elect* (2000) 47(1):67–76. doi:10.1109/41.824127

55. Chang CY, Zhao X, Yang F, Wu CE. Bifurcation and chaos in high-frequency peak current mode Buck converter. *Chin Phys B* (2016) 25(7):070504. doi:10.1088/1674-1056/25/7/070504

56. Unuk T, Arslanalp R, Tez S. Design of Current-Mode versatile Multi-Input analog multiplier topology. *AEU-International J Elect Commun* (2023) 160:154493. doi:10.1016/j.aeue.2022.154493

57. Ahmadi S, Mahmoudi A, Maddipatla D, Bazuin BJ, Azhari SJ, Atashbar MZ. A current mode instrumentation amplifier with high common-mode rejection ratio designed using a novel fully differential second-generation current conveyor. *SN Appl Sci* (2023) 5(1):34. doi:10.1007/s42452-022-05247-x

58. Lberni A, Marktani MA, Ahaitouf A, Ahaitouf A. An efficient optimisation-based design of current conveyor performances. *Int J Comp Aided Eng Tech* (2023) 18(1-3):167–80. doi:10.1504/ijcaet.2023.127794

59. Ariando D, Mandal S. A pulsed current-mode class-D low-voltage high-bandwidth power amplifier for portable NMR systems. *J Magn Reson* (2023) 348:107367. doi:10.1016/j.jmr.2023.107367

60. Keskin AÜ, Biolek D. Current mode quadrature oscillator using current differencing transconductance amplifiers (CDTA). *IEE Proceedings-Circuits, Devices Syst* (2006) 153(3):214–8. doi:10.1049/ip-cds:20050304

61. Jin J, Wang C. Single CDTA-based current-mode quadrature oscillator. *AEU-International J Elect Commun* (2012) 66(11):933–6. doi:10.1016/j.aeue.2012.03.018

62. Kacar F, Kuntman HH. A new, improved CMOS realization of CDTA and its filter applications. *Turkish J Electr Eng Comp Sci* (2011) 19(4):631–42. doi:10.3906/elk-1003-467

63. Jaikla W, Siripruchyanun M, Bajer J, Biolek D. A simple current-mode quadrature oscillator using single CDTA. *Radioengineering* (2008) 17(4):33–40. doi:10.1070/QE2008v038n12ABEH013799

64. Lahiri A. New current-mode quadrature oscillators using CDTA. *IEICE Elect Express* (2009) 6(3):135–40. doi:10.1587/elex.6.135

65. Singh A, Rai SK. OTA and CDTA-based new memristor-less meminductor emulators and their applications. *J Comput Elect* (2022) 21(4):1026–37. doi:10.1007/s10825-022-01889-7

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 09 August 2023
DOI 10.3389/fphy.2023.1239257

OPEN ACCESS

EDITED BY
Fei Yu,
Changsha University of Science and Technology, China

REVIEWED BY
Chengyi Xia,
Tiangong University, China
Fei Tan,
Xiangtan University, China

*CORRESPONDENCE
Hui Tian,
tianhui@bupt.edu.cn

RECEIVED 13 June 2023
ACCEPTED 26 July 2023
PUBLISHED 09 August 2023

CITATION
Tian Y, Tian H, Zhu X and Cui Q (2023), Dynamics analysis of the two-layer complex propagation network with individual heterogeneous decreased behavior.
Front. Phys. 11:1239257.
doi: 10.3389/fphy.2023.1239257

Dynamics analysis of the two-layer complex propagation network with individual heterogeneous decreased behavior

Yang Tian[1], Hui Tian[1]*, Xuzhen Zhu[1] and Qimei Cui[2]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, [2]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China

Due to the differences in society stratum, personal profession, and social acceptability, information propagation can be impacted by the contact capabilities of individuals. Importantly, we found that with the changes in individual psychology, their response to a phenomenon will gradually weaken. This phenomenon is called heterogeneous decreased behavior and applied in the fields of economics, sociology, and ecology. In the social network, people show a gradually decreasing degree of interest for information, named individual heterogeneous decreased behavior (IHDB). We structure a two-layer network model to describe individual behavioral contact and propose a threshold function to represent IHDB. Meanwhile, we use partition theory to explain the information propagation mechanism. Through experiments, it is demonstrated that there is a continuous information outbreak in the ultimate adoption size when individuals exhibit a positive IHDB. However, when individuals exhibit a passive IHDB, there is a discontinuous information outbreak in the ultimate adoption size. Eventually, our experiments show that the theoretical analysis coincides with the results of the simulations.

KEYWORDS

information propagation, two-layer networks, individual behavioral contact, individual heterogeneous decreased behavior, adoption threshold function

1 Introduction

1.1 Research background

With the persistent exploration of the information propagation mechanism, researchers have discovered the influencing factors of different individual behaviors on information outbreak [1–9]. Individual behavior is mainly affected by individual psychology. For example, in information propagation, some people show positive adoption, while others show negative adoption [10, 11]. These behaviors are not only applied in information propagation but also in other fields, such as economics, ecology, and medicine. By investigating abundant literature works, we found that there is a latent behavioral phenomenon in multiple fields. Particularly, in investment, when the investment quota is increased, the additional income brought by an increase in one unit of investment will decrease [12–15]. In agricultural production, when the number of chemical fertilizers

increases, the additional yield that can be brought by a unit of chemical fertilizers will decrease [16, 17]. In medicine, when the dose of drugs increases, the treatment effect of drugs on patients will decrease [18, 19]. In the field of education, the additional educational effects that can be brought about by one unit of educational resources will be reduced [20, 21]. In summary, this behavior is called a decreasing behavior which is applied in multiple fields. For propagation dynamics, the paper aims to explore the impact mode of decreasing behavior on information propagation.

Based on our survey, we found that the phenomenon of decreased behaviors also exists in information propagation. For example, for advertising effect, at the beginning, advertising may attract people's attention, but over time, the interest degree of people in advertising will gradually decrease [22, 23]. For media marketing, people's interest degree in the same type of content will also decrease [24, 25]. For teaching, learning motivation of students for the same knowledge will decrease [26, 27]. In summary, in information propagation, if the same information is repeatedly received, the interest degree of people will decrease, which is closely related to individual psychology [28]. Therefore, this paper defines the phenomenon as an individual heterogeneous decreasing behavior, which is called IHDB, to explore its impact on information propagation mechanisms.

1.2 Main contributions

In this study, we found that people can only closely contact with a small number of friends due to the differences in society stratum, personal profession, and social acceptability [29–31]. For instance, a user can face difficulty to connect with all of his/her friends in a short period of time. Furthermore, several social media platforms, such as Twitter, Instagram, Tinder, and WhatsApp, are used frequently by many individuals [32–34]. Therefore, this paper considers the two-layer contacted network to explore the individual contact capacity which needs to be considered to analyze the information propagation mechanism.

More importantly, information propagation will be influenced by the individual adoption behaviors [35, 36]. According to the explanation given in Section 1.1, we found that the individual interest in behavior adoption is closely related to the proportion of neighbors who adopt the behavior. The individual heterogeneous decreasing behavior (IHDB) illustrates that when more and more neighbors transmit a same piece of information to the individual, the passion degree of the individual to receive the information gradually decreases.

The advantages of the IHDB compared to other behaviors are as follows: i) More accurate simulation of reality: IHDB reflects different responses and response degrees of the individual in information acceptance. This is closer to the real situation because the response of a person is gradually weakened in each receiving information, which will affect the information propagation process. ii) More accurate prediction of information propagation: By considering IHDB, you can more accurately predict the spread trend of information in the networks. With the proportion of neighbors in adoption state changes, the individual behavior spreads more passively or slowly. This difference can be reflected in the model, thereby providing more accurate diffusion forecasts. iii) Planning more targeted propagation strategies: The consideration of IHDB can help decision-makers formulate more targeted information dissemination strategies. By understanding the mental characteristics and behaviors of people, we can make better customized strategies for different types of information and improve the effect of information propagation. iv) Research and identification of crucial factors: Simulation of IHDB can help researchers identify the impact factors. By observing the IHDB in the individual, it is possible to determine the factors that play a key role in the information dissemination network so as to intervene or use these factors to promote information propagation.

Enlightened by the aforementioned overview, we consider individual limited contact capacity, capture the IHDB, and define a non-rule trapezoidal-like threshold function to illustrate the IHDB feature. The non-rule trapezoidal-like threshold function displays a slow and non-linear rising, subsequently maintaining a horizontal line. In addition, we provide a partition theory to analyze the information propagation mechanism based on limited contact and IHDB. Finally, through the theoretical analysis and experimental simulation, this paper reveals the information propagation mechanism.

The remainder of this paper is organized as follows: in Section 2, we put forward a probability adoption threshold model for information propagation on the two-layer contacted network. Section 3 exhibits a division theory based on limited contact and IHDB function model. Section 4 uses simulations and theoretical assessments to validate the information propagation mechanism. Finally, Section 5 summarizes the results, and Section 6 reports the conclusion.

2 Information propagation model

2.1 Related work

A two-layered network model with N nodes is set up to explore the impacts of individual contact capability and IHDB characteristics on information propagation. Layers A and B represent two independent social network layers. Since the nodes connecting the various tiered networks are one-to-one correspondences, the same node is present in each layer. Then, the degree vector of node i is represented by $\overrightarrow{k_i} = (k_i^A, k_i^B)$, where k_i^X represents a node degree. The degree distribution $P(\vec{k})$ of the network is represented by the degree vector $\vec{k}$. Furthermore, the degree distributions $P_X(k^X)$ of $P(\vec{k})$ can be broken down in accordance with the uncorrelated feature. Considering the independence of $P_A(k^A)$ and $P_B(k^B)$, $P(\vec{k}) = P_A(k^A)P_B(k^B)$.

This paper establishes a two-layer social network model to explore a novel information propagation mechanism by exploiting S–A–R (susceptible–adopted–recovered) propagation theory, as illustrated in Figure 1A. The explanation of the S–A–R model theory is as follows: a node in the susceptible state is denoted as the S-state, in the adopted state is denoted as the A-state, and in the recovered state is denoted as the R-state. Specifically, the individuals in the S-state cannot propagate information, but they can receive it from their A-state neighbors. The A-state individuals have adopted a piece of information and will transmit it to their neighbors in the S-state. The individuals in the R-state no longer

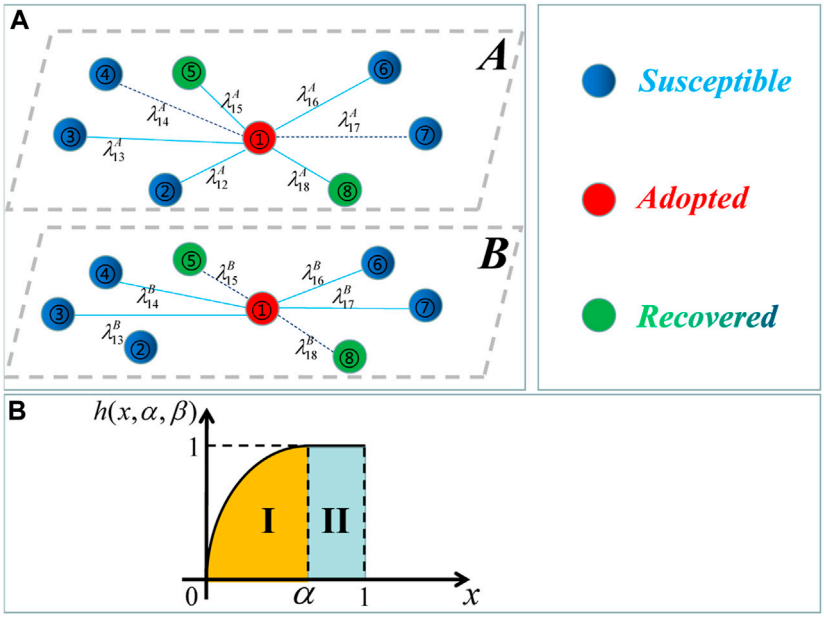


FIGURE 1
(A) Diagram of S–A–R propagation on the two-layer social network. Node 1 in the adopted state can transmit information to its neighbors in the susceptible state. Symbol λ represents the transmission probability in the social network. The dotted line means that the information has been transmitted in the last time step such as node 1 and its neighbors 4, 7 in layer *A* and node 1 and its neighbors 5, 8 in layer *B*. The solid line denotes that the information has not been transmitted along the corresponding edges. **(B)** Numerical function of a non-rule trapezoidal-like behavior. Symbol *h* denotes the proportion of information that a node in the adopted state has obtained its degree. Symbol α denotes the IHDB parameter.

receive information and exit the whole propagation process. If an individual transforms its state in the X-layer in the two-layer network model, it will change to the same state in other layers.

This paper first considers that the contact ability of individuals can be set as C. If $C < k_i$, where k_i is the degree of individual i, an individual that has been adopted who can only contact some of its neighbors. However, it can contact all of its neighbors if $C \geq k_i$. The probability of information transmission in the network can be set as λ. The probability that the information is received by the neighbor j in the S-state can be denoted as $\frac{\lambda C}{k_j^X}(X \in \{A, B\})$.

m^X is the total amount of information that the individual in the S-state has successfully received in the layer X. At the beginning, there is no information spreading on the multiple-layered network, the reason why the j value m_j^X of the individual in the S-state is 0. When a neighbor i in the A-state effectively spreads information to individual j along the related link in the A-layer or B-layer, the total information bits of individual j will rise by 1 at each time step, i.e., $m_j^X \rightarrow m_j^X + 1$. To explain the IHDB on information propagation, a non-rule trapezoidal-like function is proposed to illustrate the individual behavioral adoption in the whole network:

$$h_X(x,\alpha,\beta) = \begin{cases} \left(\frac{x}{\alpha}\right)^{\beta}, & 0 \leq x < \alpha, 0 < \beta < 1 \\ 1, & \alpha \leq x < 1, \end{cases} \quad (1)$$

where x represents the proportion between the receiving information of an individual and the number of its contacted neighbors and parameter α represents the IHDB variable. In region I of Figure 1B, the behavioral adoption probability rises slowly to 1 based on the increase in x. In region II of Figure 1B, the adoption probability remains at 1.

2.2 Propagation mechanism

We randomly select a portion of ρ_0 individuals to act as the individuals (seeds) in the A-state at the beginning of information propagation and all other individuals to act as S-state individuals. Each A-state individual randomly chooses C of its neighbors to whom information is transferred in layer $A(B)$ with probability $\lambda^A(\lambda^B)$. The total amount of information that individual j in the S-state effectively gets from layer X is $m_j^X \rightarrow m_j^X + 1$. Due to non-redundancy in information propagation, the information will not then be repeated through the same edge. Additionally, the individual in the susceptible state of X will accept the information and transmit to the adoption state with possibility $h_X(\frac{m^X}{k^X}, \alpha, \beta)$ at every time step. The S-state individual then changes to the corresponding state in other layers. Following successful information spreading, when the individual lost interest in information, the individuals in the adoption state transfer disinterested information and change to the recovery state with probability γ. In the two-layered contacted network, once there is no individual in the A-state, the information spreading process eventually comes to an end.

3 Theoretical analysis

We explore the novel IHDB properties of non-redundant information memory on two-layer networks by investigating

literature works [37, 38]. Then, in order to mathematically investigate the information propagation mechanism, we propose an information propagation theory including individual limited contact and IHDB. The ultimate spreading size is defined as the proportion of individuals in the recovery state when the information spreading progress has ended. We introduce the initialization state [39] in which the individual can receive information but cannot transmit it outside.

The probability that an edge in layer $X(X \in (A, B))$ has not seen information propagation toward its S-state neighbor j up to time t is represented by $\theta^X_{k^X_j}(t)$. With a degree of k^X_j, the probability that i and j are neighbors in layer X is represented by the expression $\frac{k^X_j P(k^X_j)}{\langle k^X \rangle}$. As a result, at time t, the individual i cannot receive the information by its neighbors in the X-layer with probability

$$\theta_A(t) = \sum_{k^A_j=0} \frac{k^A_j P\left(k^A_j\right)}{\langle k^A \rangle} \theta^A_{k^A_j}(t), \tag{2}$$

$$\theta_B(t) = \sum_{k^B_j=0} \frac{k^B_j P\left(k^B_j\right)}{\langle k^B \rangle} \theta^B_{k^B_j}(t), \tag{3}$$

respectively.

Until time t, the probability that an individual in the S-state possessing $\overrightarrow{k_i} = (k^A_i, k^B_i)$ will cumulatively take m_X bits of information in the A-layer or B-layer can be expressed as

$$\phi^A_{m_A}\left(k^A_i, t\right) = \binom{k^A_i}{m_A} \theta_A(t)^{k^A_i - m_A} \left[1 - \theta_A(t)\right]^{m_A}, \tag{4}$$

$$\phi^B_{m_B}\left(k^B_i, t\right) = \binom{k^B_i}{m_B} \theta_B(t)^{k^B_i - m_B} \left[1 - \theta_B(t)\right]^{m_B}, \tag{5}$$

respectively.

Individual i with m_X pieces of information in layer X, according to the IHDB features and the adoption threshold function, has not adopted the information and remained in the S-state with a probability $\prod_{j=0}^{m_X}[1 - h_X(\frac{j}{k^X_i}, \alpha, \beta)]$. While getting m_X bits of information from those of layer X, the probability that an individual remains in the S-state at time t can be expressed as

$$\begin{aligned}
\tau^X_{m_X}\left(k^X_i, t\right) &= \sum_{m_X=0}^{k^X_i} \phi^X_{m_X}\left(k^X_i, t\right) \prod_{j=0}^{m_X} \left[1 - h_X\left(\frac{j}{k^X_i}, \alpha, \beta\right)\right] \\
&= \sum_{m_X=0}^{\lfloor \alpha k^X_i \rfloor} \phi^X_{m_X}\left(k^X_i, t\right) \prod_{j=0}^{m_X} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right] \\
&\quad + \sum_{m_X=\lceil \alpha k^X_i \rceil}^{k^X_i} \phi^X_{m_X}\left(k^X_i, t\right) \prod_{j=0}^{\lceil \alpha k^X_i \rceil} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right] \prod_{j=\lceil \alpha k^X_i \rceil}^{m_X} (1 - 1) \\
&= \sum_{m_X=0}^{\lfloor \alpha k^X_i \rfloor} \phi^X_{m_X}\left(k^X_i, t\right) \prod_{j=0}^{m_X} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right].
\end{aligned} \tag{6}$$

As a result, the S-state individual i gets m_A and m_B pieces of information until time t and continues to be in the S-state possessing the probability

$$\begin{aligned}
s\left(\vec{k}, t\right) &= (1 - \rho_0) \sum_{m_A=0}^{k^A_i} \phi^A_{m_A}\left(k^A_i, t\right) \prod_{j=0}^{m_A} \left[1 - h_A\left(\frac{j}{k^A_i}, \alpha, \beta\right)\right] \\
&\quad \times \sum_{m_B=0}^{k^B_i} \phi^B_{m_B}\left(k^B_i, t\right) \prod_{j=0}^{m_B} \left[1 - h_B\left(\frac{j}{k^B_i}, \alpha, \beta\right)\right] \\
&= (1 - \rho_0) \tau^A_{m_A}\left(k^A_i, t\right) \tau^B_{m_B}\left(k^B_i, t\right).
\end{aligned} \tag{7}$$

The probability that a piece of information would have accumulated in the X-layer by time t when the S-state individual has not received information is expressed as

$$\eta_X = \sum_{k^X_i} P_X\left(k^X_i\right) \tau^X_{m_X}\left(k^X_i, t\right). \tag{8}$$

As a result, the percentage of the S-state individuals in the multi-layer network at time t is denoted as

$$S(t) = \sum_{\vec{k}} P\left(\vec{k}\right) s\left(\vec{k}, t\right) = (1 - \rho_0) \eta_A \eta_B. \tag{9}$$

All of the model's individuals can only alternate between three states; therefore, $\theta^X_{k^X_j}(t)$ can be transformed to

$$\theta^X_{k^X_j}(t) = \xi^X_{S,k^X_j}(t) + \xi^X_{A,k^X_j}(t) + \xi^X_{R,k^X_j}(t), \tag{10}$$

where the probability that a neighbor of individual j is in the S-state, A-state, or R-state and has not sent the information to i by time t in layer X is represented by $\xi^X_{S,k^X_j}(t)$, $\xi^X_{A,k^X_j}(t)$, and $\xi^X_{R,k^X_j}(t)$, respectively.

Individual i in the initialization state cannot contact its neighbors due to the cavity theory. The degree vector of individual j is $\overrightarrow{k_j} = (k^A_j, k^B_j)$. The S-state individual j can receive information from all neighbors in layer A with the exception of individual i of layer A and k^B_j neighbors of layer B if individual in the S-state i links its neighbor j of layer A. The probability for the individual j with degree $\overrightarrow{k_j} = (k^A_j, k^B_j)$ is indicated by $\varsigma^A_{n_A}(k^X_j - 1, t)$ who receives n_A bits of information by its neighbors in the A-layer cumulatively up to time t. The probability $\varsigma^X_{n_X}(k^X_j - 1, t)$ can be denoted as

$$\begin{aligned}
\varsigma^X_{n_X}\left(k^X_j - 1, t\right) &= \sum_{n_X=0}^{k^X_j - 1} \phi^X_{n_X}\left(k^X_j - 1, t\right) \prod_{j=0}^{n_X} \left[1 - h_X\left(\frac{j}{k^X_j}, \alpha, \beta\right)\right] \\
&= \sum_{n_X=0}^{\lfloor \alpha k^X_j \rfloor} \phi^X_{n_X}\left(k^X_j - 1, t\right) \prod_{j=0}^{n_X} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right] \\
&\quad + \sum_{n_X=\lceil \alpha k^X_j \rceil}^{k^X_j - 1} \phi^X_{n_X}\left(k^X_j - 1, t\right) \prod_{j=0}^{\lceil \alpha k^X_j \rceil} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right] \\
&\quad \prod_{j=\lceil \alpha k^X_j \rceil}^{n_X} (1 - 1) = \sum_{n_X=0}^{\lfloor \alpha k^X_j \rfloor} \phi^X_{n_X}\left(k^X_j - 1, t\right) \prod_{j=0}^{n_X} \left[1 - \left(\frac{j}{\alpha k^X_i}\right)^{\beta}\right].
\end{aligned} \tag{11}$$

j with $\vec{k}_j = (k^A_j, k^B_j)$ is also more likely to have acquired n_B bits of information from neighbors in layer B at time t by $\tau^B_{n_B}(k^B_j, t)$. Following the accumulation of n_A and n_B bits of information, the probability that j will stay in the susceptible state is given as

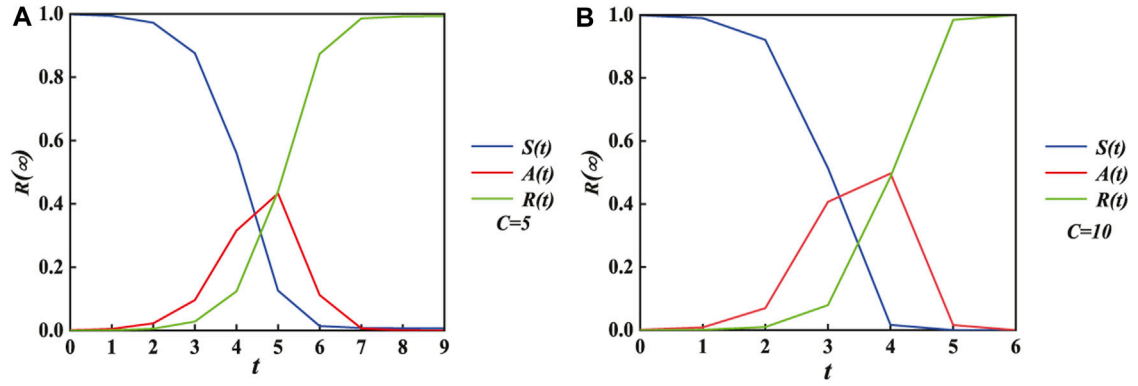


FIGURE 2
Individual status has changed three stages over time. For the same condition, the propagation time spends nine steps in subgraph **(A)** setting $C = 5$ and six steps in subgraph **(B)** setting $C = 10$. By comparison, the increase in the individual contact capacity accelerates the propagation process. The other variables include $\alpha^A = \alpha^B = 0.1$, $\beta^A = \beta^B = 0.1$, and $\lambda^A = \lambda^B = 0.5$.

$$\begin{aligned}\Theta_A\left(\vec{k},t\right) &= \sum_{n_A=0}^{k_j^A-1}\phi_{n_A}^A\left(k_j^A-1,t\right)\prod_{j=0}^{n_A}\left[1-h_A\left(\frac{j}{k_j^A},\alpha,\beta\right)\right]\\ &\times\sum_{n_B=0}^{k_j^B}\phi_{n_B}^B\left(k_j^B,t\right)\prod_{j=0}^{n_B}\left[1-h_B\left(\frac{j}{k_j^B},\alpha,\beta\right)\right]\\ &= \varsigma_{n_A}^A\left(k_j^A-1,t\right)\tau_{n_B}^B\left(k_j^B,t\right).\end{aligned} \tag{12}$$

The probability that j in layer B will continue to be in the susceptible state after receiving all of the n_A and n_B bits of information is $\overrightarrow{k_j}$ when the individual i in the S-state interacts with j by a degree of $\overrightarrow{k_j}$ is expressed as

$$\begin{aligned}\Theta_B\left(\vec{k},t\right) &= \sum_{n_A=0}^{k_j^A}\phi_{n_A}^A\left(k_j^A,t\right)\prod_{j=0}^{n_A}\left[1-h_A\left(\frac{j}{k_j^A},\alpha,\beta\right)\right]\\ &\times\sum_{n_B}^{k_j^B-1}\phi_{n_B}^B\left(k_j^B-1,t\right)\prod_{j=0}^{n_B}\left[1-h_B\left(\frac{j}{k_j^B},\alpha,\beta\right)\right]\\ &= \tau_{n_A}^A\left(k_j^A,t\right)\varsigma_{n_B}^B\left(k_j^B-1,t\right).\end{aligned} \tag{13}$$

So, the probability that individual i and individual j in the S-state are linked by an edge can be expressed as

$$\xi_{S,k_j^X}^X(t) = \left(1-\rho_0\right)\frac{\sum_{\overrightarrow{k_j}}k_j^X P_X\left(\overrightarrow{k_j}\right)\Theta_X\left(\overrightarrow{k_j}t\right)}{\langle k^X\rangle}, \tag{14}$$

where $\frac{k_j^X P_X(\overrightarrow{k_j})}{\langle k^X\rangle}$ defines the normal degree of the X-layer and $\langle k^X\rangle$ represents the probability that individual i is next to j possessing k_j^X.

The evolutionary equation of $\xi_{R,k_j^X}^X(t)$ and $\xi_{A,k_j^X}^X(t)$ of layer X can be examined in the following expression. An A-state individual possessing degree k_j^X chooses $\frac{C}{k_j^X}$ of its neighbors and contacts them. The probability of information transmission is λ. As a result, the probability that rumor will spread from individual j to its neighbors can be denoted by $\frac{\lambda C}{k_j^X}$. The following expression defines how $\theta_{k_j^X}^X(t)$ is evolved in layer X:

$$\frac{d\theta_{k_j^X}^X(t)}{dt} = -\frac{\lambda C}{k_j^X}\xi_{A,k_j^X}^X(t). \tag{15}$$

Individuals in the adoption state cease spreading information and move to the recovered state with γ. The calculation for the evolution of $\xi_{R,k_j^X}^X(t)$ is

$$\frac{d\xi_{R,k_j^X}^X(t)}{dt} = \gamma\xi_{A,k_j^X}^X(t)\left(1-\frac{\lambda C}{k_j^X}\right). \tag{16}$$

Combining Eqs 15, 16 with the original conditions $\theta_{k_j^X}^X(0) = 1$ and $\xi_{R,k_j^X}^X(0) = 0$, the development of $\xi_{R,k_j^X}^X(t)$ in layer $X(X \in \{A, B\})$ is

$$\xi_{R,k_j^X}^X(t) = \gamma\left[1-\theta_{k_j^X}^X(t)\right]\left[\frac{k_j^X}{\lambda C}-1\right]. \tag{17}$$

Combining Eq. 10, Eq. 14, and Eq. 17 will derive

$$\begin{aligned}\xi_{A,k_j^X}^X(t) &= \theta_{k_j^X}^X(t)-\xi_{S,k_j^X}^X(t)-\xi_{R,k_j^X}^X(t)\\ &= \theta_{k_j^X}^X(t)-\left(1-\rho_0\right)\frac{\sum_{\overrightarrow{k_j}}k_j^X P_X\left(\overrightarrow{k_j}\right)\Theta\left(\overrightarrow{k_j}t\right)}{\langle k^X\rangle}-\gamma\left[1-\theta_{k_j^X}^X(t)\right]\\ &\times\left(\frac{k_j^X}{\lambda C}-1\right).\end{aligned} \tag{18}$$

Substituting Eq. 18 into Eq. 15, it is possible to rewrite the evolution of $\theta_{k_j^X}^X(t)$ in layer $X(X \in \{A, B\})$ as

$$\begin{aligned}\frac{d\theta_{k_j^X}^X(t)}{dt} &= -\frac{\lambda C}{k_j^X}\left\{\theta_{k_j^X}^X(t)-\left(1-\rho_0\right)\frac{\sum_{\overrightarrow{k_j}}k_j^X P_X\left(\overrightarrow{k_j}\right)\Theta\left(\overrightarrow{k_j}t\right)}{\langle k^X\rangle}-\gamma\left[1-\theta_{k_j^X}^X(t)\right]\left(\frac{k_j^X}{\lambda C}-1\right)\right\}\\ &= \left(1-\rho_0\right)\frac{\lambda C}{k_j^X}\frac{\sum_{\overrightarrow{k_j}}k_j^X P_X\left(\overrightarrow{k_j}\right)\Theta\left(\overrightarrow{k_j}t\right)}{\langle k^X\rangle}+\gamma\left(1-\frac{\lambda C}{k_j^X}\right)-\left[\gamma+\frac{\lambda C}{k_j^X}\left(1-\gamma\right)\right]\theta_{k_j^X}^X(t).\end{aligned} \tag{19}$$

In the whole network, the expression for state change of the individuals is given as follows:

$$\frac{dR(t)}{dt} = \gamma A(t), \tag{20}$$

$$\frac{dA(t)}{dt} = -\frac{dS(t)}{dt} - \gamma A(t). \tag{21}$$

As a result, $S(t)$, $A(t)$, and $R(t)$, which represent the nodes' states at every given time step, can be calculated by combining and iterating Eqs 9, 20, 21, respectively.

While $t \to \infty$, the nodes' states in the network will not become any longer, meaning that $\frac{d\theta^X_{k^X_j}(t)}{dt} \to 0$. $R(\infty)$ is also the ultimate propagation size. Currently, the entire network consists solely of individuals in the S-state and the R-state. To determine $R(\infty)$, we must determine $\theta^X_{k^X_j}(\infty)$ as

$$\theta^X_{k^X_j}(\infty) = \xi^X_{S,k^X_j}(\infty) + \gamma\left[1 - \theta^X_{k^X_j}(\infty)\right]\left[\frac{k^X_j}{\lambda C} - 1\right]. \tag{22}$$

Then, by combining and iterating Eqs 9, 22, $S(\infty)$ and $R(\infty)$ are derived.

The crucial propagation probability is our next problem of concern. Moving the left side of Eq. 22 to the right side, it can be denoted as

$$F\left[\theta^X_{k^X_j}(\infty)\right] = \xi^X_{S,k^X_j}(\infty) - \theta^X_{k^X_j}(\infty) + \gamma\left[1 - \theta^X_{k^X_j}(\infty)\right]\left[\frac{k^X_j}{\lambda C} - 1\right]. \tag{23}$$

At the critical point $\theta^X_c(\infty)$, $F[\theta^X_{k^X_j}(\infty)]$ is tangent to the abscissa. The critical condition can be denoted as

$$\left.\frac{dF}{d\theta^X_{k^X_j}(\infty)}\right|_{\theta^X_c(\infty)} = 0. \tag{24}$$

Therefore, the critical information propagation probability is

$$\lambda = \frac{\gamma}{A + \gamma + 1}, \tag{25}$$

where

$$\left.\frac{d\xi^X_{S,k^X_j}(\infty)}{d\theta^X_{k^X_j}(\infty)}\right|_{\theta^X_c(\infty)} = 0. \tag{26}$$

4 Parameter settings

This section simulates and evaluates the developed model through testing on a multiple-layered contacted network such as ER [40] and SF networks [41]. The experiment is carried out in the ER network and the SF network with 10,000 nodes. Additionally, each layer network has $\langle k_X \rangle = 10$ as an average degree of $X \in \{A, B\}$. Furthermore, for convenience, the probability of information transmission can be given at $\lambda_A = \lambda_B = \lambda$. The experiment was conducted using double-layer contacted networks. In an ER network, there is an equal possibility of a connection forming between any two nodes, and the degrees of the nodes in layer X follow $P_X(k_X) = e^{-\langle k_X \rangle}\frac{\langle k \rangle^{k_X}}{k_X!}$. In SF networks, the degree distribution variability of nodes is negatively correlated with the degree exponent v. In an SF network, where $\zeta_X = \frac{1}{\sum_{k_X} k_X^{-v}}$, the degrees of nodes follow $P_X(k_X) = \zeta_X k_X^{-v}$. Additionally, a relatively small number of seeds ($\rho_0 = 0.001$) and a recovered probability $\gamma = 1.0$ cause A-state nodes to move to the R-state.

Furthermore, for further explanation of the critical condition in our scenario, the relative variance is unitized and written as follows:

$$\chi = N\frac{\langle R(\infty)^2 \rangle - \langle R(\infty) \rangle^2}{\langle R(\infty) \rangle}, \tag{27}$$

where $\langle \ldots \rangle$ stands for the mean set. The important parameters of the ultimate adoption size are implied by χ.

5 Experiments and discussion

5.1 Numerical analysis of the two-layer ER network

In Figure 2, with the unit transmission probability λ and the adoption threshold parameters α and β, we first investigate how the proportion of individuals in the three states of S-state, A-state, and R-state has changed with time.

Then, as shown in Figures 2A, B, $R(t)$, which represents the ultimate spreading size, changes to 1 at the end, while $S(t)$ increasingly decreases from 1 to 0 and $A(t)$ progressively drops to 0 over time. As the increase in parameter C in subgraph (a) to subgraph (b), the evolution time costed steadily decreases from 9 to 6, while $R(\infty)$, which denotes the ultimate spreading size, increases at the same step t. The time progress demonstrates that information outbreak on a two-layered contacted network can be accelerated by enhancing contact ability of individuals.

Figure 3 displays the roles of unit transmission probability in each individual's eventual adoption size with various IHDB values α in the subgraphs. The ultimate spreading size $R(\infty)$ spreads to a global network as λ rises, as shown in Figure 3A ($\beta = 0.5$) and (b) ($\beta = 0.9$). Furthermore, Figures 3A, B also indicate how IHDB can affect the propagation phase transition. In subgraph (a), the pattern of $R(\infty)$ always shows a second-order growth of continuous phase transition for any IHDB behavior exhibited by the individual ($\alpha^A = \alpha^B = 0.1, 0.5, 0.9$). The pattern of $R(\infty)$ in subgraph (b) indicates a second-order phase transition in the continuous propagation pattern when an individual exhibits a positive IHDB such as $\alpha^A = \alpha^B = 0.1$. This suggests that, when there is a small λ, a positive IHDB can result in widespread behavioral propagation. When $\alpha^A = \alpha^B = 0.5$, $R(\infty)$ also shows the same propagation phenomenon. While an individual exhibits a weak IHDB, $R(\infty)$ exhibits a first-order increase in the discontinuous pattern, i.e., $\alpha^A = \alpha^B = 0.9$.

Figure 3C displays the relative variances and critical information propagation probability of (a) and (b) individually (d). The global adoption will emerge from the deviation of behavioral propagation, which is represented by the top values of relative variance χ. Additionally, the numerical values of the simulation (symbols) agree with our theoretical analyses (lines).

For the two-layer contacted ER network, Figure 4 exhibits the joint impacts of variable (λ, α) on $R(\infty)$. Figure 4 (a) with $\beta^A = \beta^B = 0.5$ and

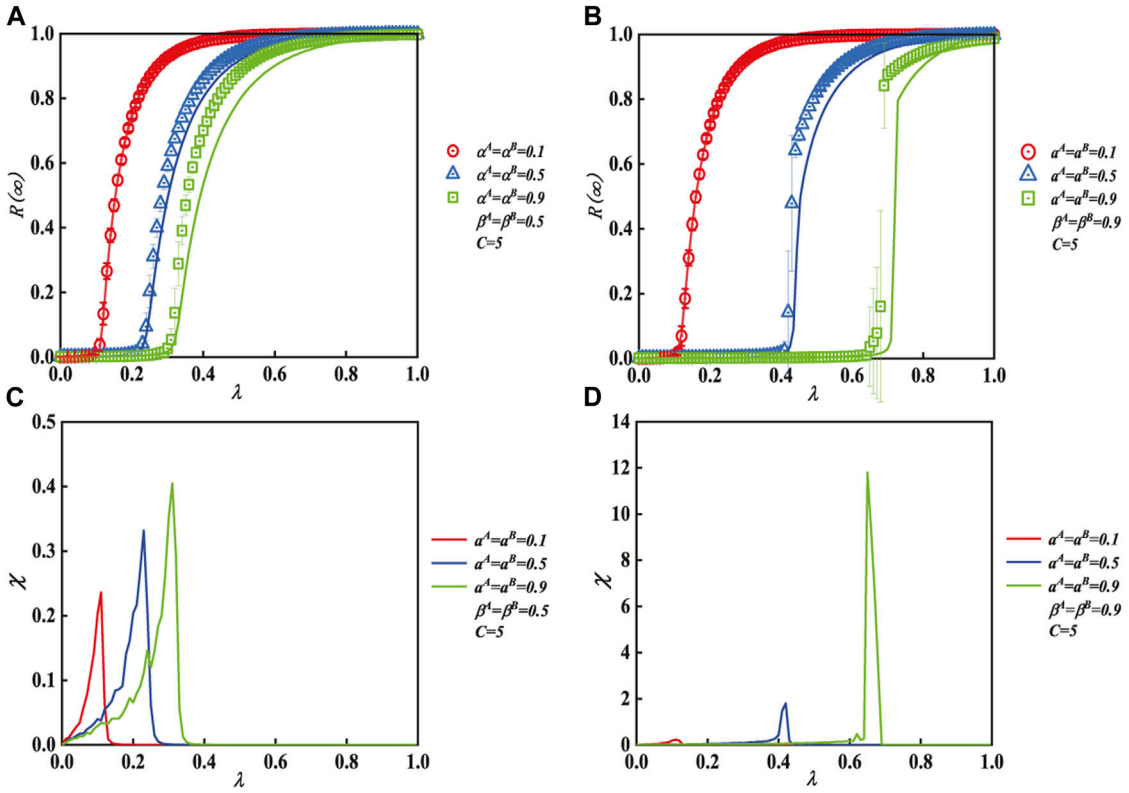


FIGURE 3
Effects of unit transmission probability of a multi-layer ER network on each individual's ultimate spreading size while using various IHDB parameters. Subgraphs **(A)** (β = 0.5) and **(B)** (β = 0.9) demonstrate how the IHDB parameter affects the propagation pattern. The critical values of subgraphs **(C)** and **(D)** show the relative deviations and the critical values of **(A)** and **(B)**, respectively.

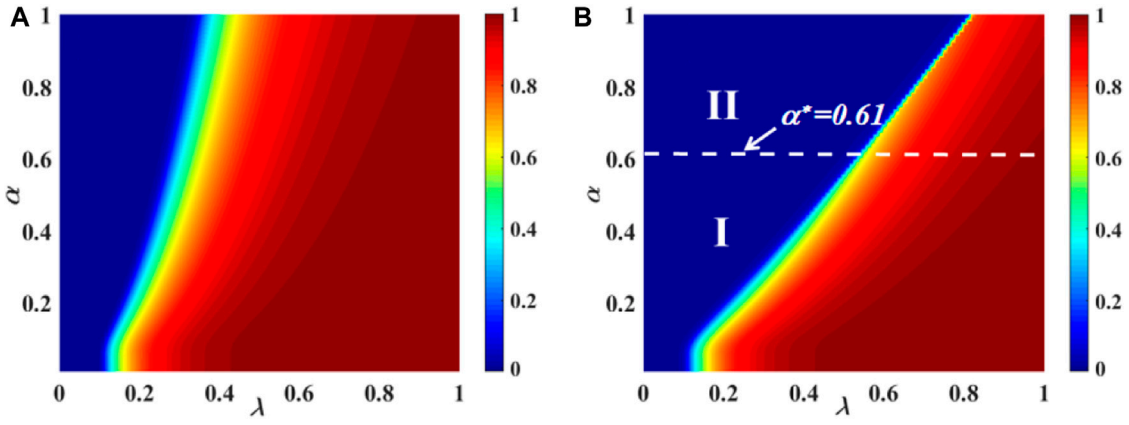


FIGURE 4
Impact of the unit transmission probability and the dynamic IHDB parameter α on each person's final spreading size for an ER network with numerous layers of contacts. The impacts of subgraphs **(A)** ($\beta^A = \beta^B$ = 0.5) and **(B)** ($\beta^A = \beta^B$ = 0.9) on the ultimate spreading size are shown with varying IHDB parameters. In subgraph **(A)**, the phase transition shows a continuous pattern in the whole area. In subgraphs **(B)**, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II.

(b) with $\beta^A = \beta^B = 0.9$ depict the effects of (λ, α) on the dissemination of information. In subgraph (a), the phase transition shows a continuous pattern in the whole area. Then, in subgraph (b), the image can be divided into two parts. In area I, a second-order continuous increase can be seen in the $R(\infty)$ pattern. The critical value between region I and region II is $\alpha^* = 0.61$. In area II, a first-order discontinuous increase can be seen in the $R(\infty)$ pattern. Additionally, the individual contact capability parameter is set at $C = 5$.

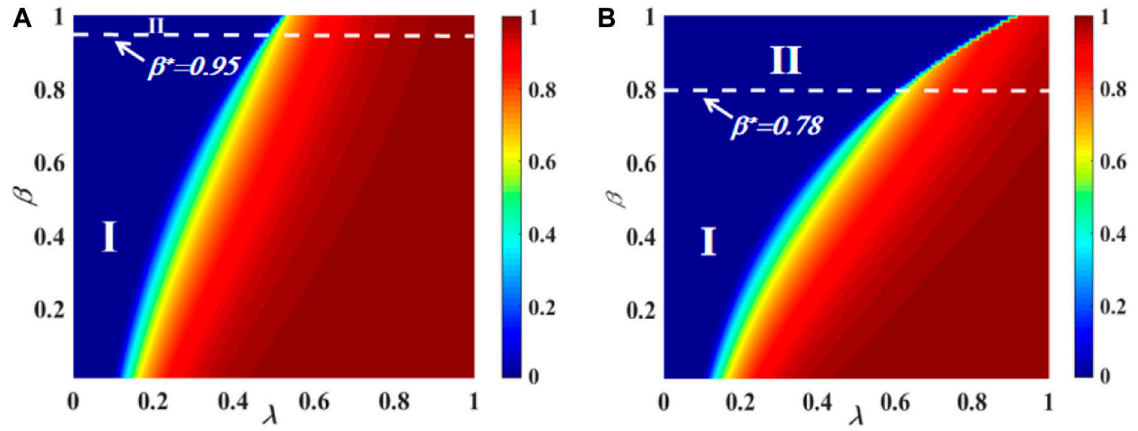


FIGURE 5
Integrative roles of the dynamic IHDB parameter α and transmission probability in each individual's ultimate spreading size for a multiple-layered contacted ER network. The impacts of subgraphs **(A)** ($\alpha^A = \alpha^B = 0.5$) and **(B)** ($\alpha^A = \alpha^B = 0.9$) on the ultimate spreading size are shown with varying IHDB parameters. In subgraph **(A)**, the phase transition shows a continuous pattern in the whole area. In subgraph **(B)**, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II.

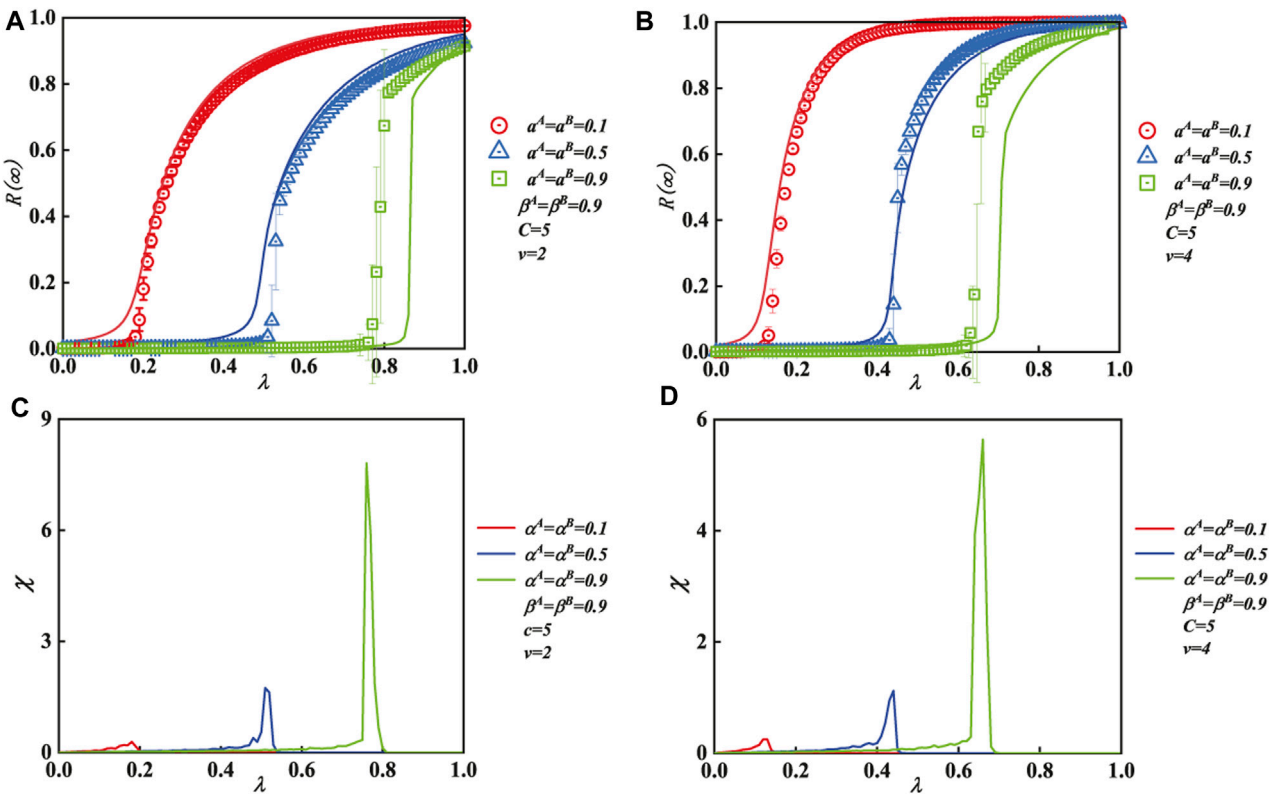


FIGURE 6
Effect of the unit transmission chance and the IHDB parameter α on each person's ultimate spreading size for the multiple contacted SF network The IHDB parameter ($\beta^A = \beta^B = 0.9$) and the same contact capacity of people ($C = 5$) are applied to each subgraph. The vertical subgraphs use a unique degree distribution exponent as well, with subgraphs **(A)** and **(B)** corresponding to $v = 2, 4$, respectively. Subgraphs **(A)** and **(B)** display the effects on the ultimate spreading size with unit transmission probability λ. The critical values of subgraphs **(C)** and **(D)** show the relative deviations and the critical values of **(A)** and **(B)**, respectively.

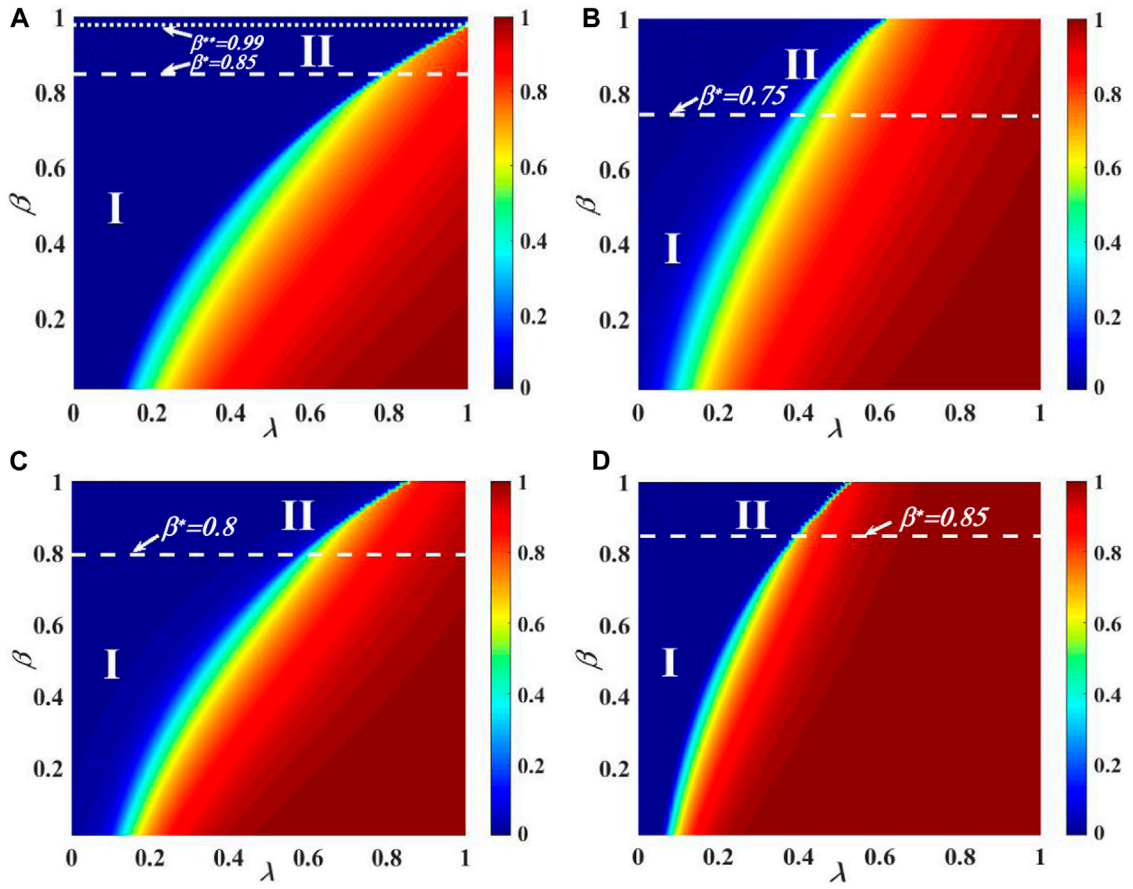


FIGURE 7
Combined effect of the unit transmission chance and the IHDB parameter β on the ultimate information outbreak for the two-layer contacted SF network. The influence of (λ, β) on the ultimate adoption size is shown in subgraphs **(A)** and **(B)** with v =2 and subgraphs **(C)** and **(D)** with v = 4, respectively. In subgraph **(A)** with C = 5, the phase transition shows three areas: the continuous second-order pattern in area I, the discontinuous first-order pattern in area II, and the static pattern in area III. In subgraph **(B)** with C = 10, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II. In subgraph **(C)** with C = 5, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II. In subgraph **(D)** with C = 10, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II.

The joint impacts of variable plane (λ, β) on $R(\infty)$ for the two-layered ER network are shown in Figure 5. Figures 5A, B depict the effects of (λ, β) on information propagation, respectively, with $\alpha^A = \alpha^B =$ 0.5 and $\alpha^A = \alpha^B = 0.9$. In subgraph (a), the image can be divided into two parts. There is a second-order continuous increase pattern in region I of $R(\infty)$ phase transition. The critical value between region I and region II is $\beta^* = 0.95$. There is a first-order discontinuous increase pattern in region II of $R(\infty)$. Then, in subgraph (b), the image can also be divided into two parts. In area I, $R(\infty)$ exhibits a second-order continuous spreading. The critical value between region I and region II is $\beta^* = 0.78$. In area II, $R(\infty)$ exhibits a first-order discontinuous spreading. Additionally, the individual contact capability parameter is set at $C = 5$.

5.2 Numerical analysis of the two-layer SF network

Figure 6 depicts the influence of IHDB variable α and transmission probability λ on the ultimate spreading size of the multi-layer contacted SF network. In each subgraph, the fundamental parameters include $C = 5$ and $\beta^A = \beta^B = 0.9$. Figures 6A, B show how the ultimate adoption size $R(\infty)$ grows as λ increases until it achieves global adoption. When $\alpha^A = \alpha^B = 0.1$ and $\alpha^A = \alpha^B = 0.5$, the final spreading size shows a continuous spreading with second-order. However, $R(\infty)$ pattern exhibits a discontinuous spreading with first-order when $\alpha^A = \alpha^B = 0.9$. Then, the same growth pattern is also exhibited in subgraph (b) ($v = 4$). Moreover, compared with subgraph (b) ($v = 4$), subgraph (a) ($v = 2$) shows an incomplete global adoption because of strong heterogeneous degree distribution. Additionally, the numerical values of the simulation (symbols) match those of our theoretical analyses (lines).

For the multiple-layered contacted SF network, Figure 7 shows the effect of $R(\infty)$ on the behavioral parameter plane (λ, β). The subgraphs (a) and (b), and (c) and (d) are set as the identical contact capacity of individuals by $C = 5$ and $C = 10$, respectively. The subgraphs (a) and (b), and (c) and (d) demonstrate the growth tendency of $R(\infty)$. In subgraph (a) with $v = 2$ and $C = 5$, the image can be divided into three parts. In the phenomenon of eventual information outbreak

10.3389/fphy.2023.1239257

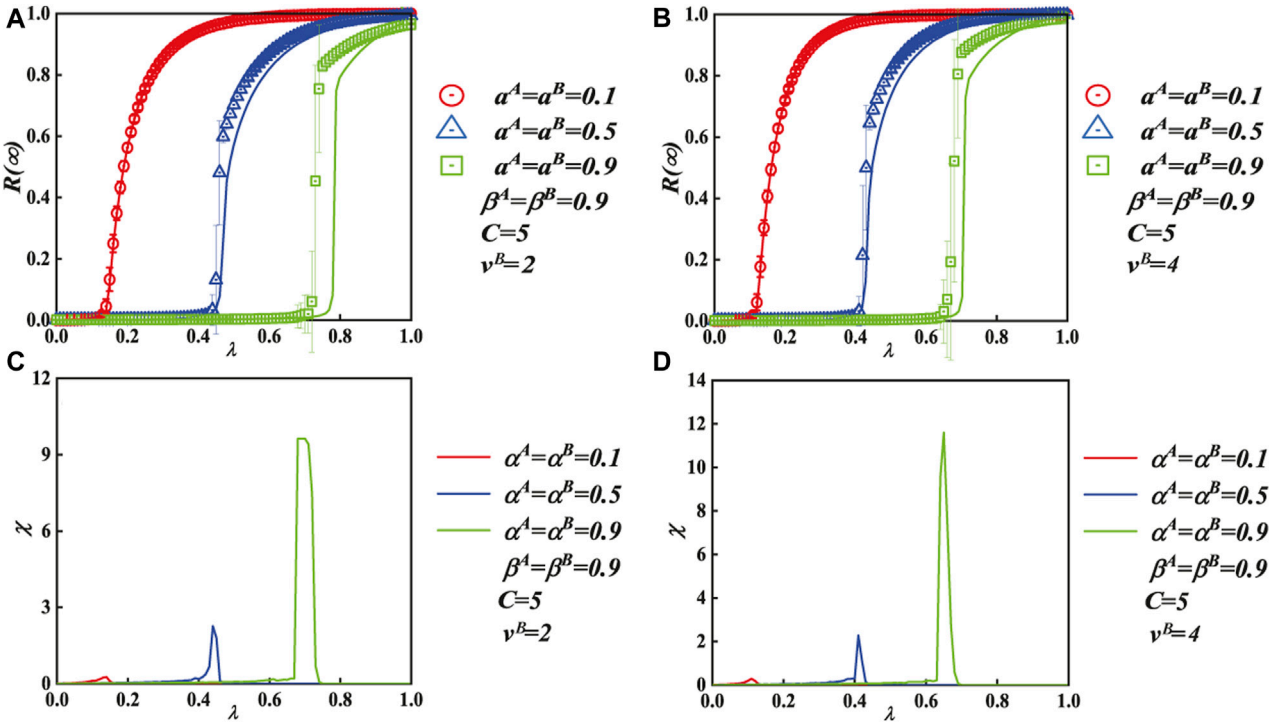


FIGURE 8
Effect of the unit transmission chance and the IHDB parameter α on each person's ultimate spreading size for the two-layer contacted ER–SF network. The IHDB parameter ($\beta^A = \beta^B = 0.9$) and the same contact capacity of people ($C = 5$) are applied to each subgraph. The vertical subgraphs use a unique degree distribution exponent as well, with subgraphs **(A)** and **(B)** corresponding to the SF layer with $v^B = 2$, 4, respectively. Subgraphs **(A)** and **(B)** display the effects on the ultimate spreading size with unit transmission probability λ. The critical values of subgraphs **(C)** and **(D)** show the relative deviations and the critical values of **(A)** and **(B)**, respectively. The initial seed is set at $\rho = 0.001$.

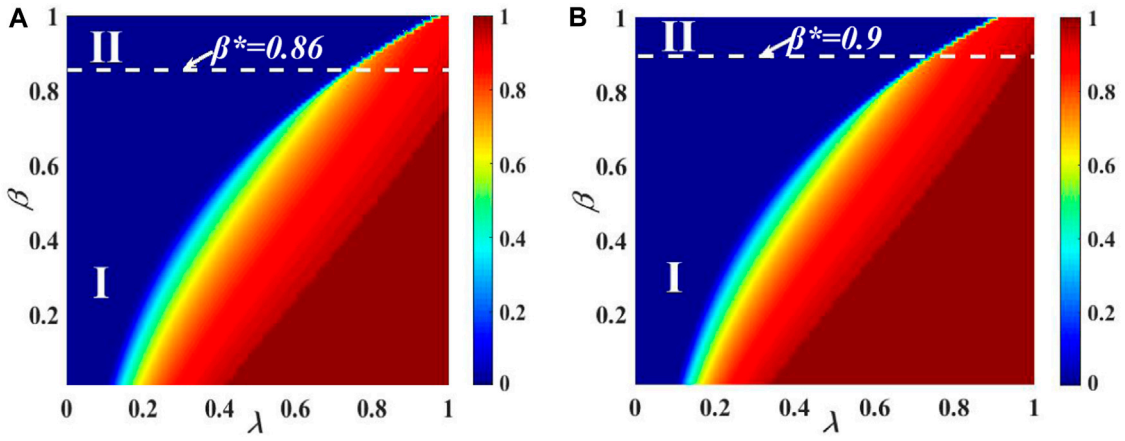


FIGURE 9
Combined effect of the unit transmission chance and the IHDB parameter β on the ultimate information outbreak for the two-layer contacted ER–SF network. The influence of (λ, β) on the ultimate adoption size is shown in subgraph **(A)** with $v = 2$ and subgraph **(B)** with $v = 4$, respectively. In subgraph **(A)**, the phase transition shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II. In subgraph **(B)**, the phase transition also shows two areas: the continuous second-order pattern in area I and the discontinuous first-order pattern in area II. Other parameters are set at $C = 5$, $\alpha^A = \alpha^B = 0.9$, and $\rho_0 = 0.001$.

$R(\infty)$, area I displays a second-order continuous propagation pattern. The critical value between region I and region II is $\beta^* = 0.85$. Area II exhibits a first-order discontinuous phase transition in the pattern of $R(\infty)$. The critical value between region II and region III is $\beta^{**} = 0.99$. In area III, the $R(\infty)$ pattern does not exhibit information outbreak. In subgraph (b) with $v = 2$ and $C = 10$, the image can be

divided into two parts. The $R(\infty)$ pattern of individual final spreading size in area I displays a second-order continuous increase. The critical value between region I and region II is $\beta^{\star} = 0.75$. In the pattern of $R(\infty)$, area II displays a first-order discontinuous increase. In subgraph (c) with $\nu = 4$ and $C = 5$, the image can be divided into two parts. While showing the pattern of information outbreak scale $R(\infty)$, area I displays a second-order continuous propagation pattern. The critical value between region I and region II is $\beta^{\star} = 0.8$. A first-order discontinuous increase in area II's pattern of $R(\infty)$ is visible. In subgraph (d) with $\nu = 4$ and $C = 10$, the image can be divided into two parts. In the pattern of individual eventual spreading size $R(\infty)$, area I displays a second-order continuous increase. The critical value between region I and region II is $\beta^{\star} = 0.85$. Area II exhibits a first-order discontinuous phase transition in the pattern of $R(\infty)$. Furthermore, the heterogeneous degree distribution alters the information propagation but cannot alter the pattern of phase transition. Because there are some hub people in the multi-layer contacted SF network, when it exhibits a strong heterogeneous degree distribution ($\nu = 2$), there is a pattern of information suppression in the phase transition.

5.3 Numerical analysis of the two-layer ER–SF network

Figure 8 exhibits the impact of IHDB variable α and transmission probability λ on the final spreading scope for the two-layer contacted ER–SF network. In each subgraph, the fundamental parameters include $C = 5$ and $\beta^{A} = \beta^{B} = 0.9$. Figures 8A, B ($\nu^{B} = 2$ and $\nu^{B} = 4$) show how the final spreading scope $R(\infty)$ grows as λ increases until it achieves global adoption. When $\alpha^{A} = \alpha^{B} = 0.1$ and $\alpha^{A} = \alpha^{B} = 0.5$, the final outbreak pattern shows a second-order continuous propagation. However, the $R(\infty)$ pattern shows a first-order discontinuous pattern when $\alpha^{A} = \alpha^{B} = 0.9$. In addition, the numerical values of the simulation (symbols) match those of our theoretical analyses (lines).

For the two-layer contacted ER–SF network, Figure 9 shows the effect of $R(\infty)$ on the behavioral parameter plane (λ, β). The subgraphs (a) and (b) are set as the identical contact capacity of individuals by $C = 5$. The subgraphs (a) and (b) demonstrate the growth tendency of $R(\infty)$. In subgraph (a) with $\nu = 2$, the image can be divided into two parts. In the phenomenon of eventual information outbreak $R(\infty)$, area I displays a second-order continuous propagation pattern. The critical value between area I and II is $\beta^{\star} = 0.86$. Area II exhibits a first-order discontinuous phase transition in the pattern of $R(\infty)$. In subgraph (b) with $\nu = 4$, the image can be divided into two parts. The $R(\infty)$ pattern of individual final spreading size in area I displays a second-order continuous increase. The critical value between region I and region II is $\beta^{\star} = 0.9$. In the pattern of $R(\infty)$, area II displays a first-order discontinuous increase.

6 Conclusion

Researchers have explored how transmission probability, information type, individual psychology, and heterogeneous behaviors affect information propagation mechanisms. In the research on information propagation, we found that individual behavior depends on individual psychology to show positive or negative, linear or non-linear adoption. In this paper, we explore how individual psychology affects the information propagation.

This paper considers the individual contact capacity, which affects the information outbreak. More importantly, we found that an individual can show the heterogeneous decreasing behavior on information propagation, which is called IHDB. Then, we proposed a non-rule trapezoidal-like probability function on the two-layer network model. Meanwhile, we proposed a novel generalized edge-based compartmental theory to analyze the information propagation mechanism. Finally, the propagation pattern on the two-layer contacted ER and SF networks was revealed by the simulation and theoretical analysis. When IHDB changes, the phenomenon of the ultimate information outbreak first increases continuously in the second-order phase transition and then increases discontinuously in the first-order phase transition. Additionally, increasing the number of contacted neighbors makes it easier for propagation information and changing the propagation pattern. Furthermore, the heterogeneous degree distribution also has influence on information spreading but has not altered the phase transition pattern. This paper demonstrates the impact of individual heterogeneous decreasing behavior on information propagation. We also present a heuristic theory to describe how individual behavior affects the propagation of information.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

YT: conceptualization, methodology, visualization, investigation, and writing—review and editing. HT: methodology, supervision, and formal analysis. XZ: conceptualization, methodology, and formal analysis. QC: formal analysis and supervision. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1807800 and the BUPT Excellent Ph.D. Students Foundation under Grant No. CX2022209.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Peng H, Peng W, Zhao D, Wang W. Impact of the heterogeneity of adoption thresholds on behavior spreading in complex networks. *Appl Maths Comput* (2020) 386: 125504. doi:10.1016/j.amc.2020.125504

2. Cui Y, Wei R, Tian Y, Tian H, Zhu X. Information propagation influenced by individual fashion-passion trend on multi-layer weighted network. *Chaos, Solitons and Fractals* (2022) 160:112200. doi:10.1016/j.chaos.2022.112200

3. Yang Q, Zhu X, Tian Y, Wang G, Zhang Y, Chen L. The influence of heterogeneity of adoption thresholds on limited information spreading. *Appl Maths Comput* (2021) 411:126448. doi:10.1016/j.amc.2021.126448

4. Cui Q, Hu X, Ni W, Tao X, Zhang P, Chen T, et al. Vehicular mobility patterns and their applications to internet-of-vehicles: A comprehensive survey. *SCIENCE CHINA-Information Sci* (2022) 65(1–42):211301. doi:10.1007/s11432-021-3487-x

5. Wang Z, Guo Q, Sun S, Xia C. The impact of awareness diffusion on sir-like epidemics in multiplex networks. *Appl Maths Comput* (2019) 349:134–47. doi:10.1016/j.amc.2018.12.045

6. Zhao D, Li L, Peng H, Luo Q, Yang Y. Multiple routes transmitted epidemics on multiplex networks. *Phys Lett A* (2014) 378:770–6. doi:10.1016/j.physleta.2014.01.014

7. Zhou L, Huang M, Tan F, Zhang Y. Mean-square bounded synchronization of complex networks under deception attacks via pinning impulsive control. *Nonlinear Dyn* (2023) 111:11243–59. doi:10.1007/s11071-023-08448-0

8. Yao W, Wang C, Sun Y, Gong S, Lin H. Event-triggered control for robust exponential synchronization of inertial memristive neural networks under parameter disturbance. *Neural Networks* (2023) 164:67–80. doi:10.1016/j.neunet.2023.04.024

9. Fei Y, Liu L, Xiao L, Li K, Cai S. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* (2019) 350:108–16. doi:10.1016/j.neucom.2019.03.053

10. Li W, Jiang L, Perc M. A limited mobility of minorities facilitates cooperation in social dilemmas. *Appl Maths Comput* (2021) 391(125705):125705. doi:10.1016/j.amc.2020.125705

11. Perc M, Jordan JJ, Rand DG, Wang Z, Boccaletti S, Szolnoki A. Statistical physics of human cooperation. *Phys Rep* (2017) 687:1–51. doi:10.1016/j.physrep.2017.05.004

12. Faff R, Ho YK, Lin W, Yap CM. Diminishing marginal returns from R&D investment: Evidence from manufacturing firms. *Appl Econ* (2013) 45:611–22. doi:10.1080/00036846.2011.608644

13. Gauriot R, Heger SA, Slonim R. Altruism or diminishing marginal utility? *J Econ Behav Organ* (2020) 180:24–48. doi:10.1016/j.jebo.2020.09.030

14. Hirschauer N, Jantsch A, Musshoff O. Developing business ethics theory and integrating economic analysis into business ethics teaching–a conceptualization based on externalities and diminishing marginal utility. *Rev Soc Economy* (2018) 76:43–72. doi:10.1080/00346764.2017.1333132

15. Ahl RE, Cook E, McAuliffe K. Having less means wanting more: Children hold an intuitive economic theory of diminishing marginal utility. *Cognition* (2023) 234:105367. doi:10.1016/j.cognition.2023.105367

16. Liobikienė G, Butkus M. Determinants of greenhouse gas emissions: A new multiplicative approach analysing the impact of energy efficiency, renewable energy, and sector mix. *J Clean Prod* (2021) 309:127233. doi:10.1016/j.jclepro.2021.127233

17. Chandio AA, Jiang Y, Gessesse AT, Dunya R. The nexus of agricultural credit, farm size and technical efficiency in sindh, Pakistan: A stochastic production frontier approach. *J Saudi Soc Agric Sci* (2019) 18:348–54. doi:10.1016/j.jssas.2017.11.001

18. Jebeli SSH, Barouni M, Orojloo PH. Estimating the marginal effect of socioeconomic factors on the demand of specialty drugs. *Glob J Health Sci* (2015) 7(28).doi:10.5539/gjhs.v7n2p28

19. Park T, Griggs SK, Chung PD. Characteristics of patients using specialty medications. *Res Soc Administrative Pharm* (2018) 14:901–8. doi:10.1016/j.sapharm.2017.10.007

20. Cordis AS, Muzatko S. Higher education spending and cpa exam performance. *J Account Educ* (2021) 55:100727. doi:10.1016/j.jaccedu.2021.100727

21. Ng ES, McGinnis Johnson. Game of loans J. Game of loans: The relationship between education debt, social responsibility concerns, and making a career choice in the public, private, and nonprofit sectors. *Nonprofit Voluntary Sector Q* (2020) 49: 292–315. doi:10.1177/0899764019867773

22. Chae I, Bruno HA, Feinberg FM. Wearout or weariness? Measuring potential negative consequences of online ad volume and placement on website visits. *J Marketing Res* (2019) 56:57–75. doi:10.1177/0022243718820587

23. Agrawal D. Effect of brand loyalty on advertising and trade promotions: A game theoretic analysis with empirical evidence. *Marketing Sci* (1996) 15:86–108. doi:10.1287/mksc.15.1.86

24. Issock Issock PB, Mpinganjira M, Roberts-Lombard M. Investigating the relevance of the traditional marketing mix across different stages of change: Empirical evidence from household recycling. *J Soc Marketing* (2021) 11:489–506. doi:10.1108/jsocm-11-2020-0221

25. Götz O, Hoelter AK, Krafft M. The role of sales and marketing in market-oriented companies. *J Personal Selling Sales Manage* (2013) 33:353–71. doi:10.2753/PSS0885-3134330401

26. Fonseca R, Michaud PC, Zheng Y. The effect of education on health: Evidence from national compulsory schooling reforms. *SERIEs* (2020) 11:83–103. doi:10.1007/s13209-019-0201-0

27. Asem EK, Rajwa B. Impact of combination of short lecture and group discussion on the learning of physiology by nonmajor undergraduates. *Adv Physiol Educ* (2023) 47: 1–12. doi:10.1152/advan.00022.2022

28. Li X, Hsee CK. The psychology of marginal utility. *J Consumer Res* (2021) 48: 169–88. doi:10.1093/jcr/ucaa064

29. Gelardi V, Le Bail D, Barrat A, Claidiere N. From temporal network data to the dynamics of social relationships. *Proc R Soc B* (2021) 288:20211164. doi:10.1098/rspb.2021.1164

30. Koltsova OY, Mararitsa LV, Terpilovskii MA, Sinyavskaya YE. Social signature in an online environment: Stability and cognitive limits. *Comput Hum Behav* (2021) 122: 106856. doi:10.1016/j.chb.2021.106856

31. Kojaku S, Hébert-Dufresne L, Mones E, Lehmann S, Ahn Y-Y. The effectiveness of backward contact tracing in networks. *Nat Phys* (2021) 17:652–8. doi:10.1038/s41567-021-01187-2

32. Zhu X, Tian H, Chen X, Wang W, Cai S. Heterogeneous behavioral adoption in multiplex networks. *New J Phys* (2018) 20:125002. doi:10.1088/1367-2630/aaf25d

33. Li H, Zhang X, Zhao C. Explaining social events through community evolution on temporal networks. *Appl Maths Comput* (2021) 404:126148. doi:10.1016/j.amc.2021.126148

34. Tan F, Zhou L, Lu J, Quan H, Liu K. Adaptive quantitative control for finite time synchronization among multiplex switched nonlinear coupling complex networks. *Eur J Control* (2023) 70:2023:100764. doi:10.1016/j.ejcon.2022.100764

35. Zhu X, Wang W, Cai S, Stanley HE. Optimal imitation capacity and crossover phenomenon in the dynamics of social contagions. *J Stat Mech Theor Exp* (2018) 2018: 063405. doi:10.1088/1742-5468/aac914

36. Nian F, Yu X, Cao J, Luo L. Phase transition in information propagation on high-order networks. *Int J Mod Phys B* (2020) 34(21):2050203. doi:10.1142/s0217979220502033

37. Yu X, Yang Q, Ai K, Zhu X, Wang W. Information spreading on two-layered multiplex networks with limited contact. *IEEE Access* (2020) 8:104316–25. doi:10.1109/access.2020.2999495

38. Wang W, Liu Q, Liang J, Hu Y, Zhou T. Coevolution spreading in complex networks. *Phys Rep* (2019) 820:1–51. doi:10.1016/j.physrep.2019.07.001

39. Wang S, Lv W, Zhang J, Luan S, Chen C, Gu X. Method of power network critical nodes identification and robustness enhancement based on a cooperative framework. *Reliability Eng Syst Saf* (2021) 207:107313. doi:10.1016/j.ress.2020.107313

40. Iacopini I, Petri G, Barrat A, Latora V. Simplicial models of social contagion. *Nat Commun* (2019) 10:2485. doi:10.1038/s41467-019-10431-6

41. Cimini G, Squartini T, Saracco F, Garlaschelli D, Gabrielli A, Guido C. The statistical physics of real-world networks. *Nat Rev Phys* (2019) 1:58–71. doi:10.1038/s42254-018-0002-6

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 23 August 2023
DOI 10.3389/fphy.2023.1253642

OPEN ACCESS

EDITED BY
Viet-Thanh Pham,
Ton Duc Thang University, Vietnam

REVIEWED BY
Samaneh Soradi-zeid,
University of Sistan and Baluchestan, Iran
Oscar Castillo,
Instituto Tecnológico de Tijuana, Mexico
Fernando Serrano,
National Autonomous University of Honduras, Honduras, in collaboration with reviewer OC

*CORRESPONDENCE
Hajid Alsubaie,
h.alsubaie@tu.edu.sa

RECEIVED 05 July 2023
ACCEPTED 07 August 2023
PUBLISHED 23 August 2023

CITATION
Alsubaie H (2023), A neural state-space-based model predictive technique for effective vibration control in nano-beams.
Front. Phys. 11:1253642.
doi: 10.3389/fphy.2023.1253642

A neural state-space-based model predictive technique for effective vibration control in nano-beams

Hajid Alsubaie*

Department of Mechanical Engineering, College of Engineering, Taif University, Taif, Saudi Arabia

Model predictive control (MPC) is a cutting-edge control technique, but its susceptibility to inaccuracies in the model remains a challenge for embedded systems. In this study, we propose a data-driven MPC framework to address this issue and achieve robust and adaptable performance. Our framework involves systematically identifying system dynamics and learning the MPC policy through function approximations. Specifically, we introduce a system identification method based on the Deep neural network (DNN) and integrate it with MPC. The function approximation capability of DNN enables the controller to learn the nonlinear dynamics of the system then the MPC policy is established based on the identified model. Also, through an added control term the robustness and convergence of the closed-loop system are guaranteed. Then the governing equation of a non-local strain gradient (NSG) nano-beam is presented. Finally, the proposed control scheme is used for vibration suppression in the NSG nano-beam. To validate the effectiveness of our approach, the controller is applied to the unknown system, meaning that solely during the training phase of the neural state-space-based model we relied on the data extracted from the time history of the beam's deflection. The simulation results conclusively demonstrate the remarkable performance of our proposed approach in effectively suppressing vibrations.

KEYWORDS

model predictive control, data-driven MPC, nano system, robust control, NSG theorem

1 Introduction

Neural networks have brought about substantial changes in the handling of nonlinear systems, holding immense potential to revolutionize the control field. Their unique ability to model and interpret complex, high-dimensional dynamics positions them as key contributors in areas where traditional mathematical models typically face challenges [1, 2]. State-space models based on neural networks are capable of mapping the intricate relationships between the inputs, outputs, and internal states of nonlinear systems, using their capacity to approximate any continuous function [3, 4]. They utilize past and current data, learning the nonlinear dynamics, to predict the future states of a system based on the present state and control inputs.

MPC is a highly effective control approach widely employed in diverse engineering domains to achieve superior control performance compared to conventional methods [5, 6]. By utilizing a predictive model of the system, MPC optimizes control actions over a finite time horizon. MPC enables the consideration of future system behavior and constraints, allowing for more precise and robust control actions [7, 8]. Hence, to now, MPC has been

widely used in various engineering domains to achieve superior control performance compared to conventional methods [9–12].

Nanostructures, including nano-beams, have generated considerable attention across a range of disciplines due to their superior mechanical attributes and the profound potential they possess for advancements in nanotechnology applications. Up to this point, an expansive corpus of research has been established within this particular field of study. Undeniably, it is of utmost importance to sustain these scholarly pursuits. Doing so will not only enhance our comprehension but will also enable us to leverage these findings more effectively for the greater benefit. For example, the study by Ohashi et al. [13] underscores the necessity for the stable delivery of nano-beams in facilitating advanced nanoscale analyses. However, the diminutive dimensions of these structures pose unique challenges pertaining to their dynamic behavior, notably when exposed to vibrational forces [14, 15].

Given the escalating demand for nanotechnology across diverse sectors, from medicine to information technology, it is clear that this area of research requires continued exploration and development [16–18]. However, the miniaturized scale of these structures brings forth distinct challenges related to their dynamic behavior, especially when exposed to vibrations. Consequently, the study and control of nano-beams have emerged as an integral field of study, aiming to ensure the reliability and operational efficacy of nano-devices. This focus is evident in the comprehensive review by Roudbari et al. [19], which emphasizes the significance of size-dependent continuum mechanics models for micro and nano-structures. Similarly, the research by Miandoab et al. [20] offers a nonlocal and strain gradient-based model for electrostatically actuated silicon nano-beams, thereby addressing specific control issues inherent in such structures.

Sliding mode controllers [21, 22] and other robust controllers [23] have been extensively investigated and suggested for nano and microsystems. However, when it comes to control in nano and microsystems, the application of MPC has not been adequately proposed. The main reason behind this limitation is the substantial amount of uncertainties present in these systems. Unlike other control methods, MPC relies on having an accurate model of the system, which is practically impossible to obtain in real-world nano and microsystems. Therefore, despite the potential advantages of MPC, its practical implementation in this domain remains unfeasible. As researchers continue to explore novel control approaches, finding ways to overcome these challenges and devise MPC strategies for nano and microsystems will be essential.

The quest for optimally controlling nonlinear and uncertain systems is a formidable challenge in modern control theory, where traditional methods like MPC and robust control present significant advantages but also face limitations. MPC's high computational costs and reliance on accurate system models make it less suited for real-time applications and systems with complex, uncertain dynamics. On the other hand, robust control handles uncertainties [24, 25] but often leads to suboptimal performance and does not directly account for state and control constraints. For example, in [26], MPC was suggested as a method for atomic force microscopy. However, this control strategy relies on the assumption that a complete and perfectly accurate system model is available, which is often not a reality in actual practice due to the unpredictability and complexity inherent in real-world scenarios.

In practice, obtaining a fully accurate model of the system is challenging due to inherent uncertainties and practical limitations. Thus, the assumptions made in the design of the controller do not hold true in practical applications. This emphasizes the need to develop control strategies that can effectively handle the uncertainties and limitations present in nano and microsystems without relying on perfect system models.

Recently, data-driven methods, as presented in studies such as [27–29] promise a more efficient and adaptive approach. These methods leverage machine learning to learn system dynamics and control policies, reduce the computational burden, and adapt to system changes. For instance, Li and Tong [30] applied an encoder-decoder neural network model for developing an MPC. Their focus was on the efficient control of an HVAC system, and their results showcased promising convergence. Also, in a more recent study, Bonassi et al. [31] offered a comprehensive discussion on the integration and evaluation of various recurrent neural network structures within the framework of MPC.

Despite the aforementioned advancements, some problems persist in the majority of studies within this field. Most notably, there is a consistent lack of guaranteed convergence and stability, which presents significant challenges for the advancement of machine learning-based MPC solutions. Hence, more research is needed to refine data-driven approaches for optimal control of nonlinear and uncertain systems, focusing on their performance, computational efficiency, robustness, convergence, generalizability, and data requirements. This challenge has served as a significant motivation for our current study. Recognizing the limitations of existing control techniques, particularly in the context of nano and microsystems, we are driven to explore innovative approaches that can overcome the hurdles associated with uncertainties in these systems. Through our study, we aspire to pave the way for practical implementation and real-world applications of advanced control methods in the realm of nano and microsystems.

We propose a neural state space-based model predictive control by integration of DNN with MPC. DNNs have the remarkable ability to learn complex patterns and capture intricate relationships from data [32, 33]. Therefore, we utilize Deep Neural Networks (DNNs) as neural state space models for the systems. Through training on accessible data, DNNs can construct nonlinear models that effectively approximate the system's behavior, even when uncertainties and disturbances are present. This provides a valuable advantage when dealing with nano-beam vibrations, where comprehensive knowledge of the system's dynamics may be elusive. The integration of DNNs with MPC enables the development of an intelligent control framework that effectively compensates for the limitations of MPC and suppresses vibrations in NSG nano-beams. In this study, we enhance the control strategy by integrating an additional control term, ensuring the robustness of the controller and promoting the convergence of the closed-loop system to the desired value. This synergistic combination of DNNs and MPC acts as a corrective component, elevating the stability and performance of the control system.

The structure of this paper is as follows: Section 2 offers a comprehensive introduction to the fundamental concepts and principles, setting the groundwork for our proposed framework. Subsequently, we present and validate our framework in subsequent sections. Section 3 focuses on the governing equations of the NSG

nano-beam, taking into account its unique characteristics. In Section 4, we apply the proposed controller to the nano-beam and thoroughly investigate its performance through simulations. Finally, in Section 5, we present the concluding remarks summarizing the key findings and suggest areas for further improvements.

2 The proposed control scheme

In this section, we present some preliminaries and our control approach. Firstly, in Section 2.1 we describe the methodology used to construct a neural state-space-based model that captures the dynamics of the system accurately. Subsequently, in Section 2.2, we delineate the MPC policy employed in our framework. We outline the optimization problem formulation and the steps involved in generating control actions over a finite time horizon. Furthermore, in Section 2.3, we introduce the robustness term that is added to enhance the controller's stability and performance. Additionally, we depict the control scheme, illustrating how the neural state-space-based MPC policy and robustness term are integrated to form a cohesive control framework.

2.1 Neural state-space models

Neural state-space models encompass a category of models that employ neural networks to depict the functions that define the nonlinear state-space representation of a system. In traditional control theory, state-space models are used to describe the behavior of dynamic systems by representing the relationship between the system's inputs, outputs, and internal states. Suppose a general state-space form with the following mathematical representation. The mathematical form of the system is given by

$$\dot{x}(t) = f_c(x(t)) + h_c(x(t))u(t) \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, and the input vector is represented by $u \in \mathbb{R}^m$. Also, f_c and h_c are two static non-linear mappings. The discrete-time formulation is given by

$$x(k+1) = f(x(k)) + h(x(k))u(k) \tag{2}$$

in linear systems, these equations are typically represented by linear functions.

Assumption 1. The system dynamics functions $f(x)$ and $h(x)$ are assumed to be Lipschitz continuous, indicating that there exists a Lipschitz constant that governs the behavior of $f(x)$ and $h(x)$ as follows

$$\| f(\bar{x}(k)) + h(\bar{x}(k))\bar{u}(k) - f(x(k)) + h(x(k))u(k) \| \leq \varepsilon_x \| \bar{x} - x \| + \varepsilon_u \| \bar{u} - u \| \tag{3}$$

in which $\varepsilon_x, \varepsilon_u \geq 0$ are constants values.

However, in many real-world scenarios, systems exhibit nonlinear behavior that cannot be accurately captured by linear models. Neural networks offer a powerful framework for representing and learning nonlinear relationships [34] making them well-suited for constructing state-space models for such systems. Here, we introduce a neural state-space model, where the state equation is represented by neural networks. The neural network represents the function that describes the behavior of the system's states, although here we used DNN, these networks can be designed as recurrent neural networks (such as LSTM or GRU), or other types of architectures depending on the characteristics of the system being modeled.

The DNN in the neural state-space model is trained using data from the system. This training involves optimizing the network parameters to minimize the discrepancy between the model's predictions and the observed behavior of the system. Various techniques, such as gradient descent or backpropagation, can be employed for this purpose. Once trained, the neural state-space model is used to simulate the behavior of the system, estimate its internal states based on available inputs and outputs, and predict future system responses. The mathematical form of the learned state space is given by

$$\hat{x}(k+1) = \hat{f}(\hat{x}(k)) + \hat{h}(\hat{x}(k))u(k) \tag{4}$$

where the variables $\hat{x}(k)$ and $u(k)$ correspond to the baseline state and control input, respectively, for the baseline model. The function $\hat{f}(\hat{x}(k))$ represents the baseline model dynamics (here is the neural network).

It is noteworthy that the learned state space model as represented in Eq. 4 can be backpropagated, and its derivatives are computable through the application of automatic differentiation. Here we assume $\hat{f}$ and $\hat{h}$ satisfy the conditions of Lipschitz continuity and general continuity. This assumption of Lipschitz continuity and general continuity for the dynamic functions is widely recognized in the field. The current study also acknowledges and incorporates this fundamental premise.

2.2 Nonlinear MPC for the baseline model

By data-driven NMPC we refer to establishing MPC policies based on the learned neural state space-based model.

The cost function of MPC associated with the neural state space model 4) is defined as follows

$$J_N = \sum_{k=0}^{N-1} Q(\hat{x}(k), u(k)) + T(\hat{x}(N)), \tag{5}$$

The cost function for plant 1) is determined by considering several factors. It incorporates the stage cost, represented by $Q(\hat{x}(k), u(k))$, which takes into account the current state x and control input u at each stage. Additionally, there is a terminal cost component denoted by $T(\hat{x}(N))$, which captures the cost associated with the final state. In the current study, the terminal cost was not applied, despite its mention within the theoretical formulations. This was intended to preserve the generality of the presentation. Also, The cost function is defined over a prediction horizon N, encompassing the control inputs $u(0, 1, \ldots, N-1)$, state variables $\hat{x}(0, 1, \ldots, N)$. Eq. 5 illustrates the system in its general form; however, the controller proposed in this study has been specifically designed for affine systems. By employing the baseline model 4) and initiating from an initial state $\hat{x}(0) = \hat{x}_0$, the data-driven NMPC is introduced. This approach aims to minimize the

cost function 5) with respect to the baseline model 4) which ius given by

$$\begin{aligned}(\hat{x}^o, u^o) &= \arg\min_{\bar{x},\bar{u}} J_N(\hat{x}, u)\\ \text{s.t.}\quad \hat{x}(k+1) &= \hat{f}(\hat{x}(k)) + \hat{h}(\hat{x}(k))u(k)\\ \hat{x}(0) &= \hat{x}_0.\end{aligned} \tag{6}$$

the augmented cost function, denoted as J, expands the original cost function 5) to include the constraints and Lagrange multipliers. It is obtained by integrating the Hamiltonian function across the prediction horizon N. Therefore, the augmented cost function can be expressed as:

$$\begin{aligned}C(k) &= Q(\hat{x}(k), u(k)) + \lambda^T(k+1)\left[\hat{f}(\hat{x}(k)) + \hat{h}(\hat{x}(k))u(k)\right]\\ \bar{J}_N &= \sum_{k=0}^{N-1}\left(C(k) - \lambda^T(k+1)\hat{x}(k+1)\right) + T(\hat{x}(N)),\end{aligned} \tag{7}$$

In this context, $\lambda(k+1)$ represents the Lagrange multiplier associated with the dynamics of the baseline model 4). It is worth noting that these Lagrange multipliers, also known as co-states, play a significant role. By solving the NMPC problem (Eq. 6), we obtain the optimal trajectories for the baseline denoted as $\hat{x}^o(k)$ and $u^o(k)$. During this optimization process, the Karush-Kuhn-Tucker (KKT) conditions are employed to derive the necessary optimality conditions, ensuring that the augmented cost function 7) is effectively minimized. These optimality conditions can be expressed as follows:

$$C_u(k) = 0, k = 0, 1, \ldots, N-1 \tag{8-a}$$

$$\lambda(k) = C_{\hat{x}}(k), k = 0, 1, \ldots, N-1 \tag{8-b}$$

$$\lambda(N) = T_{\hat{x}}(\hat{x}(N)), k = 0, 1, \ldots, N-1 \tag{8-c}$$

By utilizing the KKT conditions and the Lagrange multipliers $\lambda(k+1)$ as well as the baseline optimal solution $\hat{x}^o(k)$ and $u^o(k)$ can be computed online. Taking into account the KKT conditions, we have the following expressions:

$$\begin{aligned}C_{\hat{u}}(k) &= Q_{\hat{u}}(\hat{x}^o(k), u^o(k)) + \lambda^T(k+1)\\ &\left[\hat{f}_u(x^o(k)) + \hat{h}_u(\hat{x}^o(k))u^o(k)\right] = 0\end{aligned} \tag{9-a}$$

$$\lambda(k) = Q_{\hat{x}}(\hat{x}^o(k), u^o(k)) + \lambda^T(k+1)\left[\hat{f}_{\hat{x}}(x^o(k)) + \hat{h}_{\hat{x}}(\hat{x}^o(k))u^o(k)\right], \tag{9-b}$$

$$\lambda(N) = T_{\hat{x}}(\hat{x}^o(N)). \tag{9-c}$$

By considering Eqs 8, 9, the Lagrange multipliers $\lambda(k+1)$ can be obtained through the following calculations:

$$\lambda(k+1) = Q_{\hat{x}}(k) + \lambda^T(k+1)\hat{f}_{\hat{x}}(k), \tag{10}$$

Additionally, the Lagrange multipliers (10) are considered to be the baseline optimal Lagrange multipliers denoted as $\lambda^o(k)$.

Consider the scenario involving system 1), baseline model 4), and data-driven NMPC 6). The data-driven NMPC yields an optimal solution denoted by $\hat{x}^o(k)$ and $u^o(k)$, which serves as the baseline solution. However, due to an error between the real systems and the baseline systems a perturbation $\tilde{N}(x,t)$ may arise as a result. If this perturbation does not affect the status of the constraints, the optimal solution for system 1) can be adjusted as $u = u_\delta(k) + u^o(k)$. The focus now lies on devising a robust NMPC framework capable of effectively addressing the challenges arising from unknown bounded disturbances and errors in the neural state space model. This objective is explicitly articulated in the following part.

2.3 Robust NMPC with guaranteed convergence

We define the error of the system as $e = x - x_d$, where x_d represents the desired reference trajectory. The robust control tracking control law can be expressed as follows:

$$\begin{aligned}u_\delta &= -\frac{u_r}{\hat{h}(x)} = -\frac{u_r}{h(x)+\tilde{M}}\\ u_r &= \hat{f}_c(\hat{x}(t)) + \dot{x}_d + \delta e + \mu\, sign(e)\\ \tilde{M} &= \hat{h}(x) - h(x)\end{aligned} \tag{11}$$

Here, δ and μ are user-defined parameters that need to be positive.

Theorem 1. Assuming that the compound uncertainty remains within established boundaries, the proposed control law, as described in Eq. 12, in conjunction with the MPC defined in Eq. 6 and derived from the neural state-space model in Eq. 4 handles residual errors in the tracking control, and guarantees the convergence of the states in the state-space model 1) towards the desired values.

Proof. Suppose that the error arising from the estimation of the system's dynamics and the error in the baseline initial condition can be combined into a single term denoted as $\tilde{N} = f_c(x) + h_c(x)u^o - \hat{f}(\hat{x}) + \hat{h}(\hat{x})u^o$. By substituting Eq. 11 and Eq. 1 in the time derivative of the defined error, we achieve:

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{x}_d = f_c(x(t)) + h_c(x(t))u(t) - \dot{y}_d\\ &= f_c(x(t)) + h_c(x(t))(u_\delta + u^o) - \dot{x}_d\\ &= f_c(x(t)) + h_c(x(t))u_\delta + \hat{f}(\hat{x}) + \hat{h}(\hat{x})u^o + \tilde{N} - \dot{x}_d\end{aligned} \tag{12}$$

Now due to the optimality of u^o we know after a shoer period of time $\hat{f}(\hat{x}) + \hat{h}(\hat{x})u^o = 0$, also we know $h(x)\frac{u_r}{h(x)+\tilde{M}} = u_r - \frac{\tilde{M}}{h(x)+\tilde{M}}u_r$. Therefore we have

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{x}_d\\ &= f_c(x(t)) - \hat{f}_c(\hat{x}(t)) + \frac{\tilde{M}}{h(x)+\tilde{M}}u_r - \delta e - \mu\, sign(e) + \tilde{N}\end{aligned} \tag{13}$$

We define $f_c(x(t)) - \hat{f}_c(\hat{x}(t)) + \frac{\tilde{M}}{h(x)+\tilde{M}}u_r = \tilde{M}_d$ and substitute in Eq. 13 which results in

$$\dot{e} = \dot{x} - \dot{x}_d = \tilde{M}_d - \delta e - \mu\, sign(e) + \tilde{N} \tag{14}$$

Now, let us consider a Lyapunov function candidate denoted as $V(x)$, which is expressed as:

$$V = \frac{1}{2}e^2 \geq 0 \tag{15}$$

The derivative of the Lyapunov function $V(x)$ with respect to time is expressed as

$$\dot{V} = e\dot{e} = e\left(\tilde{M}_d - \delta e - \mu\, sign(e) + \tilde{N}\right) \leq -\delta e^2 + \left(\tilde{M}_d + \tilde{N}\right)e - \mu|e| \tag{16}$$

parameters μ should be selected in a way that $\mu > |\tilde{M}_d + \tilde{N}|$, as a result, we have

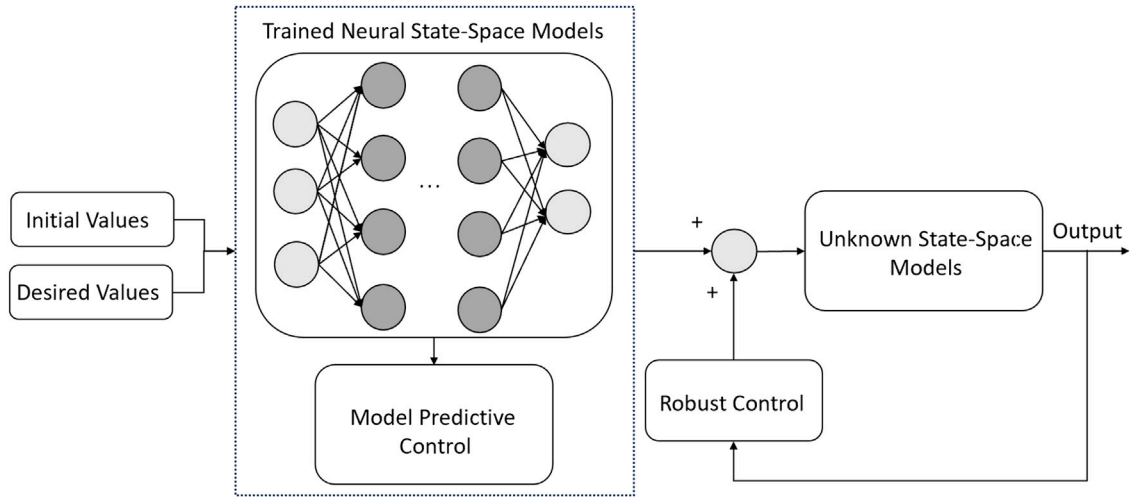


FIGURE 1
The proposed neural state-space-based model MPC.

$$\dot{V} = e\dot{e} = e\left(\tilde{M}_d - \vartheta e - \mu\, sign\,(e) + \tilde{N}\right) \leq -\delta e^2 \tag{17}$$

By utilizing Eq. 17, we can validate that the convergence of the states of the closed-loop system towards the equilibrium point is assured, in accordance with the Lyapunov stability theorem. This result completes the proof.

Remark 1. The parameters δ and μ, defined by the user, must adhere to predefined constraints in order to ensure the validity of the results and the stability of the model. Specifically, the parameter δ is required to maintain a positive value. Furthermore, the parameter μ, besides being positive, should satisfy an additional condition, namely, that $\mu > |\tilde{M}_d + \tilde{N}|$. This criterion is of paramount importance for maintaining Lyapunov stability, as elucidated in Eq. 17.

Remark 2. The deployment of the sign function in the controller design can give rise to non-smooth control inputs, leading to undesirable chattering. A prevalent and efficacious strategy to counteract such instances involves employing a continuous approximation instead of the sign function. In this context, the arctangent (atan) function emerges as a fitting option and can be used to result in smooth control inputs.

The block diagram presented in Figure 1 illustrates the proposed control technique. It incorporates robust control in Eq. 11, allowing for the inclusion of disturbances in the model. This design choice ensures that the controller is well-suited and resilient for controlling nanobeams.

3 NSG nano-beams

The Euler-Bernoulli displacement components of a hinged-hinged nanobeam are expressed as follows:

$$\begin{aligned} u_x &= d_a(x,t) - z\frac{\partial d_t(x,t)}{\partial x} \\ u_y &= 0 \\ u_z &= d_t(x,t) \end{aligned} \tag{18}$$

The nanobeam's x, y, and z displacements are symbolized by u_x, u_y, and u_z correspondingly. The axial and transverse deflections of any point on the neutral axis are represented by d_a and d_t, respectively. The independent spatial and time variables are denoted by x and t, respectively.

Here we use the NSG theorem to present the governing equation of nanobeam. Strain gradients refer to the variation of strain within a material [35, 36]. In traditional continuum mechanics, the strain is assumed to be constant throughout the material. However, at small scales, such as in microstructures or near material boundaries, the strain may vary significantly. Strain gradients take into account this variation and introduce additional terms to the constitutive equations to capture the effect. Nonlocal effects refer to the fact that the behavior of a material at a particular point depends not only on its immediate surroundings but also on a larger region. In other words, the material's response is influenced by the overall deformation state of the neighboring points. Nonlocal effects are particularly important in materials with characteristic length scales, such as granular materials or materials with microstructural features. When both strain gradients and nonlocal effects are considered together, the resulting theory is referred to as NSG theory. It provides a more accurate description of the mechanical behavior of materials at small scales and can be used to analyze phenomena such as size-dependent plasticity, fracture, and creep in microstructures [37, 38]. The formulation for the strain energy (U) of an isotropic linear elastic material, as provided by the NSG theory, can be expressed in the following manner:

$$U = \frac{1}{2}\int_V (\sigma_{11}\varepsilon_{11} + \sigma_{11}{}^{(1)}\nabla\varepsilon_{11})dV \tag{19}$$

where σ_{11} denotes the classical stress, ${\sigma_{11}}^{(1)}$ and ε_{11} represent the stress and the normal strain. The differential operator is denoted by ∇ which is equivalent to the partial derivative with respect to x. Additionally, σ_{11} and ${\sigma_{11}}^{(1)}$ are defined as follows.

$$\sigma_{11} = \int_0^L E\xi_0 \varepsilon_{11}^{'}(x')dx' \tag{20 – a}$$

$${\sigma_{11}}^{(1)} = {l_s}^2 \int_0^L E\xi_1 \varepsilon_{11}^{'}(x')dx' \tag{20 – b}$$

$$t_{11} = \sigma_{11} - \nabla {\sigma_{11}}^{(1)} \tag{20 – c}$$

where the length of the nanobeam is symbolized by L and ξ_0 represent the principal attenuation kernel function, which combines the constitutive equations describing the nonlocal effects. l_s is the strain gradient length scale parameter, Alongside, we have ξ_1, an additional kernel function that relates specifically to the nonlocal effect. Young's modulus, denoted as E, is also a key factor in the equation. The underlying assumption of nonlocal elasticity theory is that the stress at a point in a body does not only depend on the strain at that point but also depends on the strain at other points. The use of the integral in Eq. 20 represents this nonlocal behavior. The integral sums up the contributions of the strain at all points (from 0 to L, the length of the nanobeam) in the body to the stress at a particular point.

The constitutive behavior of NSG can be described by the following general equation:

$$\frac{1}{E}\left[1 - {\theta_0}^2\nabla^2\right]\left[1 - {\theta_1}^2\nabla^2\right]t_{11} = \left[1 - {\theta_1}^2\nabla^2\right]\varepsilon_{11} - {l_s}^2\left[1 - {\theta_0}^2\nabla^2\right]\nabla^2\varepsilon_{11} \tag{21}$$

where ∇^2 is Laplacian operator and the nonlocal parameters θ_0 and θ_1 are incorporated to acknowledge the significance of the nonlocal elastic stress field. These nonlocal parameters, modulate the influence of the stress field at distant points. They are typically chosen based on experimental observations or are calibrated using numerical methods to match the predictions of the nonlocal theory with the observed material behavior. Let $\theta_0 = \theta_1 = \theta$ (this is a valid assumption, for more detailed information and clarification, see [39], Eq. 21 can be reformulated as

$$\frac{1}{E}\left[1 - \theta^2\nabla^2\right]t_{11} = \left(1 - l_S^2\nabla^2\right)\varepsilon_{11} \tag{22}$$

Assuming $\boldsymbol{l_s} = \mathbf{0}$, it leads to the formulation of local elasticity theory as follows:

$$\frac{1}{E}\left[1 - \theta^2\nabla^2\right]t_{11} = \varepsilon_{11} \tag{23}$$

By setting $\theta = 0$ to zero, the strain gradient theory can be represented as follows:

$$t_{11} = E\left(1 - {l_s}^2\nabla^2\right)\varepsilon_{11} \tag{24}$$

Remark 3: It should be underscored that the assumptions of $l_s = 0$ and $\theta = 0$ are not adopted in the present study. Their inclusion here is merely illustrative, employed with the explicit intent of elucidating the interconnections between nonlocal strain gradients, strain gradients, and local elasticity. When analyzing a straight Euler-Bernoulli nanobeam under the assumptions of large deflection and small slope, the nonlinear strain relationship derived from Von Karman's theory can be represented in the following manner:

$$\varepsilon_{11} = \frac{\partial d_a(x,t)}{\partial x} + \frac{1}{2}\left(\frac{\partial d_t(x,t)}{\partial x}\right)^2 - z\frac{\partial^2 d_t(x,t)}{\partial x^2} \tag{25}$$

where ε_{11} represents the longitudinal strain, consequently one can achieve:

$$\delta\int_0^t U dt = \int_0^t\int_0^L\left[N_c\left(\frac{\partial\delta d_a}{\partial x} + \frac{\partial d_t}{\partial x}\frac{\partial\delta d_t}{\partial x}\right) - M_c\frac{\partial^2\delta d_t}{\partial x^2}\right]dxdt + \int_0^t\left[N_{nc}\left(\frac{\partial\delta d_a}{\partial x} + \frac{\partial d_t}{\partial x}\frac{\partial\delta d_t}{\partial x}\right) - M_{nc}\frac{\partial^2\delta d_t}{\partial x^2}\right]\Big|_0^L dt \tag{26}$$

in which N_{nc}, M_{nc}, N_c, and M_c are given by

$$N_{nc} = \int_A {\sigma_{11}}^{(1)}dA, M_{nc} = \int_A z{\sigma_{11}}^{(1)}dA, N_c = \int_A t_{11}dA, M_c = \int_A zt_{11}dA, \tag{27}$$

where M_c represents the classical normal moment, N_c represents the classical force, M_{nc} represents the non-classical normal moment, and N_{nc} represents the non-classical force. Furthermore, the work of the applied external forces is given by

$$\delta\int_0^t W dt = \int_0^t\int_L (f_a\delta d + f_t\delta d_t)dxdt \tag{28}$$

In the provided equation, f_a and f_t denote the distributed axial and transverse loads, respectively. Additionally, the expression for the first variation of kinetic energy (K_e) is given as follows:

$$\delta\int_0^t K_e dt = \int_0^t\int_0^L I_A\left(\frac{\partial d_a}{\partial t}\frac{\partial\delta d_a}{\partial t} + \frac{\partial d_t}{\partial t}\frac{\partial\delta d_t}{\partial t}\right)dxdt \tag{29}$$

where $I_A = \frac{1}{12}bh^3$. Note that this formulation is based on several common assumptions, which include: 1) The principle of virtual work holds true, which means that the virtual work done by the applied forces is equal to the change in kinetic and potential energy 2) The quantities d_a and d_t are assumed to be differentiable functions of time. 3) The moment of inertia I_A is constant across the length of the system. This would imply a uniform mass distribution along the system.

The given expression for Hamilton's principle, which is employed to derive the equations of motion, is as follows:

$$\delta\int_0^t [K_e - (U - W)]dt = 0 \tag{30}$$

By applying Hamilton's principle (30) and considering the rotational inertia of the beam to be negligible, we obtain the governing equation for the nanobeam according to the NSG theory. The aforementioned equation can be represented in the following manner:

$$D_{11}l_m^2\frac{\partial^6 d_t}{\partial x^6}-D_{11}\frac{\partial^4 d_t}{\partial x^4}+\left[\frac{A_{11}}{2L}\int_0^L\left(\frac{\partial d_t}{\partial x}\right)^2dx-\frac{A_{11}l_m^2}{L}\int_0^L\left(\frac{\partial d_t}{\partial x}\frac{\partial^3 d_t}{\partial x^3}+\left(\frac{\partial^2 d_t}{\partial x^2}\right)^2\right)dx\right]\times\left[\frac{\partial^2 d_t}{\partial x^2}-\theta^2\frac{\partial^4 d_t}{\partial x^4}\right]+I_A\frac{\partial^2}{\partial t^2}\left[(\theta)^2\frac{\partial^2 d_t}{\partial x^2}-d_t\right]=\theta^2\frac{\partial^2 q_t}{\partial x^2}-f_t \tag{31}$$

To render Eq. 31 in a dimensionless manner, the subsequent variables are introduced:

$$\bar{x}=\frac{x}{L},\quad \bar{d}_t=\frac{d_t}{r},\quad \bar{z}=\frac{z}{h},\quad \bar{t}=t\sqrt{\frac{EI}{\rho AL^4}},\quad \nu=\frac{\theta}{L},\quad \kappa=\frac{l_m}{L} \tag{32}$$

By substituting t $r=\sqrt{\frac{I}{A}}$, the resulting equation becomes dimensionless, and the governing equation can be expressed as follows.

$$\beta^2\bar{D}_{11}\frac{\partial^6\bar{d}_t}{\partial\bar{x}^6}-\bar{D}_{11}\frac{\partial^4\bar{d}_t}{\partial\bar{x}^4}+\left[\frac{\bar{A}_{11}}{2}\int_0^1\left(\frac{\partial\bar{d}_t}{\partial\bar{x}}\right)^2d\bar{x}-\kappa^2\bar{A}_{11}\int_0^1\left(\frac{\partial\bar{d}}{\partial\bar{x}}\frac{\partial^3\bar{d}_t}{\partial\bar{x}^3}+\left(\frac{\partial^2\bar{d}_t}{\partial\bar{x}^2}\right)^2\right)d\bar{x}\right]\frac{\partial^2\bar{d}_t}{\partial\bar{x}^2}-\left[\frac{\alpha^2\bar{A}_{11}}{2}\int_0^1\left(\frac{\partial\bar{d}_t}{\partial\bar{x}}\right)^2d\bar{x}-\nu^2\kappa^2\bar{A}_{11}\int_0^1\left(\frac{\partial\bar{d}}{\partial\bar{x}}\frac{\partial^3\bar{d}_t}{\partial\bar{x}^3}+\left(\frac{\partial^2\bar{d}_t}{\partial\bar{x}^2}\right)^2\right)d\bar{x}\right]\frac{\partial^4\bar{d}_t}{\partial\bar{x}^4}+\nu^2\bar{I}_A\frac{\partial^4\bar{d}_t}{\partial\bar{x}^2\partial\bar{t}^2}-\bar{I}_A\frac{\partial^2\bar{d}_t}{\partial\bar{t}^2}=\nu^2\frac{\partial^2\bar{q}}{\partial\bar{x}^2}-\tau \tag{33}$$

Given that we are dealing with a homogeneous nanobeam, it can be demonstrated that $\bar{A}_{11}=1$, $\bar{D}_{11}=1$, and $\bar{I}_A=1$ (see [40] for detailed information). Now, we can employ the Galerkin approach to convert the partial differential equation into a nonlinear ordinary differential equation. This procedure entails separating the temporal and spatial components of $\bar{d}_t(\bar{x},\bar{t})$ as outlined in [41].

$$\begin{aligned}\bar{d}_t(\bar{x},\bar{t})&=\varrho(\bar{x})\psi(\bar{t})\\ \varrho(\bar{x})&=sin(\pi\bar{x})\end{aligned} \tag{34}$$

Within the provided context, $\psi(\bar{t})$ signifies the temporal component that is yet to be determined, whereas $\varrho(\bar{x})$ represents the spatial component of the transverse deflection. Note that the mode shape $\varrho(\bar{x})=sin(\pi\bar{x})$, is a common choice for a beam that is hinged, or simply supported, at both ends. The primary rationale behind this selection is based on the boundary conditions of a simply supported beam and the mathematical properties of the sine function. Additionally, the concentrated force $\bar{q}(\bar{x},\bar{t})$ is defined as follows.

$$\bar{q}(\bar{x},\bar{t})=f_t(\bar{t})\delta\left(\bar{x}-\frac{1}{2}\right), \tag{35}$$

By combining Eqs 32, 33 with Eq. 38, and subsequently multiplying both sides of Eq. 38 by the spatial component $\varrho(\bar{x})$, and integrating over the length of the beam, an intriguing transformation is obtained. This transformation leads us to an ordinary differential equation as follows

$$\ddot{\psi}(\bar{t})+\beta_1\psi(\bar{t})+\beta_2\psi^3(\bar{t})=b\tau(\bar{t}) \tag{36}$$

while the coefficients β_1 and β_2 are determined by the following expressions

$$\begin{aligned}\beta_1&=\frac{\beta^2\bar{D}_{11}\int_0^1\varrho^{(6)}\varrho d\bar{x}-\bar{D}_{11}\int_0^1\varrho^{(4)}\varrho d\bar{x}}{\alpha^2\int_0^1\varrho''\varrho d\bar{x}-\int_0^1(\varrho)^2d\bar{x}}\\ \beta_2&=-\frac{\frac{\bar{A}_{11}}{2}\int_0^1(\varrho')^2d\bar{x}.\int_0^1\varrho''\varrho d\bar{x}-\kappa^2\bar{A}_{11}\int_0^1\varrho'''\varrho'd\bar{x}.\int_0^1\varrho''\varrho d\bar{x}-\kappa^2\bar{A}_{11}\int_0^1(\varrho'')^2d\bar{x}.\int_0^1\varrho''\varrho d\bar{x}}{\nu^2\int_0^1\varrho''\varrho d\bar{x}-\int_0^1(\varrho)^2d\bar{x}}\\ &\quad-\frac{\frac{\nu^2\bar{A}_{11}}{2}\int_0^1(\varrho')^2d\bar{x}.\int_0^1\varrho^{(4)}\varrho d\bar{x}-\nu^2\kappa^2\bar{A}_{11}\int_0^1\varrho'''\varrho'd\bar{x}.\int_0^1\varrho^{(4)}\varrho d\bar{x}-\nu^2\kappa^2\bar{A}_{11}\int_0^1(\varrho'')^2d\bar{x}.\int_0^1\varrho^{(4)}\varrho d\bar{x}}{\nu^2\int_0^1\varrho''\varrho d\bar{x}-\int_0^1(\varrho)^2d\bar{x}}\\ b&=-(\nu^2\pi^2+1)\end{aligned} \tag{37}$$

In the given context, $\varrho^{(i)}$ represent the i^{th} derivative of ϱ with respect to time. On the other hand, ϱ' refers to the first derivative of ϱ with respect to $\bar{x}$. Taking into account that $\psi(\bar{t})=x_1, \dot{\psi}(\bar{t})=\dot{x}_1=x_2$, we derive the following non-dimensional state-space equation of motion:

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=-\beta_1x_1-\beta_2(x_1)^3-b\tau(\bar{t})\end{cases} \tag{38}$$

where x_1 denotes the non-dimensional deflection of the beam and x_2 represents its derivative.

4 Numerical results

Herein, we present the numerical simulation showcasing the stabilization of a nanobeam through the implementation of the proposed control scheme. The parameters used for the simulation of the nanobeam are $\nu=\kappa=0.1$. Considering the formulation in Appendix, we obtain the corresponding values of $\beta_1=97.4$, $\beta_2=-19.97$, and $b=1.09$, based on the given ν and κ values. The criteria for the design parameters of the controller are detailed in Remark 1. For the numerical simulations here, we have chosen the parameters such that μ equals 10 and δ equals 1.

To generate training data, we employed random inputs to stimulate the system, measuring and recording both the deflection and its derivative. Subsequently, the collected training data was used to train the neural network offline. For training, we employed 200-time histories of deflection and its corresponding derivative. An example of this 200-time history samples used in the training phase can be seen in Figure 2. In this study, random inputs have been utilized to facilitate the learning of the neural state space representation of the model. The core rationale behind this selection pertains to the enhancement of the model's generalization capabilities. By deploying random inputs, we can expose the model to a more extensive and varied spectrum of data, thereby augmenting the robustness of the learning process. This strategy ensures that the model experiences a wide variety of situations during the training phase, equipping it with the ability to better

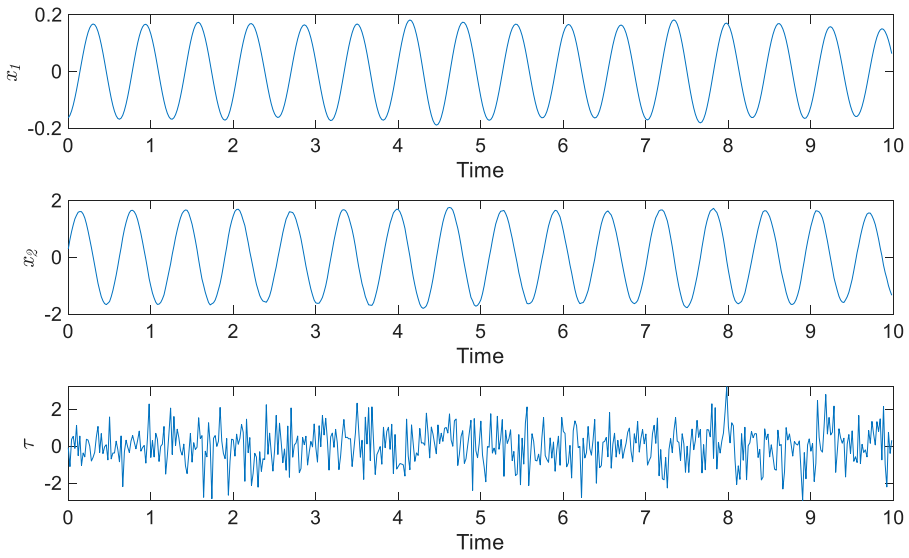


FIGURE 2
A training time history used for training of state space neural network.

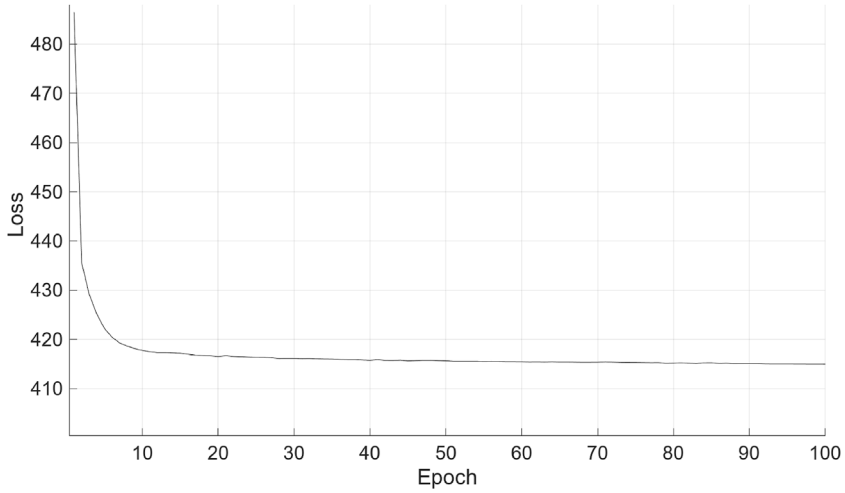


FIGURE 3
The loss function of neural state space model during training.

adapt to unforeseen scenarios when it is subsequently implemented in a real-world context. Utilizing a specific or limited type of input data for training could lead to the development of a bias in the model towards this data. This bias could adversely affect the model's performance when presented with diverse data or scenarios. To circumvent this potential bias and guarantee the broad generalizability of our model, we have chosen to employ random inputs.

The software used for the simulations is MATLAB 2022a. In the initial phase of model learning, the computational cost is primarily dependent on the number of training samples. However, considering the low-dimensional nature of the system, these costs are relatively moderate compared to typical regression and classification problems tackled by feed-forward neural networks. Once the state-space model has been learned, the computational expenditure for implementing the controller aligns with that of a typical MPC application. Hence, while the pre-training phase causes additional computational costs, the operational costs of the controller do not significantly deviate from conventional MPC approaches.

Figure 3 illustrates the loss function of the neural network training for the neural state-space model representation of the system. This loss function provides insight into the optimization process of the neural network. By monitoring the loss function, we can assess the progress and convergence of the training process,

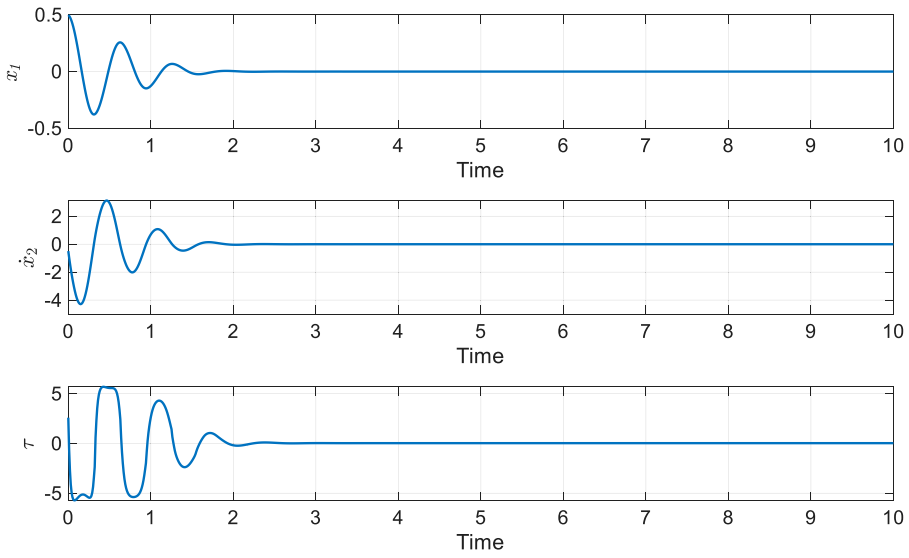


FIGURE 4
Time history of states and control input of the system while $[x_1(0), x_2(0)] = [0.5, 0]$.

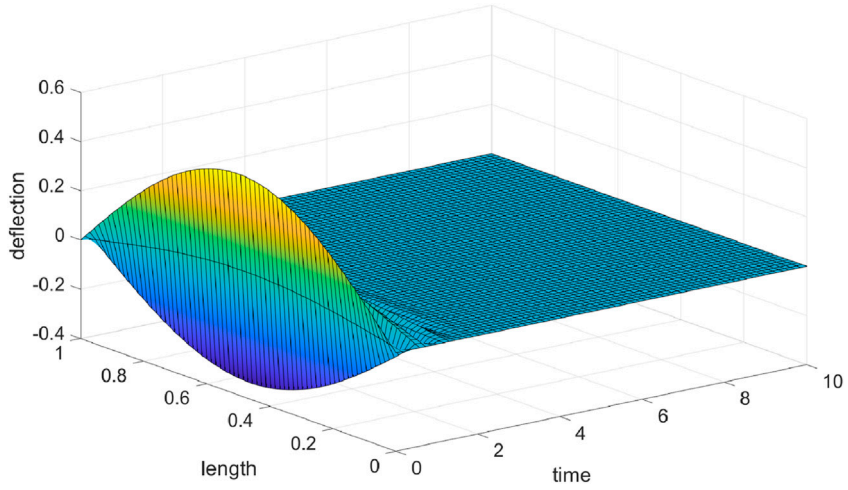


FIGURE 5
The deflection of NSG nano-beam while $[x_1(0), x_2(0)] = [0.5, 0]$.

ensuring that the neural network captures the essential dynamics of the system accurately.

In what follows, two distinct situations have been taken into account, and the proposed controller has been implemented for each. The reasoning behind having chosen two different initial conditions — $[x_1(0), x_2(0)] = [[0.5, 0]$ and $[x_1(0), x_2(0)] = [-0.5, 0.1]$ — has been to accommodate a wide range of situations. By selecting these initial conditions, we have been able to comprehensively explore both positive and negative initial positions. Furthermore, these conditions have also allowed us to investigate the impact of zero and non-zero speeds at the starting point.

Figure 4 display the outcomes of the stabilization process for the nanobeam, employing the suggested control technique with the initial states of the system set as $[x_1(0), x_2(0)] = [0.5, 0]$. These figures vividly exhibit the remarkable capability of the proposed robust adaptive controller to effectively counteract and completely reject disturbances. Additionally, Figure 5 showcases the temporal evolution of the nanobeam's deflection when utilizing the proposed control scheme. The figures demonstrate that the controller, which integrates a neural state-space model equipped with a robust term estimator, adeptly addresses control problems of the unknown system. This particular ability to handle uncertainties plays a crucial role in controlling nano systems.

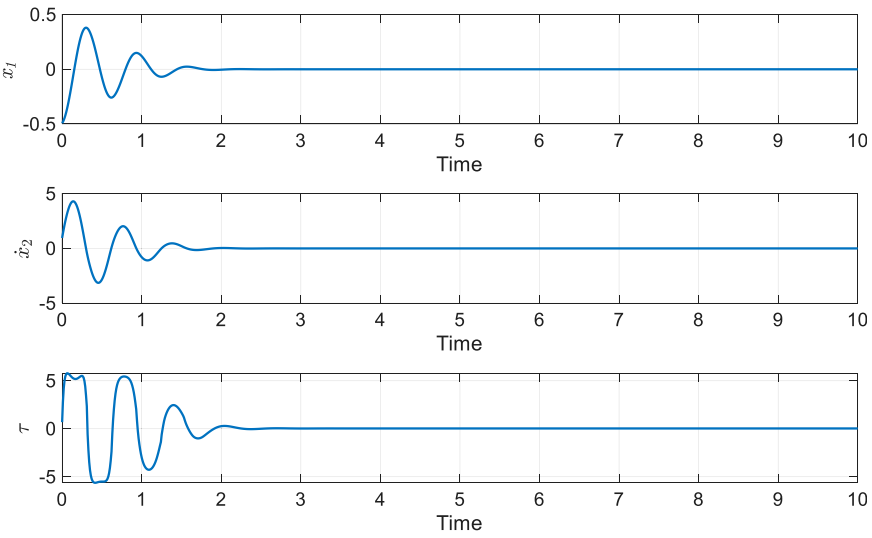


FIGURE 6
Time history of states and control input of the system while $[x_1(0), x_2(0)] = [-0.5, 0.1]$.

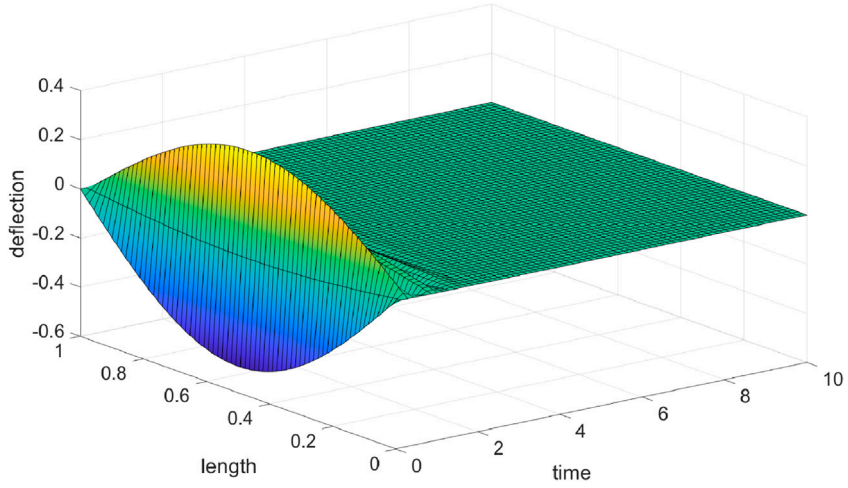


FIGURE 7
The deflection of NSG nano-beam while $[x_1(0), x_2(0)] = [-0.5, 0.1]$.

To evaluate the effectiveness of our proposed controller, we performed an additional test by varying the initial values of the system's states. Specifically, we selected $[x_1, x_2] = [-0.5, 0.1]$. Figure 6 illustrates the controller's performance in achieving system stabilization under these modified conditions. The figure demonstrates that the proposed controller operates within an acceptable range, ensuring feasible control signal values. Notably, despite the system's dynamics being completely unknown, the proposed controller exhibits outstanding performance, achieving state stabilization in less than 2 time units. Figure 7 illustrates the deflection of the system, clearly indicating that the suggested controller facilitates rapid vibration suppression in the nano beam.

To facilitate a more in-depth assessment of the proposed controller's efficacy, Table 1 outlines the settling time, as well as the maximum and norm of control signals for both numerical instances illustrated in this section.

In summary, the simulations and numerical results presented in Table 1 clearly demonstrate that the proposed control scheme excels in vibration suppression in the nanobeam with completely unknown dynamics, ensuring the stability and robustness of the system. Compared to conventional MPC and robust controls [42, 43], our method provides significant advantages in handling nano-beam vibrations, especially when full knowledge of the system's dynamics is not readily available. By combining DNNs with MPC, we develop an intelligent control framework that effectively mitigates MPC's limitations and reduces vibrations in NSG nano-

TABLE 1 Settling time and control signal properties.

Example	Settling time	Maximum of the control signal	Norm of the control signal
Example 1, $[x_1(0), x_2(0)] = [0.5, 0]$	1.350	5.07	75.06
Example 2, $[x_1(0), x_2(0)] = [-0.5, 0.1]$	1.551	5.25	76.57

beams. We further enhance the control strategy by introducing an extra control term for robustness and improved system convergence. However, it is important to note that our method requires pre-processing and data collection for training the model before real-world deployment, unlike traditional approaches.

5 Conclusion

The present study introduced a neural state-space-based MPC framework with guaranteed convergence. The framework entailed a systematic identification of system dynamics and the learning of the MPC policy through function approximations. Specifically, the system dynamics were captured utilizing DNN, and the MPC policy was established based on the identified model. Additionally, the robustness and convergence of the closed-loop system were guaranteed by incorporating an additional control term. Subsequently, the governing equation of motion for NSG nano-beams was presented and derived. Then, the proposed control technique was validated by applying it to NSG nano-beams. The obtained results exhibited exceptional performance, confirming the efficacy of the proposed method. In this study, the robust control term has been consistently applied in conjunction with the optimal control term at all stages. Nevertheless, there are ways to further streamline the system without compromising accuracy. Incorporating event-triggered approaches could be beneficial in this regard. These strategies would enable the controller to be deployed only as required and then deactivated afterward, creating a more optimal control. Therefore, a potential area for future research in this domain would be to enhance the proposed controller's efficiency through the integration of event-trigger mechanisms.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

HA: Writing–original draft.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Zhu Q, Liu Y, Wen G. Adaptive neural network control for time-varying state constrained nonlinear stochastic systems with input saturation. *Inf Sci* (2020) 527: 191–209. doi:10.1016/j.ins.2020.03.055

2. Ge SS, Hang CC, Lee TH, Zhang T. *Stable adaptive neural network control*. Springer Science and Business Media (2013).

3. Wu Q, Wang Y, Wang H. An extended linearized neural state space based modeling and control. *IFAC Proc Volumes* (2014) 35:975–80. doi:10.3182/20020721-6-ES-1901.00977

4. Sun W, Wang X, Cheng Y. Reinforcement learning method for continuous state space based on dynamic neural network. 2008 7th World Congress on Intelligent Control and Automation. IEEE (2008). p. 750–4.

5. Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems–A review of model predictive control (MPC). *Building Environ* (2014) 72:343–55. doi:10.1016/j.buildenv.2013.11.016

6. Camacho E, Alba C. Model predictive control [sl]: Springer science and business media. *Citado* (2013) 2:79. doi:10.1007/978-0-85729-398-5

7. Guo N, Lenzo B, Zhang X, Zou Y, Zhai R, Zhang T. A real-time nonlinear model predictive controller for yaw motion optimization of distributed drive electric vehicles. *IEEE Trans Vehicular Technol* (2020) 69:4935–46. doi:10.1109/tvt.2020.2980169

8. Shen C, Shi Y. Distributed implementation of nonlinear model predictive control for AUV trajectory tracking. *Automatica* (2020) 115:108863. doi:10.1016/j.automatica.2020.108863

9. Woo D-O, Junghans L. Framework for model predictive control (MPC)-based surface condensation prevention for thermo-active building systems (TABS). *Energy and Buildings* (2020) 215:109898. doi:10.1016/j.enbuild.2020.109898

10. Karamanakos P, Liegmann E, Geyer T, Kennel R. Model predictive control of power electronic systems: Methods, results, and challenges. *IEEE Open J Industry Appl* (2020) 1:95–114. doi:10.1109/ojia.2020.3020184

11. Arroyo J, Manna C, Spiessens F, Helsen L. Reinforced model predictive control (RL-MPC) for building energy management. *Appl Energ* (2022) 309:118346. doi:10.1016/j.apenergy.2021.118346

12. Schwenzer M, Ay M, Bergs T, Abel D. Review on model predictive control: An engineering perspective. *Int J Adv Manufacturing Technol* (2021) 117:1327–49. doi:10.1007/s00170-021-07682-3

13. Ohashi H, Yamazaki H, Yumoto H, Koyama T, Senba Y, Takeuchi T, et al. Stable delivery of nano-beams for advanced nano-scale analyses. *J Phys Conf Ser* (2013) 425: 052018. doi:10.1088/1742-6596/425/5/052018

14. Arefi M, Pourjamshidian M, Arani AG. Free vibration analysis of a piezoelectric curved sandwich nano-beam with FG-CNTRCs face-sheets based on various high-order

shear deformation and nonlocal elasticity theories. *The Eur Phys J Plus* (2018) 133:193. doi:10.1140/epjp/i2018-12015-1

15. Sheykhi M, Eskandari A, Ghafari D, Arpanahi RA, Mohammadi B, Hashemi SH. Investigation of fluid viscosity and density on vibration of nano beam submerged in fluid considering nonlocal elasticity theory. *Alexandria Eng J* (2023) 65:607–14. doi:10.1016/j.aej.2022.10.016

16. Bhatt M, Shende P. Surface patterning techniques for proteins on nano-and micro-systems: A modulated aspect in hierarchical structures. *J Mater Chem B* (2022) 10:1176–95. doi:10.1039/d1tb02455h

17. Lyshevski SE. *Nano-and micro-electromechanical systems: Fundamentals of nano-and microengineering*. CRC Press (2018).

18. Sabarianand D, Karthikeyan P, Muthuramalingam T. A review on control strategies for compensation of hysteresis and creep on piezoelectric actuators based micro systems. *Mech Syst Signal Process* (2020) 140:106634. doi:10.1016/j.ymssp.2020.106634

19. Roudbari MA, Jorshari TD, Lü C, Ansari R, Kouzani AZ, Amabili M. A review of size-dependent continuum mechanics models for micro-and nano-structures. *Thin-Walled Structures* (2022) 170:108562. doi:10.1016/j.tws.2021.108562

20. Miandoab EM, Yousefi-Koma A, Pishkenari HN. Nonlocal and strain gradient based model for electrostatically actuated silicon nano-beams. *Microsystem Tech* (2015) 21:457–64. doi:10.1007/s00542-014-2110-2

21. Cao Q, Wei DQ. Dynamic surface sliding mode control of chaos in the fourth-order power system. *Chaos, Solitons and Fractals* (2023) 170:113420. doi:10.1016/j.chaos.2023.113420

22. Jahanshahi H, Rajagopal K, Akgul A, Sari NN, Namazi H, Jafari S. Complete analysis and engineering applications of a megastable nonlinear oscillator. *Int J Non-Linear Mech* (2018) 107:126–36. doi:10.1016/j.ijnonlinmec.2018.08.020

23. Vagia M. How to extend the travel range of a nanobeam with a robust adaptive control scheme: A dynamic surface design approach. *ISA Trans* (2013) 52:78–87. doi:10.1016/j.isatra.2012.09.001

24. Jahanshahi H, Zambrano-Serrano E, Bekiros S, Wei Z, Volos C, Castillo O, et al. On the dynamical investigation and synchronization of variable-order fractional neural networks: The hopfield-like neural network model. *Eur Phys J Spec Top* (2022) 231:1757–69. doi:10.1140/epjs/s11734-022-00450-8

25. Jahanshahi H, Yao Q, Khan MI, Moroz I. Unified neural output-constrained control for space manipulator using tan-type barrier Lyapunov function. *Adv Space Res* (2023) 71:3712–22. doi:10.1016/j.asr.2022.11.015

26. Keighobadi J, Faraji J, Rafatnia S. Chaos control of atomic force microscope system using nonlinear model predictive control. *J Mech* (2017) 33:405–15. doi:10.1017/jmech.2016.89

27. Long X, He Z, Wang Z. Online optimal control of robotic systems with single critic NN-based reinforcement learning. *Complexity* (2021) 2021:1–7. doi:10.1155/2021/8839391

28. Chen Y, Tong Z, Zheng Y, Samuelson H, Norford L. Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings. *J Clean Prod* (2020) 254:119866. doi:10.1016/j.jclepro.2019.119866

29. Carlet PG, Favato A, Bolognani S, Dörfler F. Data-driven continuous-set predictive current control for synchronous motor drives. *IEEE Trans Power Electron* (2022) 37:6637–46. doi:10.1109/tpel.2022.3142244

30. Li Y, Tong Z. Model predictive control strategy using encoder-decoder recurrent neural networks for smart control of thermal environment. *J Building Eng* (2021) 42:103017. doi:10.1016/j.jobe.2021.103017

31. Bonassi F, Farina M, Xie J, Scattolini R. On recurrent neural networks for learning-based control: Recent results and ideas for future developments. *J Process Control* (2022) 114:92–104. doi:10.1016/j.jprocont.2022.04.011

32. Khan A, Sohail A, Zahoora U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. *Artif intelligence Rev* (2020) 53:5455–516. doi:10.1007/s10462-020-09825-6

33. Stanley KO, Clune J, Lehman J, Miikkulainen R. Designing neural networks through neuroevolution. *Nat Machine Intelligence* (2019) 1:24–35. doi:10.1038/s42256-018-0006-z

34. Jahanshahi H, Shahriari-Kahkeshi M, Alcaraz R, Wang X, Singh VP, Pham V-T. Entropy analysis and neural network-based adaptive control of a non-equilibrium four-dimensional chaotic system with hidden attractors. *Entropy* (2019) 21:156. doi:10.3390/e21020156

35. Hutchinson J, Fleck N. Strain gradient plasticity. *Adv Appl Mech* (1997) 33:295–361.

36. Fleck N, Muller G, Ashby MF, Hutchinson JW. Strain gradient plasticity: Theory and experiment. *Acta Metallurgica et materialia* (1994) 42:475–87. doi:10.1016/0956-7151(94)90502-9

37. Li X, Li L, Hu Y, Ding Z, Deng W. Bending, buckling and vibration of axially functionally graded beams based on nonlocal strain gradient theory. *Compos Structures* (2017) 165:250–65. doi:10.1016/j.compstruct.2017.01.032

38. Ebrahimi F, Barati MR, Dabbagh A. A nonlocal strain gradient theory for wave propagation analysis in temperature-dependent inhomogeneous nanoplates. *Int J Eng Sci* (2016) 107:169–82. doi:10.1016/j.ijengsci.2016.07.008

39. Lim CW, Zhang G, Reddy JN. A higher-order nonlocal elasticity and strain gradient theory and its applications in wave propagation. *J Mech Phys Sol* (2015) 78:298–313. doi:10.1016/j.jmps.2015.02.001

40. Alsubaie H, Yousefpour A, Alotaibi A, Alotaibi ND, Jahanshahi H. Fault-tolerant terminal sliding mode control with disturbance observer for vibration suppression in non-local strain gradient nano-beams. *Mathematics* (2023) 11:789. doi:10.3390/math11030789

41. Rhoads JF, Shaw SW, Turner KL. The nonlinear response of resonant microbeam systems with purely-parametric electrostatic actuation. *J Micromechanics Microengineering* (2006) 16:890–9. doi:10.1088/0960-1317/16/5/003

42. Do L, Korda M, Hurák Z. Controlled synchronization of coupled pendulums by koopman model predictive control. *Control Eng Pract* (2023) 139:105629. doi:10.1016/j.conengprac.2023.105629

43. Wang Y, Liu X. Model predictive position control of permanent magnet synchronous motor servo system with sliding mode observer. *Asian J Control* (2023) 25:443–61. doi:10.1002/asjc.2817

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 25 August 2023
DOI 10.3389/fphy.2023.1252568

OPEN ACCESS

EDITED BY
Viet-Thanh Pham,
Ton Duc Thang University, Vietnam

REVIEWED BY
Fuhong Min,
Nanjing Normal University, China
Hilaire Fotsin,
University of Dschang, Cameroon
Junwei Sun,
Zhengzhou University of Light Industry, China

*CORRESPONDENCE
Zhengjun Yao,
yaozj@nuaa.edu.cn

RECEIVED 04 July 2023
ACCEPTED 15 August 2023
PUBLISHED 25 August 2023

CITATION
Fu S, Wang X, Gu H, Cao X and Yao Z (2023), Multi-scroll Hopfield neural network under electromagnetic radiation and its brain-like coupling synchronization.
Front. Phys. 11:1252568.
doi: 10.3389/fphy.2023.1252568

Multi-scroll Hopfield neural network under electromagnetic radiation and its brain-like coupling synchronization

Sen Fu[1,2,3], Xia Wang[2], Haiyang Gu[3], Xiaojing Cao[2] and Zhengjun Yao[1]*

[1]College of Materials Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China, [2]Aircraft Technology Branch of Hunan Aerospace Co., Ltd., Changsha, China, [3]China Aerospace Science and Industry Corporation, Beijing, China

Multi-scroll attractors have attracted attention because of their more complex topological structures and artificially controllable attractor structures. This paper proposes a new nonvolatile magnetic-controlled memristor and uses it to simulate the effect of membrane flux changes caused by neuronal exposure to electromagnetic radiation. A series of complex chaotic phenomena are found by plotting phase diagrams, bifurcation diagrams, attractor domains and 01 tests, including multi-scroll chaotic attractors controlled by memristors, symmetric bifurcation behavior, coexistence phenomena enhanced by initial offset. The mechanisms behind them are explained through equilibrium point analysis. A dual memristive HNN (MHNN) coupling synchronization model is proposed to simulate the synchronization between regions within the human brain. The Lyapunov function of the error is constructed to prove that this coupling synchronization scheme is ultimately bounded. The feasibility of this synchronization scheme is verified by establishing a Simulink model and conducting simulation experiments.

KEYWORDS

memristor, Hopfield neural network (HNN), multi-scroll, multistability, synchronization

1 Introduction

Numerous neurophysiological and neuroanatomical studies have shown that human brain activity is closely related to the dynamic behavior of biological neurons and neural networks. In order to reveal the mystery of how the brain processes, manipulates and accesses information, scientists have conducted long-term research on the structure and working mechanism of biological neural networks and established various artificial neuron and neural network models [1–10]. Hopfield neural networks (HNN) have been widely studied for their simple mathematical form and rich dynamical behavior. Liang *et al.* investigated the long time behavior of the mild solution to delayed reaction-diffusion HNNs driven by infinite dimensional Wiener processes. They analyzed the existence, uniqueness, and stability of this system under the local Lipschitz function by constructing an appropriate Lyapunov-Krasovskii function and utilizing the semigroup theory. Pu *et al.* proposed to introduce fractional calculus to implement HNN. They implemented the fractional HNN by utilizing fractor in the form of an analog circuit and the fractional steepest descent approach. In addition, they construct the Lyapunov function to prove the stability of fractional HNN

and analyze its attractors [11]. Danca *et al.* unveiled the existence of hidden chaotic sets in a simplified HNN with three neurons. They proved that besides two stable cycles, the system also has hidden chaotic attractors and hidden chaotic transients, which converge to regular motions along the stable cycles after a relatively long lifetime [12]. In order to make HNN generate more rich and complex brain-like dynamical behaviors, more and more scholars introduce the brain-like element memristor into them [13–19].

A memristor is a nonlinear circuit element whose resistance changes in response to the current flowing through it or the voltage at both ends [20–23]. This nonlinear behavior is very similar to the plasticity of synapses in the human brain [24]. In the process of transferring action potentials, the properties of synapses also change dynamically. Therefore, in recent years, many scholars have used memristors to replace the invariant synaptic weights in HNN and proposed a series of memristive HNN (MHNN) [25–28]. For example, Leng *et al.* proposed a new circuit to emulate the Coupled Hyperbolic Memristors and utilized it to simulate the synaptic crosstalk of a HNN. With various crosstalk strengths, multi-stability, asymmetry attractors, and anti-monotonicity are observed in this MHNN [29]. Dong *et al.* proposed a novel memristive synaptic HNN with time delay, which used a memristor synapse to simulate the electromagnetic induced current caused by the membrane potential difference between two adjacent neurons. By choosing time delay and the coupling strength of memristors as bifurcation parameters, they obtained sufficient conditions of zero bifurcation and zero-Hopf bifurcation [26]. Besides simulating synapses, memristors are also used to simulate the effects of electromagnetic radiation on neurons. With the popularity of electronic products, electromagnetic radiation fills people's daily lives. In order to explore the dynamical behavior of the human brain under electromagnetic radiation, more and more scholars introduce magnetically controlled memristors into HNN and propose a series of new MHNN [30–32]. For instance, Lin *et al.* studied the chaotic dynamics of a three-neuron HNN under electromagnetic radiation stimulation, and found hidden extreme multistability that includes hyperchaos and transient chaos. In addition, they also designed a circuit based on HNN composed of commercially available electronic components to verify the theoretical analysis [33]. Wan *et al.* investigated the hidden multistability and parallel bifurcation behaviors of a HNN under the simulation of external electromagnetic radiation and dual bias currents. They also designed an equivalent analog circuit and verified the numerical simulation results by Multisim simulation and hardware experiment based on discrete electronic components [34].

In recent years, various complex dynamical behaviors have been found in both memristive synaptic weight HNN and HNN under electromagnetic radiation, including multi-scroll or multistructure chaotic attractors. Compared with general chaotic attractors, they are more attractive for their more complex topological structure and artificially controllable attractor structure. Zhang *et al.* introduced a non-ideal magnetically controlled memristor model containing a sign function into HNN, and constructed a memristive HNN model with multiple double-scroll attractors. The odd and even numbers of double scrolls can be flexibly controlled by the internal parameters of the memristor. In particular, they found the coexisting behavior induced by the initial state offset of the memristor, and the number of coexisting attractors was closely related to the total number of scrolls, and eventually tended to infinity as the total number of scrolls increased [35]. Yu *et al.* proposed a locally active memristor containing a smooth sign function and established a MHNN satisfying the Lipschitz condition by replacing the synaptic weights of HNN. From it, they found controllable multi-scroll behavior and extreme multistability. In addition, they physically implemented this multi-scroll MHNN using FPGA technology and applied it to image encryption [36]. Lai *et al.* established a novel flux-controlled memristor model using hyperbolic function series. By taking the memristor as synapses in a HNN, they constructed three MHNNs. These MHNNs can generate multi-double-scroll chaotic attractors or grid multi-double-scroll chaotic attractors, and the number of double scrolls in the attractors is controlled by the memristor [37].

Inspired by the above research status, We propose a new model of MHNN under electromagnetic radiation, from which we discover the complex dynamic behavior. The main novelty and contributions of this study are summarized as follows:

1) We propose a new memristor model with controllable number of power interruption steady states, whose memductance does not contain any polynomials.
2) By using it to simulate the effects of membrane flux changes caused by electromagnetic radiation, a new model of MHNN under electromagnetic radiation is proposed
3) A series of complex chaotic phenomena are found, including memristor-controlled multi-scroll attractors, symmetric bifurcation behaviors, initial offset boosting coexistence.
4) Inspired by inter-brain region synchronization of the human brain, we propose a dual MHNN coupling synchronization model. Through Lyapunov function and Simulink experiments, the feasibility of this synchronization scheme is verified.

The rest of this article is organized as follows. In Section 2, the memristor model is constructed. In Section 3, a MHNN under electromagnetic are constructed and analyzed. In Section 4, The dynamic numerical simulation is carried out. In Section 5, an dual coupling synchronous model is designed and proved. Section 6 summarizes the paper.

2 Novel nonvolatile memristor

2.1 Construction of memristor model

Over the past few years, a number of memristor models that can assist in the generation of multi-scroll attractors have been proposed one after another, and they are summarized in Table 1. Based on the general form of memristors, we propose a novel non-volatile memristor model that can induce controllable multi-scrolls in HNN. It can be expressed by the following equation:

$$\begin{cases} i_m = W(x)v_m \\ W(x) = a - b\left(\dfrac{x}{1+|x|}\right) \\ \dot{x} = mv_m - nh(x) \end{cases} \tag{1}$$

TABLE 1 Summary of memristors helping construct multi-scroll attractor.

References	Memristor	Internal state variable function
Reference [14]	$\begin{cases} i_m = \sin(x)v_m \\ \frac{dx}{dt} = av_m - bh(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - m\left[\sum_{i=0}^{N} tanh(n(x-(2i+1)m)) + \sum_{i=0}^{N} \tanh(n(x+(2i+1)m))\right] \\ h_2(x) = x - m\left[\sum_{i=0}^{N} tanh(n(x-2im)) + \sum_{i=0}^{N} \tanh(n(x+2im)) - \tanh(nx)\right] \end{cases}$
Reference [15]	$\begin{cases} i_m = (a+bx^2)v_m \\ \frac{dx}{dt} = cv_m - dh(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=1}^{N} sgn(x-(2i-1)) - \sum_{i=1}^{N} sgn(x+(2i-1)) \\ h_2(x) = x - sgn(x) - \sum_{i=1}^{N} sgn(x-2i) - \sum_{i=1}^{N} sgn(x+2i) \end{cases}$
Reference [35]	$\begin{cases} i_m = (a+bh(x))v_m \\ \frac{dx}{dt} = cv_m - dh(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=1}^{N} sgn(x-(2i-1)) - \sum_{i=1}^{N} sgn(x+(2i-1)) \\ h_2(x) = x - sgn(x) - \sum_{i=1}^{N} sgn(x-2i) - \sum_{i=1}^{N} sgn(x+2i) \end{cases}$
Reference [36]	$\begin{cases} i_m = (a+bh(x))v_m \\ \frac{dx}{dt} = cv_m - dh(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=1}^{M} s(n(x+2i-1)) - \sum_{i=1}^{M} s(n(x-2i+1)) \\ h_2(x) = x - s(x) - \sum_{i=1}^{M} s(n(x+2i)) - \sum_{i=1}^{M} s(n(x-2i)) \\ s(x) = \frac{x}{1+\lvert x\rvert} \end{cases}$
Reference [38]	$\begin{cases} i_m = h(x)v_m \\ \frac{dx}{dt} = -v_m \end{cases}$	$h(x) = \begin{cases} h_1(x) = -(2N-1)kc - b + \sum_{m=1}^{4N-1} k(-1)^{m-1}\lvert x-(2N-m)c\rvert \\ h_2(x) = -(2N-2)kc - b + \sum_{m=1}^{4N-3} k(-1)^{m-1}\lvert x-(2N-m-1)c\rvert \end{cases}$
Reference [39]	$\begin{cases} i_m = (p+qf^2(x))v_m \\ \frac{dx}{dt} = v_m - h(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=1}^{N} sgn(x-(2i-1)) - \sum_{i=1}^{N} sgn(x+(2i-1)) \\ h_2(x) = x - sgn(x) - \sum_{i=1}^{N} sgn(x-2i) - \sum_{i=1}^{N} sgn(x+2i) \end{cases}$
Reference [40]	$\begin{cases} i_m = kh(x)v_m \\ \frac{dx}{dt} = pv_m \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=0}^{N-1} sgn(x-(2i-1)) - \sum_{i=1}^{M} sgn(x+(2i-1)) \\ h_2(x) = x + sgn(x) - \sum_{i=0}^{N-1} sgn(x-2i) - \sum_{i=0}^{N-1} sgn(x+2i) \end{cases}$
Reference [41]	$\begin{cases} i_m = (a+b\lvert x\rvert)v \\ \frac{dx}{dt} = cv - d(x) \end{cases}$	$h(x) = \begin{cases} h_1(x) = x - \sum_{i=0}^{N-1} sgn(x-(2i-1)) - \sum_{i=1}^{M} sgn(x+(2i-1)) \\ h_2(x) = x - sgn(x) - \sum_{i=0}^{N-1} sgn(x-2i) - \sum_{i=0}^{N-1} sgn(x+2i) \end{cases}$

where $W(x)$ is the memductance of the memristor. And $h(x)$ is the function of the internal state variable of the memristor, which has two forms, $h_1(x)$ and $h_2(x)$, to choose from, as follows:

$$h_1(x) = \begin{cases} x, M = 0 \\ x - \sum_{i=1}^{M} \left(\text{sgn}(x + (2i-1)) + \text{sgn}(x - (2i-1))\right), \\ M = 1, 2, 3, \cdots \end{cases} \quad (2)$$

and

$$h_2(x) = \begin{cases} x - \text{sgn}(x), \ N = 0 \\ x - \text{sgn}(x) - \sum_{j=1}^{N} \left(\text{sgn}(x + 2j) + \text{sgn}(x - 2j)\right), \\ N = 1, 2, 3, \cdots \end{cases} \quad (3)$$

where sgn$(x)$is a symbolic function. By choosing different forms of the internal state variable function of the memristor and changing the parameters M, N, any odd or even number of scrolls can be easily induced in HNN.

2.2 Hysteresis characteristics and non-volatility

For the proposed memristor model, we first need to verify whether it satisfies the three fingerprints of a memristor [42]. Let the parameters a, b, m and n in Eq. 1 be 2.1, 0.5, 4.5 and 1.9 respectively, and take $h_1(x)$ and M = 2 as an example. Connect the memristor to an AC voltage source $V_m = A\sin(Ft)$. The voltage across the memristor and the current through it form a figure-eight-shaped pinched hysteresis loop (PHL) on the v-i plane. We take the frequency F of the AC voltage source as 30, 90 and 150 respectively, and superimpose the PHLs that appear each time on the same plane. The results are shown in Figure 1A. From the final results, it can be seen that when the frequency of the voltage source increases, the area enclosed by the PHL gradually shrinks. And when F = 150, the PHL almost shrinks into a straight line. From the test results of PHL, it can be seen that the model described by Eq. 1 is a memristor. To verify the non-volatility of the memristor, we usually need to draw a power-off plot (POP) of the memristor.

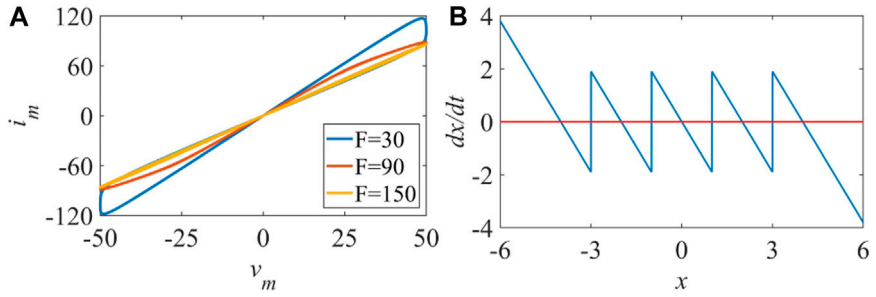


FIGURE 1
PHL and POP of nonvolatile memristor: **(A)** PHL varying with F, **(B)** POP.

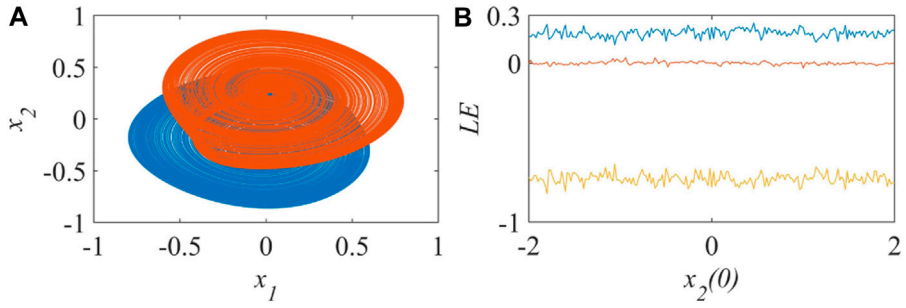


FIGURE 2
The coexistence **(A)** and the variation of Lyapunov exponent spectrum with the initial potential of neuron 2 **(B)** of the three-dimensional HNN.

When the voltage across the memristor disappears, i.e., $v_m = 0$ in Eq. 1, the dynamic equation of the internal state variable of the memristor is simplified as follows:

$$\dot{x} = mv_m - nh(x) \tag{4}$$

Taking the internal variable x as the horizontal axis and $\dot{x}$ as the vertical axis, we can draw the POP of the memristor. Figure 1B shows that there are five zeros in the POP of the memristor at this time, and the slopes at these zeros are all negative, which means that the memristor has five stable equilibrium points after power-off. Since there are more than two steady states, the memristor described by Eq. 1 has non-volatility.

3 MHNN under electromagnetic radiation

3.1 MHNN model construction

HNN has a simple mathematical form and complex dynamic behavior. In the study of chaotic dynamics, the membrane capacitance and membrane resistance of neurons are usually set to 1 to further simplify the dynamic equation of HNN. At this time, any HNN can be expressed by the following formula:

$$\dot{\mathbf{X}} = -\mathbf{X} + \mathbf{W}\tanh(\mathbf{X}) + \mathbf{I} \tag{5}$$

Where $\mathbf{X}$ is the neuron membrane potential vector, $\mathbf{I}$ is the external stimulus current vector, $\mathbf{W}$ is the weight matrix between neurons. In this paper, the weight matrix is:

$$W = \begin{bmatrix} 1.5 & 2.9 & 0.7 \\ -2 & 1.2 & 0 \\ 3 & -20 & 0.1 \end{bmatrix} \tag{6}$$

The external stimulus current is uniformly set to zero. Let the initial membrane potentials of neuron one and neuron three be 0.1, and let the initial membrane potentials of neuron two be 1 and -1 respectively. The three-dimensional HNN can exhibit complex chaotic coexistence behavior at this time, and its phase diagram is shown in Figure 2A. By continuously changing the initial membrane potential of neuron 2, we can obtain the corresponding Lyapunov exponent spectrum diagram. Figure 2B shows that when the initial membrane potential of neuron 2 takes values in [-2, 2], it finally corresponds to a chaotic attractor.With the development of electronic products, people are exposed to electromagnetic radiation more and more frequently. According to Maxwell's equations, the effect of electromagnetic radiation on a single neuron can be described by the fluctuation of magnetic flux on the cell membrane. The coupling between magnetic flux and membrane voltage can be realized by a magnetically controlled memristor [43]. Specifically for this paper, we add the memductance term of the memristor described by Eq. 1 to the dynamic equation of neuron 2 to simulate the effect of neuron 2 exposed to

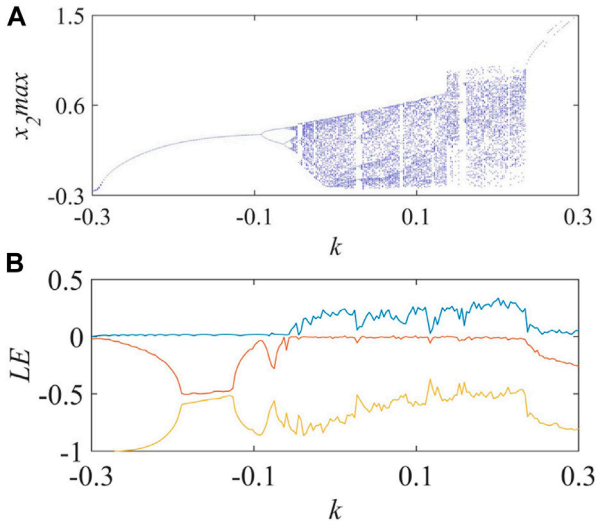


FIGURE 3
Bifurcation behavior of parameter k: **(A)** bifurcation diagram, **(B)** Lyapunov exponential spectrum.

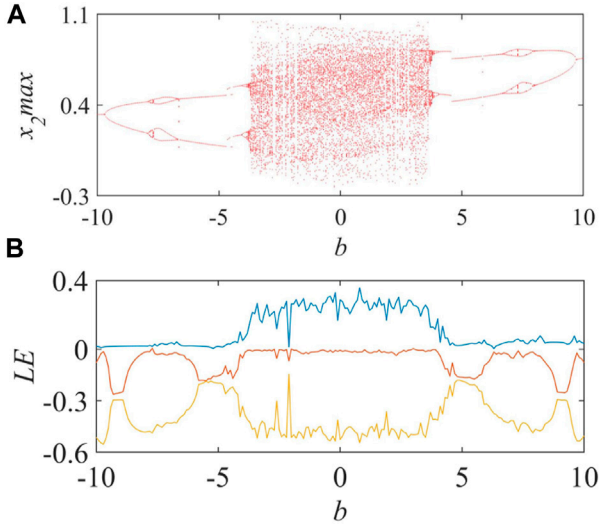


FIGURE 4
Bifurcation behavior of parameter b: **(A)** bifurcation diagram, **(B)** Lyapunov exponential spectrum.

electromagnetic radiation. At this time, the dynamic equation of the original three-dimensional HNN becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -x_1 \\ -x_2 \\ -x_3 \\ 0 \end{bmatrix} + \mathbf{W} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ kx_2\left(a - b\left(\frac{x_4}{1+|x_4|}\right)\right) \\ 0 \\ mx_2 - nh(x_4) \end{bmatrix} \tag{7}$$

where k represents the intensity of electromagnetic radiation, a, b, m and n are memristor parameters. $h(x_4)$ is the internal state variable function of the memristor, which includes Eqs 2, 3.

3.2 Equilibrium analysis

Let the left side of Eq. 7 be zero, then we get the following system of equations:

$$\begin{cases} 0 = -x_1 + 1.5\tanh(x_1) + 2.9\tanh(x_2) + 0.7\tanh(x_3) \\ 0 = -x_2 - 2\tanh(x_1) + 1.2\tanh(x_2) \\ \quad + kx_2\left(a - b\left(\frac{x_4}{1+|x_4|}\right)\right) \\ 0 = -x_3 + 3\tanh(x_1) - 20\tanh(x_2) + 0.1\tanh(x_3) \\ 0 = mx_2 - nh(x_4) \end{cases} \tag{8}$$

After Gaussian elimination, it can be simplified to the following form:

$$\begin{cases} H_1 = -x_2 - 2\tanh(x_1) + 1.2\tanh(x_2) \\ \quad + kx_2\left(a - b\left(\frac{x_4}{1+|x_4|}\right)\right) = 0 \\ H_2 = -x_3 + 3\tanh(x_1) - 20\tanh(x_2) \\ \quad + 0.1\tanh(x_3) = 0 \\ x_1 = 0.5x_3 + 12.9\tanh(x_2) + 0.65\tanh(x_3) \\ x_2 = \frac{nh(x_4)}{m} \end{cases} \tag{9}$$

After eliminating x_1 and x_2, we can use the graphical method to solve the equation. Let the roots obtained be (r_1, r_2, r_3, r_4). These roots are the equilibrium points of the MHNN. Next, we linearize the MHNN at the equilibrium points. The result of the Jacobian matrix for Eq. 7 is shown in Eq. 10.

$$J = \begin{bmatrix} 0.5 - 1.5\tanh(x_1)^2 & 2.9 - 2.9\tanh(x_2)^2 & 0.7 - 0.7\tanh(x_3)^2 & 0 \\ 2\tanh(x_1)^2 - 2 & -1.2\tanh(x_2)^2 + k\left(a - \frac{bx_4}{|x_4|+1.0}\right) + 0.2 & 0 & -kx_2\left(\frac{b}{|x_4|+1} - \frac{bx_4\,\mathrm{sgn}(x_4)}{(|x_4|+1)^2}\right) \\ 3 - 3\tanh(x_1)^2 & 20\tanh(x_2)^2 - 20 & -0.1\tanh(x_3)^2 - 0.9 & 0 \\ 0 & m & 0 & -nh'(x_4) \end{bmatrix} \tag{10}$$

Substituting the equilibrium point (r_1, r_2, r_3, r_4), we can obtain the eigenvalues of the Jacobian matrix. According to Shil'nikov's theorem, if there exists a real eigenvalue δ and two complex conjugate eigenvalues $\alpha + \beta i$ and $|\alpha/\delta| < 1$ and $\delta\alpha < 0$ are satisfied, then the system will exhibit chaos at the equilibrium point. In the following analysis, we will substitute specific values for specific analysis.

4 Kinetic analysis of MHNN

Let the parameters $a = 2.1$, $m = 2$, $n = 1.9$. Taking $h_1(x)$ as the internal state variable function of the memristor and M = 0 as an example, the bifurcation behavior of the MHNN with respect to parameters k and b is studied. The parameter k in Eq. 7 represents the intensity of electromagnetic radiation. By continuously changing it in the range of [-0.3, 0.3], we can obtain the bifurcation diagram and the Lyapunov exponent spectrum diagram with respect to parameter k, as shown in Figure 3. Combining the bifurcation diagram and the Lyapunov exponent spectrum diagram, it can be seen that the MHNN is very sensitive to parameter k. In the interval $(-0.3, -0.095)$, the system exhibits a period one. Near $k = -0.095$, the system bifurcates from period one to period two. Then near $k = -0.065$, the system evolves from period two to period four. After that, with the acceleration of period doubling, the system goes

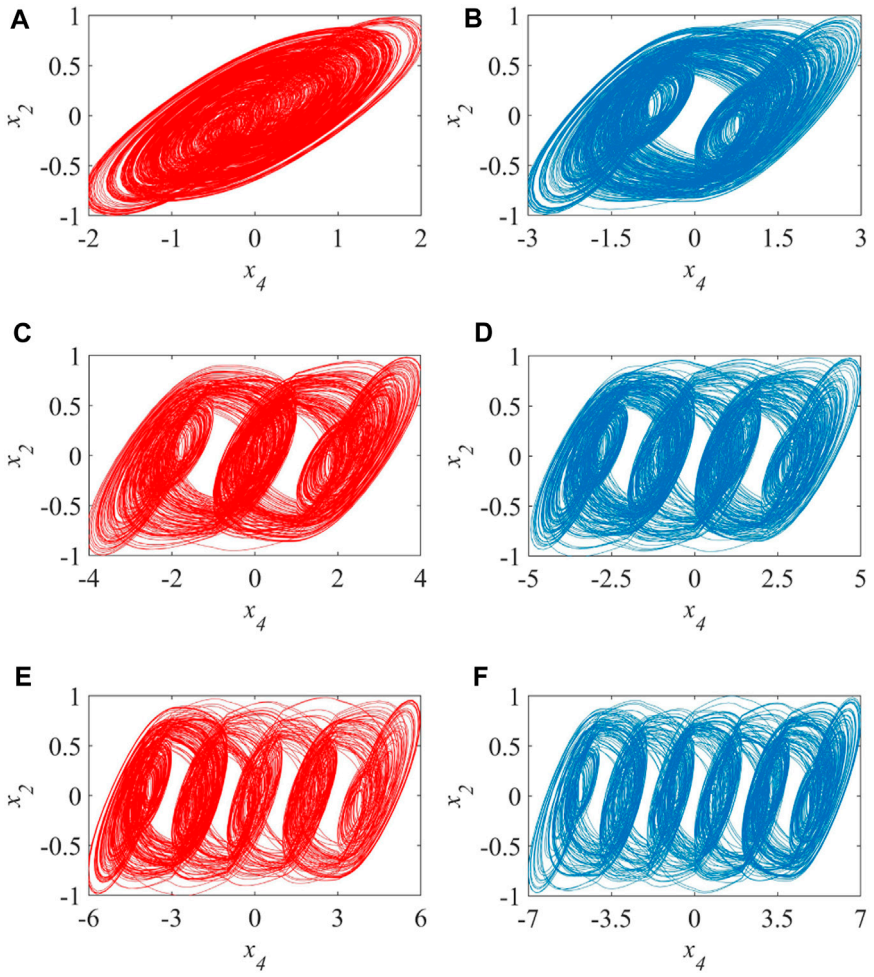


FIGURE 5
Phase diagram of multi-scroll MHNN: $h(x) = h_1(x)$ (red), $h(x) = h_2(x)$ (blue).

to chaos. In the interval [-0.055, 0.235], chaos and periodicity alternate. Finally, after $k = 0.235$, the system degenerates from chaos to period one and no longer exhibits chaotic behavior.

In addition to parameter k, the bifurcation behavior of the MHNN with respect to parameter b is more interesting. Similarly, let parameter b continuously change in the interval [-10, 10], we can obtain the bifurcation diagram and the Lyapunov exponent spectrum diagram with respect to parameter b, as shown in Figure 4. In the interval (−10, −9.7), the system exhibits a period one. Near $b = -9.7$, the system bifurcates from period one to period two. Then near $b = -7.9$, the system evolves from period two to period four. Near $b = -6.6$, the system degenerates from period two to period one. Until near $b = -4.05$, the system again starts to change from period two to period four. After that, with the acceleration of period doubling, the system goes to chaos. In the interval [-3.85, 3.85], chaos and periodicity alternate. Near $b = 4.05$, the system degenerates from period four to period two. Near $b = 6.6$, the system evolves from period two to period four. Then near $b = 7.9$, the system degenerates from period four to period two, and finally degenerates to period one near $b = 9.7$. It is not difficult to find that the system evolves and degenerates at almost symmetrical positions. In addition, the Lyapunov exponent spectrum in Figure 4 is almost symmetrical about the vertical axis. In summary, the MHNN has a symmetrical bifurcation behavior with respect to parameter b.

4.1 Bifurcation behavior

4.1.1 Multi-scroll chaotic attractor

Due to the introduction of the memristor, the MHNN can exhibit multi-scroll behavior that does not exist in the general HNN. Let the parameters $a = 2.1$, $b = 0.1$, $m = 5$, $n = 1.9$ and $k = 0.2$ and the initial values be [0.1, 0.1, 0.1, 0.1]. First, we choose $h_1(x)$ as the internal state variable function of the memristor. By setting the parameter $M = 0$, $M = 1$, $M = 2$ respectively, the MHNN exhibits a single scroll chaotic attractor, a three scroll chaotic attractor and a five scroll chaotic attractor. Then we choose $h_2(x)$ as the internal state variable function of the memristor. Similarly, by setting the parameter $N = 0$, $N = 1$, $N = 2$ respectively, the MHNN exhibits a double scroll chaotic attractor, a four scroll chaotic attractor and a six scroll chaotic attractor. Their phase diagrams are plotted in Figure 5. From the simulation results, it is easy to summarize the control law of the memristor for the number of scrolls. When

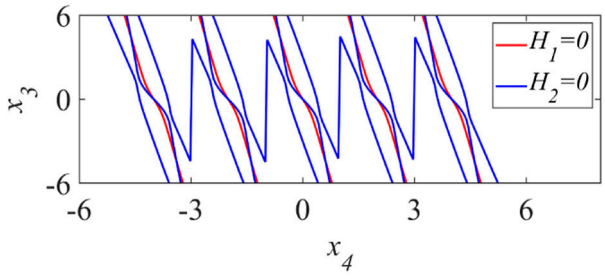


FIGURE 6
Trajectory diagram of the equation in Eq. 9.

choosing $h_1(x)$ as the internal state variable function of the memristor, the MHNN can easily induce $2M + 1$ scrolls. And when choosing $h_2(x)$ as the internal state variable function of the memristor, the MHNN can easily induce $2N + 2$ scrolls.

With the help of Shil'nikov's theorem, the law behind this can be further explained. Taking the appearance of a five-scroll chaotic attractor as an example, according to the idea provided in Section 3.2, all the equilibrium points of the system at this time can be obtained. First, the graphical method is used to obtain the rough values of the roots. Using all the intersections in Figure 6 as a guide, numerical methods are used to obtain more accurate results of x_3 and x_4. Then the obtained results are substituted into Eq. 9 to obtain all the roots. Next, QR decomposition is used to obtain the eigenvalues corresponding to each equilibrium point, and the final results are shown in Table 2. Table 2 shows that for the MHNN that exhibits a five-scroll chaotic attractor, there are a total of 15 saddle-focus points at this time. And according to Shil'nikov's theorem, there will be chaotic attractors near these equilibrium points, which is consistent with the simulation results in Figure 5. By observing the distribution of equilibrium points in Figure 6 and Table 2, it is not difficult to summarize that as the parameter M or N increases, the equilibrium points gradually extend along the direction of x_4. That is to say, by changing the form of the internal variable function of the memristor and its parameters M or N, the number of equilibrium points of the MHNN can be easily controlled. And according to Shil'nikov's theory, there will be chaotic attractors near these saddle-focus type equilibrium points. This directly makes the range of chaotic attractors of the MHNN also increase accordingly, which is specifically manifested as an increase in the number of scrolls in this section.

4.1.2 Initial offset boosting coexistence

In addition to having complex topological structures of multiple scroll attractors, the MHNN also has complex coexistence behavior. Let the parameters of the MHNN be $a = 2.1$, $b = 0.1$, $m = 2$, $n = 1.9$ and $k = 0.2$. First, we choose $h_1(x)$ as the internal state variable function of the memristor, and take $M = 2$ as an example. The initial membrane potentials of neurons 1, 2, and 3 are all set to 0.1, while the initial values of the internal state variable of the memristor are −4, −2, 0, 2 and 4 respectively. By superimposing the phase diagrams corresponding to each initial value together, we obtain the situation of five attractors coexisting as shown in Figure 7A. Observing these five coexisting attractors, it is not difficult to find that their shapes and sizes are highly similar, and they only shift by a fixed distance in the x_4 direction. Figure 7B shows the

TABLE 2 Equilibrium point analysis results of MHNN when $h(x) = h_1(x)$ and $M = 2$.

Equilibrium points				Eigenvalues			Type
x1	x2	x3	x4				
−0.0718	−0.2331	4.4648	−4.6135	−1.9002	−0.9984	0.5322 ± 2.3371i	Saddle focus
0.0000	0.0000	0.0000	−4.0000	2.4764	−1.9000	−1.1202 ± 2.6998i	Saddle focus
0.0717	0.2331	−4.4655	−3.3865	−1.8997	−0.9984	0.5315 + 2.3373i	Saddle focus
−0.0716	−0.2332	4.4662	−2.6136	−1.9004	−0.9984	0.5313 ± 2.337i	Saddle focus
0.0000	0.0000	0.0000	−2.0000	2.4758	−1.9000	−1.1212 ± 2.7003i	Saddle focus
0.0713	0.2332	−4.4682	−1.3862	−1.8991	−0.9984	0.5293 ± 2.3377i	Saddle focus
−0.0708	−0.2333	4.4710	−0.6140	−1.9019	−0.9984	0.5287 ± 2.3364i	Saddle focus
0.0000	0.0000	0.0000	0.0000	−1.9000	2.4728	−1.1264 ± 2.7026i	Saddle focus
0.0691	0.2336	−4.4819	0.6148	−1.8981	−0.9984	0.5193 ± 2.3385i	Saddle focus
0.0684	0.2338	−4.4868	2.6151	−1.8996	−0.9984	0.5167 ± 2.3379i	Saddle focus
0.0000	0.0000	0.0000	2.0000	2.4698	−1.9000	−1.1316 ± 2.7049i	Saddle focus
−0.0687	−0.2337	4.4848	1.3850	−1.9009	−0.9984	0.5187 ± 2.3372i	Saddle focus
0.0000	0.0000	0.0000	4.0000	2.4692	−1.9000	−1.1326 ± 2.7054i	Saddle focus
0.0000	0.0000	0.0000	4.0000	2.4692	−1.9000	−1.1326 ± 2.7054i	Saddle focus
−0.0683	−0.2338	4.4875	3.3848	−1.9003	−0.9984	0.5165 ± 2.3376i	Saddle focus

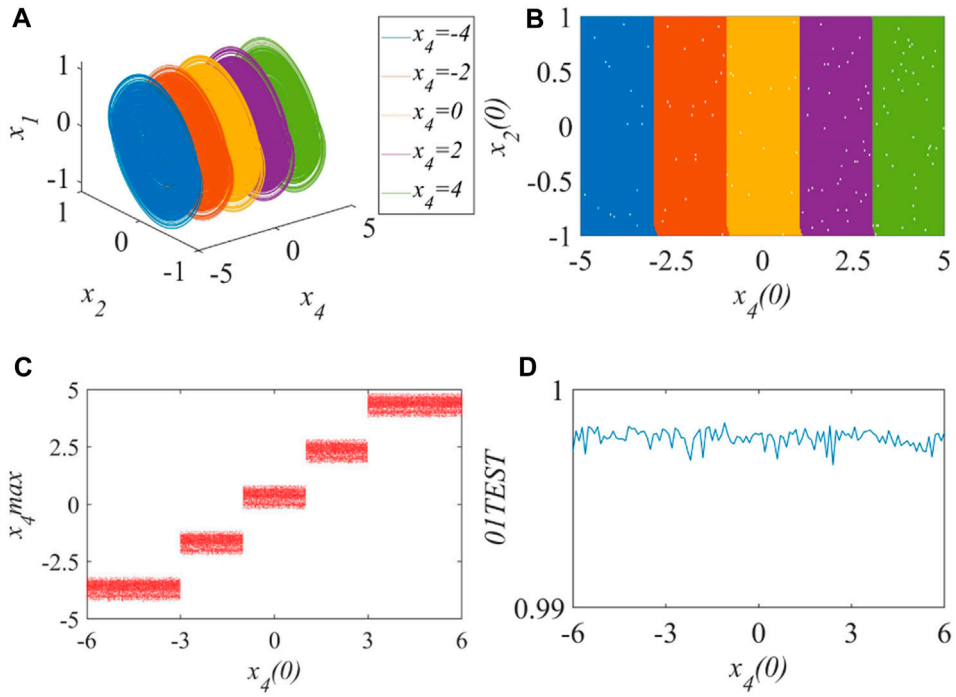


FIGURE 7
Initial offset boosting coexistence when $h_1(x)$ is selected and M = 2: **(A)** Phase diagram, **(B)** Attraction of basin, **(C)** Bifurcation diagram, **(D)** Result of 0–1 test.

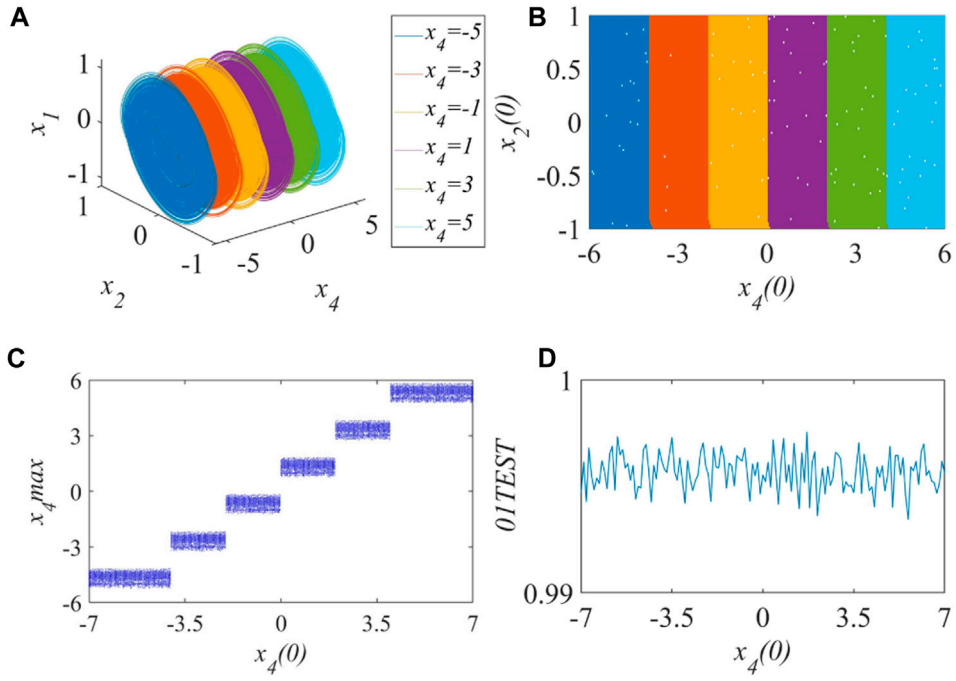


FIGURE 8
Initial offset boosting coexistence when $h_2(x)$ is selected and N = 2: **(A)** Phase diagram, **(B)** Attraction of basin, **(C)** Bifurcation diagram, **(D)** Result of 0–1 test.

basins of attraction corresponding to these attractors. These basins of attraction have the characteristics of clear boundaries and similar shapes, and their positions are very regular, that is, they shift by a fixed distance in the x_4 direction. By plotting the bifurcation diagram and 01 test results with respect to the initial value of the internal variable of the memristor, we obtain Figures 7C, D, which further verify the results in Figures 7A, B.

In addition, $h_2(x)$ is chosen as the internal state variable function of the memristor, and $N = 2$ is taken as an example. The initial membrane potentials of neurons 1, 2, and 3 are all set to 0.1, while the initial values of the internal state variable of the memristor are −5, −3, −1, 1, 3 and 5 respectively. Similarly, by superimposing the phase diagrams corresponding to each initial value together, six attractors coexisting can be obtained as shown in Figure 8A. Observing these six coexisting attractors, it is not difficult to find that their shapes and sizes are highly similar, and they only shift by a fixed distance in the x_4 direction. Figure 8B shows the basins of attraction corresponding to these attractors. These basins of attraction have the characteristics of clear boundaries and similar shapes, and their positions are very regular, that is, they shift by a fixed distance in the x_4 direction. By plotting the bifurcation diagram and 01 test results with respect to the initial value of the internal variable of the memristor, Figures 8C, D are obtained, which further verify the results in Figures 8A, B.

From the analysis results of the previous section, it can be known that by choosing the form of the internal variable function of the memristor and its parameters M or N, the number of equilibrium points of the MHNN can be easily controlled. And there will be chaotic attractors near these saddle-focus type equilibrium points. Different from the previous section, the MHNN does not exhibit multi-scroll phenomena, but manifests as coexistence induced by initial position offset. By analogy with the change law of the number of scrolls of multi-scroll attractors, it can be inferred that when choosing $h_1(x)$ as the internal state variable function of the memristor, there are $2M + 1$ coexisting attractors in the MHNN. And when choosing $h_2(x)$ as the internal state variable function of the memristor, there are $2N + 2$ coexisting attractors in the MHNN.

5 Coupling synchronization of MHNN

5.1 Dual coupling synchronous model

Studies have shown that synchronization plays an important role in memory processing. Synchronization between brain regions supports working memory and long-term memory by facilitating communication between neurons and enhancing neuronal plasticity [44]. Brain regions are synchronized through the connection of some neurons to form inter-regional neural networks, and thus complete the information transmission between different brain regions. Designing a suitable controller is one of the most fundamental methods for controlling complex systems to achieve synchronization, and many scholars have proposed different control strategies [45–55]. In this paper, we use two bounded sub-MHNNs to represent different brain regions. By coupling these two sub-neural networks with a single neuron, a dual-MHNN coupling model is established:

$$\begin{cases} \dot{x}_1 = -x_1 + 1.5\tanh(x_1) + 2.9\tanh(x_2) \\ \qquad + 0.7\tanh(x_3) + p(x_1 - y_1) \\ \dot{x}_2 = -x_2 - 2\tanh(x_1) + 1.2\tanh(x_2) \\ \qquad + kx_2\left(a - b\left(\dfrac{x_4}{1+|x_4|}\right)\right) \\ \dot{x}_3 = -x_3 + 3\tanh(x_1) - 20\tanh(x_2) + 0.1\tanh(x_3) \\ \dot{x}_4 = mx_2 - nh_1(x_4) \\ \dot{y}_1 = -y_1 + 1.5\tanh(y_1) + 2.9\tanh(y_2) \\ \qquad + 0.7\tanh(y_3) - p(x_1 - y_1) \\ \dot{y}_2 = -y_2 - 2\tanh(y_1) + 1.2\tanh(y_2) \\ \qquad + ky_2\left(a - b\left(\dfrac{y_4}{1+|y_4|}\right)\right) \\ \dot{y}_3 = -y_3 + 3\tanh(y_1) - 20\tanh(y_2) + 0.1\tanh(y_3) \\ \dot{y}_4 = my_2 - nh_2(y_4) \end{cases} \tag{11}$$

where p represents the coupling strength of the sub-neural networks. To make the two sub-neural networks successfully synchronized, the difference between their outputs needs to be zero, that is:

$$\lim_{t\to\infty} e_1 = \lim_{t\to\infty} x_i - y_i = 0 \tag{12}$$

where $i = 1, 2, 3$. To prove the validity of the above equation, the following Lyapunov function can be constructed:

$$V = \frac{1}{2}\left(e_1^2 + e_2^2 + e_3^2\right) \tag{13}$$

Then its derivative with respect to time is:

$$\begin{aligned} \dot{V} &= e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 \\ &= (2p-1)e_1^2 - e_2^2 - e_3^2 \\ &\quad + (1.5e_1 - 2e_2 + 3e_3)(\tanh(x_1) - \tanh(y_1)) \\ &\quad + (2.9e_1 + 1.2e_2 - 20e_3)(\tanh(x_2) - 2.9\tanh(y_2)) \\ &\quad + (0.7e_1 + 0.1e_3)(\tanh(x_3) - \tanh(y_3)) \\ &\quad - bke_2\left(\frac{x_2x_4}{1+|x_4|} - \frac{y_2y_4}{1+|y_4|}\right) \\ &\le (2p-1)e_1^2 - e_2^2 - e_3^2 \\ &\quad + 2|1.5e_1 - 2e_2 + 3e_3| \\ &\quad + 2|2.9e_1 + 1.2e_2 - 20e_3| \\ &\quad + 2|0.7e_1 + 0.1e_3| \\ &\quad - bke_2\left(\frac{x_2x_4}{1+|x_4|} - \frac{y_2y_4}{1+|y_4|}\right) \end{aligned} \tag{14}$$

Since $-1 < \frac{x}{1+|x|} < 1$ and both x_2 and y_2 are bounded, there exists a sufficiently large constant c, satisfying:

$$\frac{x_2x_4}{1+|x_4|} - \frac{y_2y_4}{1+|y_4|} < c \tag{15}$$

Therefore, Eq. 14 can be further relaxed as:

$$\begin{aligned} \dot{V} &\le (2p-1)e_1^2 - e_2^2 - e_3^2 \\ &\quad + 2|1.5e_1 - 2e_2 + 3e_3| \\ &\quad + 2|2.9e_1 + 1.2e_2 - 20e_3| \\ &\quad + 2|0.7e_1 + 0.1e_3| \\ &\quad + |cbke_2| \\ &\le (2p-1)e_1^2 - e_2^2 - e_3^2 \\ &\quad + 10.2|e_1| + (6.4 + |cbk|)|e_2| + 46.2|e_3| \\ &\le (2p-1)\left(|e_1| - \frac{5.1}{1-2p}\right)^2 - \frac{5.1^2}{2p-1} \\ &\quad - \left(|e_2| - \frac{6.4+|cbk|}{2}\right)^2 + \left(\frac{6.4+|cbk|}{2}\right)^2 \\ &\quad - (|e_3| - 23.1) + 23.1^2 \end{aligned} \tag{16}$$

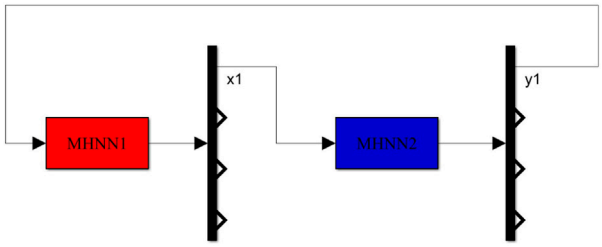


FIGURE 9
Simulink schematic diagram.

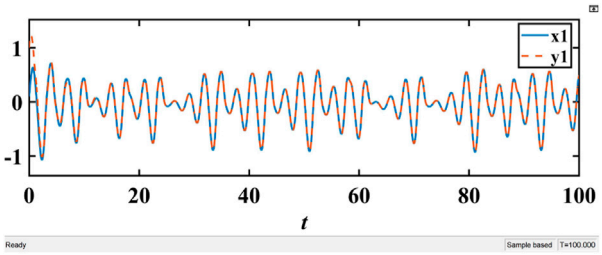


FIGURE 10
Time domain diagram of the membrane potential of two subnetworks of neuron 1.

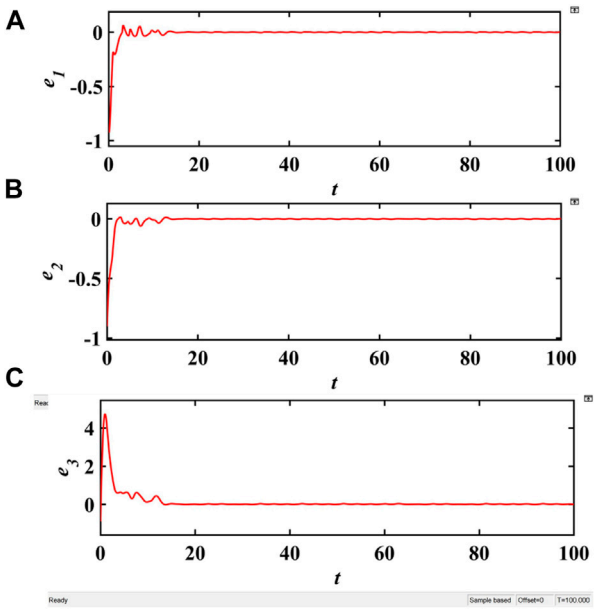


FIGURE 11
Membrane potential difference of subnetwork over time.

Therefore, when $2p - 1 < 0$ holds, Eq. 13 is ultimately uniformly bounded.

5.2 Simulink simulation

The schematic diagram of Simulink experiment is shown in Figure 9 and let the system parameters be $a = 2.1$, $b = 0.1$, $m = 2$, $n = 1.9$, $k = 0.1$. By building a Simlulink model, the coupling synchronization model described by Eq. 11 can be easily simulated. When the coupling strength $p = -1$, the simulation results are shown in Figures 10, 11. Figure 11 shows that after the two sub-memristive HNNs are coupled, the difference between the potentials on the corresponding neurons quickly drops to near zero. Figure 10 is the time domain diagram of the potentials on neuron 1 of the two sub-networks, which shows that they match well.

6 Conclusion

This paper has investigated the nonlinear dynamics and applications of a new non-volatile magnetic-controlled memristor. It is demonstrated that the memristor can simulate the effect of electromagnetic radiation on neuronal membrane flux. By introducing this memristor to a 3D HNN, a 4D MHNN under electromagnetic radiation is constructed. By plotting phase diagrams, bifurcation diagrams, basins of attraction and 01 tests, a series of complex chaotic phenomena are found, including memristor-controlled multi-scroll chaotic attractors, symmetric bifurcation behaviors, initial offset boosting coexistence. Through equilibrium point analysis, the mechanisms behind them are explained. Finally, a dual MHNN coupling synchronization model simulating the inter-brain region synchronization of the human brain is proposed. By constructing a Lyapunov function for the error, this coupling synchronization scheme is proved to be ultimately bounded. By building a Simulink model, we verify the feasibility of this synchronization scheme by simulation experiments. This study contributes to the understanding of memristive systems, synchronization of brain regions and their potential applications in engineering.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

Conceptualization, SF and ZY; methodology, SF; software, SF and XW; validation, SF, XW, and HG; formal analysis, SF and ZY; investigation, XC; resources, ZY; data curation, SF; writing—original draft preparation, SF; writing—review and editing, SF, XW, and ZY; visualization, HG; supervision, ZY; project administration, ZY and SF; funding acquisition, SF and ZY. All authors contributed to the article and approved the submitted version.

Funding

This paper was supported by the Integrated Innovation Project of the Group Company under Grant 2023-JC-13.

Conflict of interest

SF, XW, and XC were employed by the Aircraft Technology Branch of Hunan Aerospace Co., Ltd. SF and HG were employed by the China Aerospace Science and Industry Corporation.

The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* (1952) 117:500–44. doi:10.1113/jphysiol.1952.sp004764

2. Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. *Proc IRE* (1962) 50:2061–70. doi:10.1109/jrproc.1962.288235

3. Deng Z, Wang C, Lin H, Sun Y. A memristive spiking neural network circuit with selective supervised attention algorithm. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2022) 42:2604–17. doi:10.1109/TCAD.2022.3228896

4. Izhikevich EM. Simple model of spiking neurons. *IEEE Trans Neural networks* (2003) 14:1569–72. doi:10.1109/tnn.2003.820440

5. Hindmarsh JL, Rose R. A model of neuronal bursting using three coupled first order differential equations. *Proc R Soc Lond Ser B. Biol Sci* (1984) 221:87–102. doi:10.1098/rspb.1984.0024

6. Chua LO, Yang L. Cellular neural networks: Theory. *IEEE Trans circuits Syst* (1988) 35:1257–72. doi:10.1109/31.7600

7. Chen Z, Liang Q, Wei Z, Chen X, Shi Q, Yu Z, et al. An overview of *in vitro* biological neural networks for robot intelligence. *Cyborg Bionic Syst* (2023) 4:0001. doi:10.34133/cbsystems.0001

8. Hopfield JJ. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci* (1984) 81:3088–92. doi:10.1073/pnas.81.10.3088

9. Lin H, Wang C, Yu F, Hong Q, Xu C, Sun Y. A triple-memristor hopfield neural network with space multi-structure attractors and space initial-offset behaviors. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2023) 1. doi:10.1109/TCAD.2023.3287760

10. Xu Q, Wang Y, Iu HH-C, Wang N, Bao H. Locally active memristor-based neuromorphic circuit: Firing pattern and hardware experiment. *IEEE Trans Circuits Syst Regular Pap* (2023) 70:3130–41. doi:10.1109/TCSI.2023.3276983

11. Pu Y-F, Yi Z, Zhou J-L. Fractional hopfield neural networks: Fractional dynamic associative recurrent neural networks. *IEEE Trans Neural Networks Learn Syst* (2017) 28:2319–33. doi:10.1109/tnnls.2016.2582512

12. Danca M-F, Kuznetsov N. Hidden chaotic sets in a Hopfield neural system. *Solitons and Fractals* (2017) 103:144–50. doi:10.1016/j.chaos.2017.06.002

13. Yu F, Yu Q, Chen H, Kong X, Mokbel AAM, Cai S, et al. Dynamic analysis and audio encryption application in iot of a multi-scroll fractional-order memristive hopfield neural network. *Fractal and Fractional* (2022) 6:370. doi:10.3390/fractalfract6070370

14. Lai Q, Wan Z, Zhang H, Chen G. Design and analysis of multiscroll memristive hopfield neural network with adjustable memductance and application to image encryption. *IEEE Trans Neural Networks Learn Syst* (2022) 2022:1. doi:10.1109/tnnls.2022.3146570

15. Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi-scroll memristive hopfield neural network. *IEEE Trans Netw Sci Eng* (2023) 10:845–58. doi:10.1109/tnse.2022.3223930

16. Wan Q, Li F, Chen S, Yang Q. Symmetric multi-scroll attractors in magnetized Hopfield neural network under pulse controlled memristor and pulse current stimulation. *Solitons and Fractals* (2023) 169:113259. doi:10.1016/j.chaos.2023.113259

17. Yu F, Chen H, Kong X, Yu Q, Cai S, Huang Y, et al. Dynamic analysis and application in medical digital image watermarking of a new multi-scroll neural network with quartic nonlinear memristor. *The Eur Phys J Plus* (2022) 137:434. doi:10.1140/epjp/s13360-022-02652-4

18. Chen C, Min F, Hu F, Cai J, Zhang Y. Analog/digital circuit simplification for Hopfield neural network. *Solitons and Fractals* (2023) 173:113727. doi:10.1016/j.chaos.2023.113727

19. Chen C, Min F, Zhang Y, Bao H. ReLU-type Hopfield neural network with analog hardware implementation. *Solitons and Fractals* (2023) 167:113068. doi:10.1016/j.chaos.2022.113068

20. Chua L. Memristor-the missing circuit element. *IEEE Trans Circuit Theor* (1971) 18:507–19. doi:10.1109/tct.1971.1083337

21. Xu Q, Wang Y, Chen B, Li Z, Wang N. Firing pattern in a memristive Hodgkin–Huxley circuit: Numerical simulation and analog circuit validation. *Solitons and Fractals* (2023) 172:113627. doi:10.1016/j.chaos.2023.113627

22. Liu X, Mou J, Zhang Y, Cao Y. A new hyperchaotic map based on discrete memristor and meminductor: Dynamics analysis, encryption application, and dsp implementation. *IEEE Trans Ind Elect* (2023) 2023:1–10. doi:10.1109/TIE.2023.3281687

23. Yu F, Xu S, Xiao X, Yao W, Huang Y, Cai S, et al. Dynamics analysis, fpga realization and image encryption application of a 5d memristive exponential hyperchaotic system. *Integration* (2023) 90:58–70. doi:10.1016/j.vlsi.2023.01.006

24. Jo SH, Chang T, Ebong I, Bhadviya BB, Mazumder P, Lu W. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett* (2010) 10:1297–301. pMID: 20192230. doi:10.1021/nl904092h

25. Chen C, Chen J, Bao H, Chen M, Bao B. Coexisting multi-stable patterns in memristor synapse-coupled hopfield neural network with two neurons. *Nonlinear Dyn* (2019) 95:3385–99. doi:10.1007/s11071-019-04762-8

26. Eftekhari L, Amirian MM. Stability analysis of fractional order memristor synapse-coupled hopfield neural network with ring structure. *Cogn Neurodynamics* (2023) 17:1045–59. doi:10.1007/s11571-022-09844-9

27. Huang L-L, Zhang Y, Xiang J-H, Liu J. Extreme multistability in a hopfield neural network based on two biological neuronal systems. *IEEE Trans Circuits Syst Express Briefs* (2022) 69:4568–72. doi:10.1109/tcsii.2022.3183340

28. Lin H, Wang C, Yu F, Sun J, Du S, Deng Z, et al. A review of chaotic systems based on memristive hopfield neural networks. *Mathematics* (2023) 11:1369. doi:10.3390/math11061369

29. Leng Y, Yu D, Hu Y, Yu SS, Ye Z. Dynamic behaviors of hyperbolic-type memristor-based Hopfield neural network considering synaptic crosstalk. *Interdiscip J Nonlinear Sci* (2020) 30:033108. doi:10.1063/5.0002076

30. Chen C, Min F, Zhang Y, Bao B. Memristive electromagnetic induction effects on hopfield neural network. *Nonlinear Dyn* (2021) 106:2559–76. doi:10.1007/s11071-021-06910-5

31. Chen M, Chen C-j., Bao B-c., Xu Q. Multi-stable patterns coexisting in memristor synapse-coupled hopfield neural network. In: *Mem-elements for neuromorphic circuits with artificial intelligence applications*. Amsterdam, Netherlands: Elsevier (2021). p. 439–59.

32. Hu Z, Wang C. Hopfield neural network with multi-scroll attractors and application in image encryption. *Multimedia Tools Appl* (2023) 2023. doi:10.1007/s11042-023-15670-w

33. Lin H, Wang C, Tan Y. Hidden extreme multistability with hyperchaos and transient chaos in a hopfield neural network affected by electromagnetic radiation. *Nonlinear Dyn* (2020) 99:2369–86. doi:10.1007/s11071-019-05408-5

34. Wan Q, Yan Z, Li F, Liu J, Chen S. Multistable dynamics in a hopfield neural network under electromagnetic radiation and dual bias currents. *Nonlinear Dyn* (2022) 109:2085–101. doi:10.1007/s11071-022-07544-x

35. Zhang S, Zheng J, Wang X, Zeng Z, He S. Initial offset boosting coexisting attractors in memristive multi-double-scroll hopfield neural network. *Nonlinear Dyn* (2020) 102:2821–41. doi:10.1007/s11071-020-06072-w

36. Yu F, Kong X, Mokbel AAM, Yao W, Cai S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive hopfield

neural network with a novel local active memeristor. *IEEE Trans Circuits Syst Express Briefs* (2023) 70:326–30. doi:10.1109/tcsii.2022.3218468

37. Lai Q, Wan Z, Kuate PDK. Generating grid multi-scroll attractors in memristive neural networks. *IEEE Trans Circuits Syst Regular Pap* (2023) 70:1324–36. doi:10.1109/tcsi.2022.3228566

38. Wang C, Liu X, Xia H. Multi-piecewise quadratic nonlinearity memristor and its 2 n-scroll and 2 n+ 1-scroll chaotic attractors system. *Chaos* (2017) 27:033114. doi:10.1063/1.4979039

39. Xia X, Zeng Y, Li Z. Coexisting multiscroll hyperchaotic attractors generated from a novel memristive jerk system. *Pramana* (2018) 91:82–14. doi:10.1007/s12043-018-1657-3

40. Zhang S, Zheng J, Wang X, Zeng Z. Multi-scroll hidden attractor in memristive hr neuron model under electromagnetic radiation and its applications. *Chaos* (2021) 31:011101. doi:10.1063/5.0035595

41. Lin H, Wang C, Sun Y, Wang T. Generating n-scroll chaotic attractors from a memristor-based magnetized hopfield neural network. *IEEE Trans Circuits Syst Express Briefs* (2023) 70:311–5. doi:10.1109/tcsii.2022.3212394

42. Adhikari SP, Sah MP, Kim H, Chua LO. Three fingerprints of memristor. *IEEE Trans Circuits Syst Regular Pap* (2013) 60:3008–21. doi:10.1109/tcsi.2013.2256171

43. Lin H, Wang C, Deng Q, Xu C, Deng Z, Zhou C. Review on chaotic dynamics of memristive neuron and neural network. *Nonlinear Dyn* (2021) 106:959–73. doi:10.1007/s11071-021-06853-x

44. Fell J, Axmacher N. The role of phase synchronization in memory processes. *Nat Rev Neurosci* (2011) 12:105–18. doi:10.1038/nrn2979

45. Tan F, Zhou L, Lu J, Quan H, Liu K. Adaptive quantitative control for finite time synchronization among multiplex switched nonlinear coupling complex networks. *Eur J Control* (2023) 70:100764. doi:10.1016/j.ejcon.2022.100764

46. Zhang C, Yan L, Gao Y, Wang W, Li K, Wang D, et al. A new adaptive iterative learning control of finite-time hybrid function projective synchronization for unknown time-varying chaotic systems. *Front Phys* (2023) 11:1127884. doi:10.3389/fphy.2023.1127884

47. Yao W, Wang C, Sun Y, Gong S, Lin H. Event-triggered control for robust exponential synchronization of inertial memristive neural networks under parameter disturbance. *Neural Networks* (2023) 164:67–80. doi:10.1016/j.neunet.2023.04.024

48. Zhou L, Huang M, Tan F, Zhang Y. Mean-square bounded synchronization of complex networks under deception attacks via pinning impulsive control. *Nonlinear Dyn* (2023) 111:11243–59. doi:10.1007/s11071-023-08448-0

49. Lin L, Zhuang Y, Xu Z, Yang D, Wu D. Encryption algorithm based on fractional order chaotic system combined with adaptive predefined time synchronization. *Front Phys* (2023) 11:1202871. doi:10.3389/fphy.2023.1202871

50. Ma M-L, Xie X-H, Yang Y, Li Z-J, Sun Y-C. Synchronization coexistence in a rulkov neural network based on locally active discrete memristor. *Chin Phys B* (2023) 32:058701. doi:10.1088/1674-1056/acb9f7

51. Xu Q, Liu T, Ding S, Bao H, Li Z, Chen B. Extreme multistability and phase synchronization in a heterogeneous bi-neuron rulkov network with memristive electromagnetic induction. *Cogn Neurodynamics* (2023) 17:755–66. doi:10.1007/s11571-022-09866-3

52. Zhang C, Zhang C, Zhang X, Wang F, Liang Y. Dynamic event-triggered control for intra/inter-layer synchronization in multi-layer networks. *Commun Nonlinear Sci Numer Simulation* (2023) 119:107124. doi:10.1016/j.cnsns.2023.107124

53. Yu Y, Xiang L, Liu B, Xia C. Moment-based analysis of pinning synchronization in complex networks with sign inner-coupling configurations. *Front Phys* (2023) 11:1179469. doi:10.3389/fphy.2023.1179469

54. Zhou L, Lin H, Tan F. Fixed/predefined-time synchronization of coupled memristor-based neural networks with stochastic disturbance. *Solitons and Fractals* (2023) 173:113643. doi:10.1016/j.chaos.2023.113643

55. Yao W, Wang C, Sun Y, Zhou C. Robust multimode function synchronization of memristive neural networks with parameter perturbations and time-varying delays. *IEEE Trans Syst Man, Cybernetics: Syst* (2020) 52:260–74. doi:10.1109/tsmc.2020.2997930

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 08 September 2023
DOI 10.3389/fphy.2023.1273872

OPEN ACCESS

EDITED BY
Jesus Manuel Munoz-Pacheco,
Benemérita Universidad Autónoma de Puebla, Mexico

REVIEWED BY
Kehui Sun,
Central South University, China
Aceng Sambas,
Muhammadiyah University of Tasikmalaya, Indonesia

*CORRESPONDENCE
Chunhua Wang,
wch1227164@hnu.edu.cn

RECEIVED 07 August 2023
ACCEPTED 23 August 2023
PUBLISHED 08 September 2023

CITATION
Hu Z, Lin H and Wang C (2023), A novel grid multi-structure chaotic attractor and its application in medical image encryption.
Front. Phys. 11:1273872.
doi: 10.3389/fphy.2023.1273872

A novel grid multi-structure chaotic attractor and its application in medical image encryption

Zhenhua Hu[1,2], Hairong Lin[1] and Chunhua Wang[1]*

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, [2]College of Science, Hunan City University, Yiyang, China

Grid multi-scroll/wing chaotic systems are complex non-linear dynamic systems, which are widely used in secure communication. The grid multi-scroll/wing chaotic systems are usually realized by using the function control method, which has a complex realization method, many control parameters, and a simple unit attractor structure. In this paper, based on the Hopfield neural network, a memristive Hopfield neural network model is proposed by using the memristor synapse control method. The model can generate novel grid multi-structure chaotic attractors, which have the characteristics of a simple implementation method, few control parameters, and complex unit attractor structure. Firstly, the generation mechanism of the grid multi-structure chaotic attractors is analyzed by the equilibrium points and stability. Secondly, its basic dynamical characteristics including the Lyapunov exponent spectrum, fractal dimension, time series, power spectrum, bifurcation diagram, and Poincaré section are analyzed. Thirdly, an analog circuit of the neural network model is designed and realized by Multisim. Finally, combined with the chaos encryption principle, an image encryption scheme is designed based on the generated grid multi-structure attractors. Experimental results show that compared with the existing schemes, the proposed scheme has larger information entropy, higher key sensitivity, and a good application prospect.

KEYWORDS

multi-structure attractor, memristor, Hopfield neural network, image encryption, circuit implementation

1 Introduction

Chaos theory is an important discovery of human natural science in the 20th century and has been widely concerned in the fields of non-linear circuits, neural networks, information security, and so on [1–3]. Chaotic attractors are the core of chaos theory, which corresponds to the motion of a disordered steady state in a chaotic system. Since Lorenz [4] discovered the first double-wing attractor in 1962, the construction of complex attractors in chaotic systems has been an important topic in the field of chaos research. After years of exploration, many chaotic systems with different types of attractors have been discovered [5–8], but a complete theoretical model has not yet been developed. The study of chaotic attractors is a very important and key link in the study of chaos theory. From the whole research process, the study of chaotic attractors has just started, and more chaotic attractors are still to be explored and discovered. Furthermore, the wide application of chaotic signals generated by chaotic

attractors in the field of information security is another reason that attracts many researchers.

At present, there are three types of chaotic attractors found at home and abroad: 1) Self-excited attractors and hidden attractors. The chaotic attractors discovered earlier are closely related to the unstable equilibrium point of the system and are called self-excited attractors, such as the classical Lorenz attractors [4], Chua's attractors [9], Chen attractors [10], Lv attractors [11] and so on [12–16]. With the deepening of research on self-excited attractors, some scholars have found that there is an attractor basin of chaos that does not intersect with any open small neighborhood of the equilibrium point of the system, which is called a hidden attractor [17]. In recent years, various hidden attractors have been proposed, such as the hidden attractor of no equilibrium point [18], the hidden attractor of a stable equilibrium point [19], and the hidden attractor of infinite equilibrium points [20]. 2) transient chaotic attractors, chaotic attractors, and hyperchaotic attractors. Chaotic systems can be characterized by the Lyapunov exponent, and chaotic attractors can be divided into different types according to the maximum Lyapunov exponent spectrum. In general, a transient chaotic attractor is defined as the existence of a positive Lyapunov exponent in finite time [21]. The attractor has a positive Lyapunov exponent in infinite time and is called a chaotic attractor. Hyperchaotic attractors are defined as the simultaneous existence of two or more positive Lyapunov exponents in infinite time [22]. In the past few decades, the research on transient chaotic attractors, chaotic attractors, and hyperchaotic attractors has been fruitful [23–25]. 3) Double scroll/wing attractor, multi-scroll/wing attractor, and grid multi-scroll/wing attractor. Double scroll/wing attractors are classical chaotic attractors with special scroll and dynamic trajectories of butterfly wings. Many simple chaotic systems can produce double scroll/wing attractors, such as Chua's system, Lorenz system, Sprott system [26], Jerk circuit [27], and so on [28, 29]. With the deepening of research on double scroll/wing attractors, various functions such as periodic function [30], piecewise linear function [31], and multistage logic pulse function [32] have been used to expand double scroll/wing attractors in multiple directions to generate multi-scroll/wing attractors [33–35] and grid multi-scroll/wing attractors [36–38]. Among them, the grid multi-scroll/wing attractor has flexible adjustability and high complexity and has a very broad application prospect in secure communication and chaos control [39, 40], so it has become a research hotspot in academia.

The artificial neural network is a kind of non-linear system with complex dynamic characteristics [41, 42]. In recent years, the study of neural network dynamics has gradually become a new research hotspot in the intersection of physics, mathematics, computer science, and neuroscience [43–45]. Among them, Hopfield neural network is a complex non-linear system with chaotic behaviors [46]. Due to its chaotic characteristic, various Hopfield neural network models with different neurons and synaptic weights have been proposed and analyzed [47–50]. In particular, memristive Hopfield neural networks have attracted much attention because of their complex chaotic behavior and potential applications in image encryption. For example, Lin et al. [51] constructed a memristive Hopfield neural network with an arbitrary number of scroll attractors. Yu et al. [52, 53] designed two different memristive Hopfield neural networks which can generate multi-double-scroll attractors. Furthermore, Lai et al. [54] proposed a memristive Hopfield neural network with grid multi-scroll attractors. Meanwhile, the memristive neural network is used to design an image encryption scheme. Although the grid multi-scroll/wing attractor has excellent characteristics, the existing grid multi-scroll/wing attractor implementation methods are complex, the unit structure is simple, and the control parameters are many, and the exploration in recent years seems to indicate that the research of grid multi-scroll/wing attractor is difficult to make updated progress.

To solve these problems, this paper proposes a new method for the generation of novel grid multi-structure chaotic attractors controlled by memristor synapses based on the Hopfield neural network. This method is based on the neural network with complex dynamic characteristics, which successfully gets rid of the scroll/wing attractor unit structure, and makes it appear a more complex and changeable chaotic attractor unit structure. In addition, only two memristor synaptic control parameters are needed to adjust the grid size of the chaotic attractor. The basic dynamic characteristics of the new grid multi-structure attractor are studied by theoretical analysis and numerical simulation. At the same time, the existence of the attractor is verified from the aspect of the circuit. Finally, a scheme of grid multi-structure attractor image encryption is designed based on chaotic encryption theory. The experimental results show that the chaotic attractor exhibits complex dynamic characteristics, so it has better encryption performance and effect in image encryption.

The rest of this paper is organized as follows. Section 2 constructs a memristive Hopfield neural network. Section 3 analyzes the dynamical behaviors of the neural network. Section 4 designs and implements an analog neural network circuit. Section 5 designs a medical image encryption scheme. Section 6 concludes the article.

2 Construction of grid multi-structure attractors

2.1 Introduction of memristor and Hopfield neural network

A memristor is a non-linear circuit element with synaptic memory properties [55], often described in terms of voltage and current relationships. In this paper, a multi-piecewise non-linear flux-controlled memristor model is introduced [51], which can be expressed as

$$\begin{cases} i = (a + b\varphi)v \\ \dot{\varphi} = cv - dh(\varphi) \end{cases} \tag{1}$$

where a, b, c, and d represent the four positive parameters of the memristor, and $W(\varphi) = (a + b\varphi)$ represents the memristor conductance associated with the memristor state variable φ. In addition, $h(\varphi)$ is the internal state variable function of the memristor, which can be expressed as

$$h(\varphi) = \begin{cases} h_1(\varphi) \\ h_2(\varphi) \end{cases} \tag{2}$$

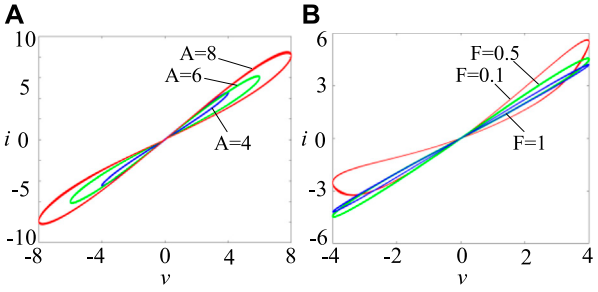


FIGURE 1
Pinched hysteresis loop characteristics of the memristor. **(A)** F = 0.1. **(B)** A = 4.

where

$$h_1(\varphi) = \begin{cases} \varphi, N = 0 \\ \varphi - \sum_{i=1}^{N} (\mathrm{sgn}(\varphi + (2i-1)) + \mathrm{sgn}(\varphi - (2i-1))) \\ \mathrm{N} = 1, 2, 3, \ldots \end{cases} \quad (3)$$

$$h_2(\varphi) = \begin{cases} \varphi - \mathrm{sgn}(\varphi), \mathrm{M} = 0 \\ \varphi - \mathrm{sgn}(\varphi) - \sum_{j=1}^{M} (\mathrm{sgn}(\varphi + 2j) + \mathrm{sgn}(\varphi - 2j)) \\ M = 1, 2, 3, \ldots \end{cases} \quad (4)$$

the sgn(.) is a symbolic function, N and where M are the two control parameters of the memristor. To verify that the above mathematical model is a memristor model, consider the case when $N = 3$. With fixed memristor parameters $a = 1$, $b = 0.05$, $c = 2.2$, $d = 1.2$, $\varphi_0 = 0$, the pinched hysteresis loop of the memristor is simulated by MATLAB R2017a when sinusoidal voltage $v = A\sin(2\pi Ft)$ is applied at both ends of the memristor. The simulation results are shown in Figure 1. Under different excitation amplitude A, the pinched hysteresis loop of the model always passes through the origin of the voltage and current plane. With the increase of voltage frequency F, the area of the pinched hysteresis loop of the memristor decreases gradually. When the voltage frequency increases to infinity, the pinched hysteresis loop shrinks to a one-valued function. In summary, the proposed mathematical model satisfies the three characteristics of the memristor and is a memristor model.

The Hopfield neural network can produce complex chaotic phenomena and is a classical model for studying chaotic dynamics. A Hopfield neural network with n neurons can be expressed as [46]

$$C_i\dot{v}_i = -\frac{v_i}{R_i} + \sum_{j=1}^{n} w_{ij}\tanh(v_j) + I_i (i, j \in N^*) \quad (5)$$

where, C_i, v_i, and R_i respectively represent the membrane capacitance, membrane voltage and membrane resistance of neuron i. w_{ij} represents the synaptic connection weight from neuron j to neuron i, and tanh(.) is the neuronal activation function. In addition, I_i represents the external bias current of the neuron. Generally speaking, the dynamic behavior of the Hopfield neural network is closely related to its synaptic weight coefficient. According to neural network model (Eq. 5), a Hopfield neural network model with four neurons is constructed by selecting

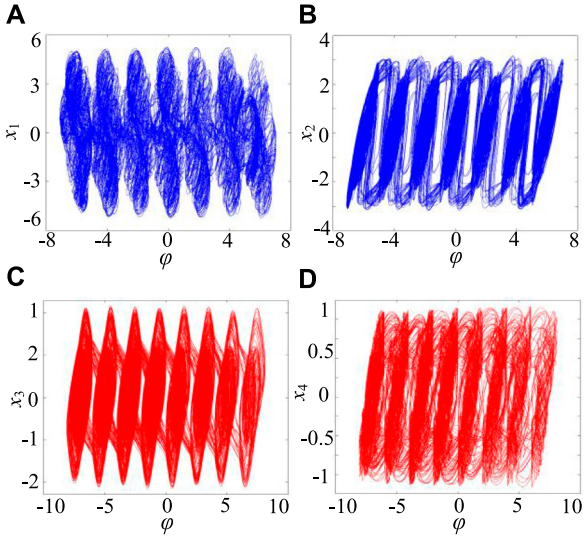


FIGURE 2
Multi-structure chaotic attractor of system (7). **(A)** 7-structure attractor on the φ-x_1 plane. **(B)** 7-structure attractor on the φ-x_2 plane. **(C)** 8-structure attractor on the φ-x_3 plane. **(D)** 8-structure attractor on the φ-x_4 plane.

appropriate synaptic weight coefficients, let $C_i = 1$, $R_i = 1$, $I_i = 0$, and its expression is as follows

$$\begin{cases} \dot{x}_1 = -x_1 + 3\tanh(x_2) + 3\tanh(x_3) - 13\tanh(x_4) \\ \dot{x}_2 = -x_2 - \tanh(x_1) + 1.5\tanh(x_2) + 7\tanh(x_3) - 5\tanh(x_4) \\ \dot{x}_3 = -x_3 - 4\tanh(x_2) + 1.8\tanh(x_3) + 4\tanh(x_4) \\ \dot{x}_4 = -x_4 + 0.7\tanh(x_1) + 2\tanh(x_4) \end{cases} \quad (6)$$

where x_i (i = 1, 2, 3, 4) is the membrane voltage of neuron i.

2.2 Construction of single-directional multi-structure attractors

Memristors have synaptic memory and programmability and can be used to simulate neural synapses [55]. When replacing the resistive synapses between neuron 3 and neuron 1 in the neural network (6) with the multi-piecewise non-linear memristor (1) described above, a memristive Hopfield neural network can be constructed as follows

$$\begin{cases} \dot{x}_1 = -x_1 + 3\tanh(x_2) + \rho W(\varphi)\tanh(x_3) - 13\tanh(x_4) \\ \dot{x}_2 = -x_2 - \tanh(x_1) + 1.5\tanh(x_2) + 7\tanh(x_3) - 5\tanh(x_4) \\ \dot{x}_3 = -x_3 - 4\tanh(x_2) + 1.8\tanh(x_3) + 4\tanh(x_4) \\ \dot{x}_4 = -x_4 + 0.7\tanh(x_1) + 2\tanh(x_4) \\ \dot{\varphi} = c\tanh(x_3) - dh(\varphi) \end{cases} \quad (7)$$

where, ρ represents the memristor synaptic coupling strength, $\rho W(\varphi) = \rho(a + b\varphi)$ represents the memristor synaptic coupling weight.

When $a = 1$, $b = 0.01$, $c = 2.2$, $d = 1.2$, $\rho = 3$, the initial value (0.1, 0.1, 0.1, 0.1, 0.1), the system in theρ-axis can produce n structure attractor. Taking $N = 3$ and $M = 3$ respectively, the numerical

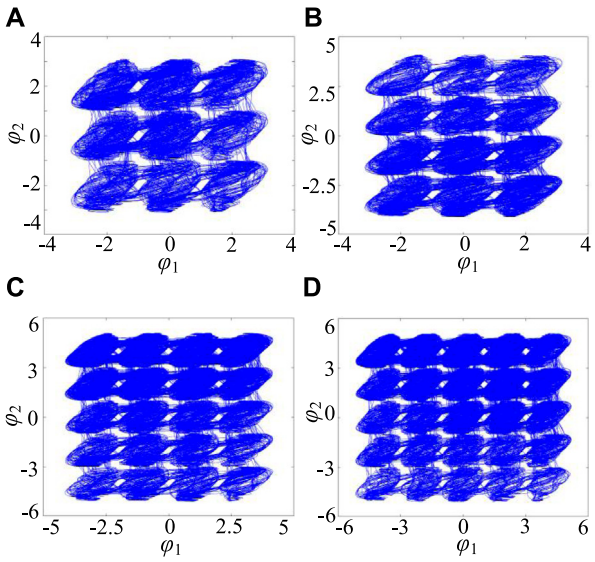


FIGURE 3
Grid multi-structure chaotic attractor of system (8). **(A)** 3 × 3-structure attractor (N_1 = 1, N_2 = 1). **(B)** 3 × 4-structure attractor (N_1 = 1, M_2 = 1). **(C)** 4 × 5-structure attractor (M_1 = 1, N_2 = 2). **(D)** 5 × 5-structure attractor (N_1 = 2, N_2 = 2).

simulation results of 7-structure chaotic attractor and 8-structure chaotic attractor are obtained according to the system (7), as shown in Figure 2. From the attractor substructure in Figure 2, it is not difficult to find that, unlike the multi-scroll/wing chaotic attractor, the multi-structure chaotic attractor contains the unit structure attractor with multiple trajectory-disorder, rather than the conventional scroll/wing.

2.3 Construction of grid multi-structure attractors

Further studies show that the chaotic attractor can be reconstructed and expanded in different directions by using other synapses in the same type of multi-segment non-linear memristor synaptic replacement system (7), and the grid multi-structure chaotic attractor can be constructed. The system model is as follows

$$\begin{cases} \dot{x}_1 = -x_1 + 3\tanh(x_2) + \rho_1 W(\varphi_1)\tanh(x_3) - 13\tanh(x_4) \\ \dot{x}_2 = -x_2 - \tanh(x_1) + 1.5\tanh(x_2) + 7\tanh(x_3) - \rho_2 W(\varphi_2)\tanh(x_4) \\ \dot{x}_3 = -x_3 - 4\tanh(x_2) + 1.8\tanh(x_3) + 4\tanh(x_4) \\ \dot{x}_4 = -x_4 + 0.7\tanh(x_1) + 2\tanh(x_4) \\ \dot{\varphi}_1 = c\tanh(x_3) - dh(\varphi_1) \\ \dot{\varphi}_2 = c\tanh(x_4) - dh(\varphi_2) \end{cases} \tag{8}$$

where ρ_i represents the memristor synaptic coupling strength, $\rho_i W(\varphi_i) = \rho_i(a + b\varphi_i)$ represents the memristor synaptic coupling weight.

Select $a = 1$, $b = 0.01$, $c = 2.2$, $d = 1.2$, $\rho_1 = 3$, $\rho_2 = 5$, and the initial value is (0.1, 0.1, 0.1, 0.1, 0.1). By taking different memristor control parameters N_i and M_i, the $n \times m$ grid multi-structure chaotic attractor can be obtained from system (8), as shown in Figure 3.

TABLE 1 Relation between memristor synaptic control parameters and grid attractors.

N_1/M_1	φ_1 (n)	N_2/M_2	φ_2 (m)
0	1/2	0	1/2
1	3/4	1	3/4
2	5/6	2	5/6
⋯	⋯	⋯	⋯
N/M	$(2N + 1)/(2M + 2)$	N/M	$(2N + 1)/(2M + 2)$

Obviously, by controlling the two control parameters of the two memristor synapses, a grid multi-structure chaotic attractor of arbitrary size can be obtained, as shown in Table 1. Therefore, compared with the existing multi-parameter function control methods of grid multi-scroll/wing chaotic attractors [30–32], the proposed control method of grid multi-structure chaotic attractors is greatly simplified. In addition, by comparing the structure of system (7) and system (8), we can see that: Only one memristor synapse is introduced in system (7), so that the attractor expands horizontally in the direction of introducing the magnetic flux variable φ, while system (8) introduces two memristor synapses at the same time, so that the attractor expands simultaneously in both directions of the magnetic flux variable φ1 and the magnetic flux variable φ2, thus forming a more complex grid multi-structure chaotic attractor. This is a phenomenon that has not been seen in many chaotic structures before.

3 Basic dynamics analysis

In this section, the dynamic generation mechanism and chaotic characteristics of system (8) are analyzed by dynamic analysis method and numerical simulation.

3.1 Equilibrium points and their stabilities

The equilibrium point is a necessary condition for the generation of self-excited chaos, which can reveal the formation mechanism of chaos attractor from the system level. If the right-hand side of Eq. 8 is equal to 0, the equilibrium state equation of the system can be obtained as follows

$$\begin{cases} -x_1 + 3\tanh(x_2) + \rho_1 W(\varphi_1)\tanh(x_3) - 13\tanh(x_4) = 0 \\ -x_2 - \tanh(x_1) + 1.5\tanh(x_2) + 7\tanh(x_3) - \rho_2 W(\varphi_2)\tanh(x_4) = 0 \\ -x_3 - 4\tanh(x_2) + 1.8\tanh(x_3) + 4\tanh(x_4) = 0 \\ -x_4 + 0.7\tanh(x_1) + 2\tanh(x_4) = 0 \\ c\tanh(x_3) - dh(\varphi_1) = 0 \\ c\tanh(x_4) - dh(\varphi_2) = 0 \end{cases} \tag{9}$$

Equation 9 is a sixth-order equation, using the MATLAB platform to solve the equilibrium point by graphical analysis method. To have resistance control parameters $N_1 = N_2 = 1$ as an example, set $a = 1$, $b = 0.01$, $c = 2.2$, $d = 1.2$, $\rho_1 = 3$, $\rho_2 = 5$, Eq. 9 can be converted to

10.3389/fphy.2023.1273872

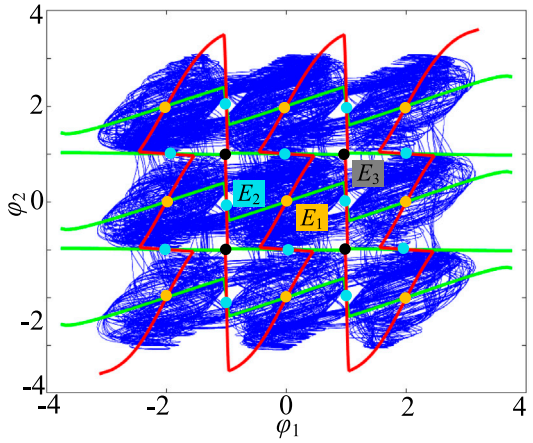


FIGURE 4
Function f_1 and f_2 curves, equilibrium point, and superposition diagram of 3 × 3 structure attractor.

$$\begin{cases} x_3 = \text{atanh}\left(dh\left(\varphi_1\right)/c\right) \\ x_4 = \text{atanh}\left(dh\left(\varphi_2\right)/c\right) \\ x_1 = \text{atanh}\left(\left(x_4 - 2\text{tanh}\left(x_4\right)\right)/0.7\right) \\ x_2 = \text{atanh}\left(\left(-x_3 + 1.8\text{tanh}\left(x_3\right) + 4\text{tanh}\left(x_4\right)\right)/4\right) \\ f_1\left(\varphi_1, \varphi_2\right) = -x_1 + 3\text{tanh}\left(x_2\right) + \rho_1 W\left(\varphi_1\right)\text{tanh}\left(x_3\right) - 13\text{tanh}\left(x_4\right) = 0 \\ f_2\left(\varphi_1, \varphi_2\right) = -x_2 - \tanh\left(x_1\right) + 1.5\text{tanh}\left(x_2\right) + 7\text{tanh}\left(x_3\right) - \rho_2 W\left(\varphi_2\right)\text{tanh}\left(x_4\right) = 0 \end{cases} \tag{10}$$

The functions f_1 and f_2 are drawn in MATLAB R2017a, as shown in Figure 4, where f_1 is represented by a green curve and f_2 by a red curve. It can be observed from the figure that f_1 and f_2 have 25 intersection points in different positions, that is, the system (8) has 25 equilibrium points at this time. Near the equilibrium point, the Jacobian matrix of the system can be expressed as

$$J = \begin{bmatrix} -1 & 3m_2 & \rho_1 W\left(\varphi_1\right)m_3 & -13m_4 & \rho_1 b\text{tanh}\left(x_3\right) & 0 \\ -m_1 & -1 + 1.5m_2 & 7m_3 & -\rho_2 W\left(\varphi_2\right)m_4 & 0 & -\rho_2 b\text{tanh}\left(x_4\right) \\ 0 & -4m_2 & -1 + 1.8m_3 & 4m_4 & 0 & 0 \\ 0.7m_1 & 0 & 0 & -1 + 2m_4 & 0 & 0 \\ 0 & 0 & cm_3 & 0 & -dh'\left(\varphi_1\right) & 0 \\ 0 & 0 & 0 & cm_4 & 0 & -dh'\left(\varphi_2\right) \end{bmatrix} \tag{11}$$

where $m_i = \text{sech}^2(x_i)$, $i = 1, 2, 3, 4$. Through numerical calculation, these 25 equilibrium points have three different stability, namely, unstable saddle coke of indicator 4 (yellow), unstable saddle coke of indicator 5 (blue), and unstable saddle coke of indicator 6 (black), as shown in Table 2. Obviously, system (8) generates 9 single structure attractors in the neighborhood of 9 unstable saddle focal points with index-4, 10 unstable saddle focal points with index-5 produce bond bands in the direction of φ_1 and φ_2, and 4 unstable saddle focal points with index-6 play a role in strengthening the connection in the diagonal direction. Finally, under the action of 25 unstable equilibrium points, a 3 × 3 grid multi-structure chaotic attractor phenomenon is generated, as shown in Figure 4. A large number of simulation results show that with the increase of the control parameters N_1/M_1 and N_2/M_2 of the two memristic synapses, the equilibrium point of the system will gradually expand along the φ_1 and φ_2 axes, resulting in a larger number of grid multi-structure chaotic attractors on the φ_1-φ_2 phase plane.

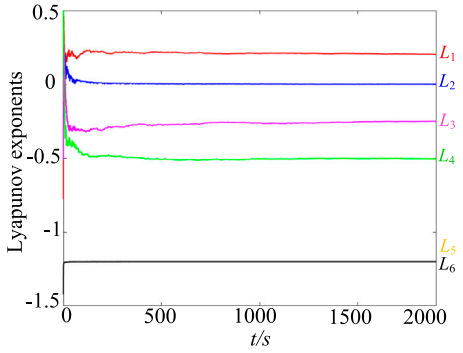


FIGURE 5
Lyapunov exponents of system (8).

3.2 Lyapunov exponents and Kaplan-Yorke dimension

Lyapunov exponent spectrum analysis is an important method to study the dynamical characteristics of chaotic systems. It reveals the chaotic characteristics of the system from the perspective of the average exponential rate of convergence or divergence between adjacent orbits in the phase space of the system. Under the condition of multi-structure attractor generation in a 3 × 3 grid, six Lyapunov indices of the system were obtained by using the Wolf's Jacobian-based method, as shown in Figure 5. Where L_1 = 0.2146, L_2 = 0.0015, L_3 = −0.2608, L_4 = −0.5047, L_5 = −1.201, L_6 = −1.202. Obviously, it is not difficult to find that system (8) has a positive Lyapunov exponent; a Lyapunov exponent is approaching 0; The sum of all Lyapunov exponents is negative. Therefore, under these conditions, the system is chaotic. The Kaplan-Yorke dimension corresponding to its Lyapunov index is

$$D_{KY} = j + \frac{1}{\left|L_{j+1}\right|}\sum_{i=1}^{j} L_i = 2 + \frac{L_1 + L_2}{\left|L_3\right|} = 2.8286 \tag{12}$$

Obviously, the Kaplan-Yorke dimension of the system is fractional, which further proves that the grid multi-structure attractors generated by the system are chaotic.

TABLE 2 Representative equilibrium points of system (8), corresponding to eigenvalues and types of equilibrium points.

Equilibrium points	Eigenvalues	Stabilities
E_1 (0,0,0,0,0,0)	(−1.2,−1.2, 0.3181 ± 5.9165i, 0.3319 ± 1.9652i)	unstable saddle focus with index 4
E_2 (0,0,0,0,-1,0)	(1.2e + 6,−1.2, 0.3213 ± 5.9156i, 0.3286 ± 1.9678i)	unstable saddle focus with index 5
E_3 (0,0,0,0,1,1)	(1.2e + 6, 1.2e + 6, 0.3165 ± 5.9186i, 0.3335 ± 1.9586i)	unstable saddle focus with index 6

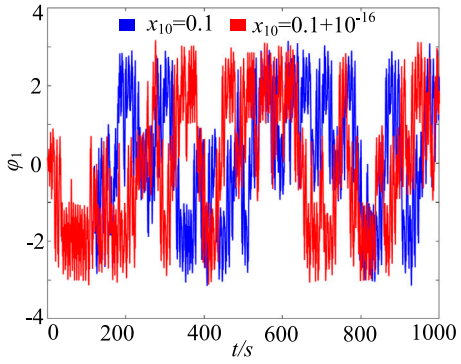


FIGURE 6
Time domain waveform of state variable φ_1.

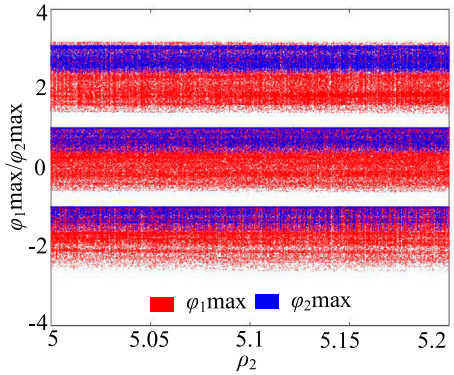


FIGURE 8
Bifurcation diagram of system (8) with ρ_2.

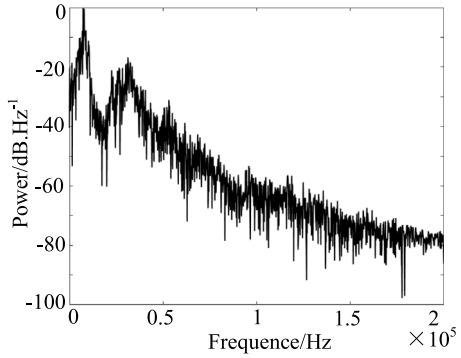


FIGURE 7
Power spectrum of system (8).

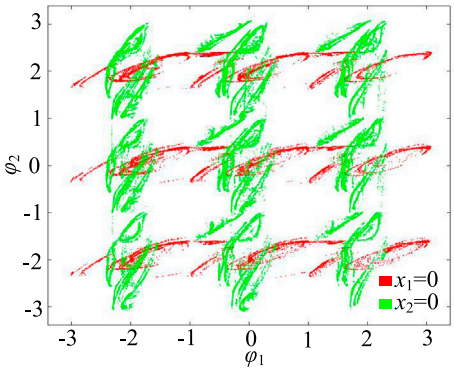


FIGURE 9
Poincaré maps of system (8) in the plane φ_1-φ_2.

3.3 Time series and power spectrum

When system parameters are kept unchanged, the timing waveform of system (8) in the direction of state variable φ_1 is shown by the blue curve in Figure 6. It can be seen from the figure that its motion trajectory oscillates randomly in a certain region and has distinct aperiodic characteristics. In addition, after slightly changing the initial value x_{10}, the time-domain waveform obtained by system (8) is shown by the red curve in Figure 6. It is not difficult to see that when the initial values are 0.1 and 0.1 + 10^{-16} respectively, there is no obvious difference between the two timings in (0, 120 s) time, but when t is greater than 120 s, they show completely different time evolution tracks. Therefore, even with a size adjustment of 10^{-16} for x_{10}, the trajectory of the system can make a huge difference, which means that the motion state of the system is not only aperiodic but also extremely sensitive to initial conditions. To further verify its non-periodicity, the power spectrum of system (8) is shown in Figure 7. It can be seen that the system exhibits a continuous power spectrum similar to a noise signal in a limited frequency range, which indicates that the system has obvious aperiodic chaotic characteristics.

3.4 Bifurcation diagrams and Poincaré maps

The bifurcation diagram is an important tool to describe the dynamic state of chaotic systems with parameter variation. Figure 8 shows the structure of local branch forks within the range of parameter $\rho_2 \in [5, 5.2]$, where red is the bifurcation in the direction of state variable φ_1, and blue is the bifurcation in the direction of state variable φ_2. It can be seen from Figure 8 that there are pieces of quasi-random points in the bifurcation diagram, indicating that the system is in a chaotic state. More importantly, different from the general chaotic bifurcation structure, the bifurcation diagram of system (8) presents three parallel bifurcation structures simultaneously in the directions of φ_1 and φ_2, and they are all in a chaotic state. This means that the system generates a 3 × 3 grid multi-structure chaotic attractor. In addition, Figure 9 shows the Poincaré map of a 3 × 3 grid multi-structure chaotic attractor in the φ_1-φ_2 phase plane. Among them, the red section diagram is generated by the $x_1 = 0$ section, and the green section diagram is generated by the $x_2 = 0$ section. Through observation, it is not difficult to find that the Poincare map of system (8) shows a series of irregular dense points, which further

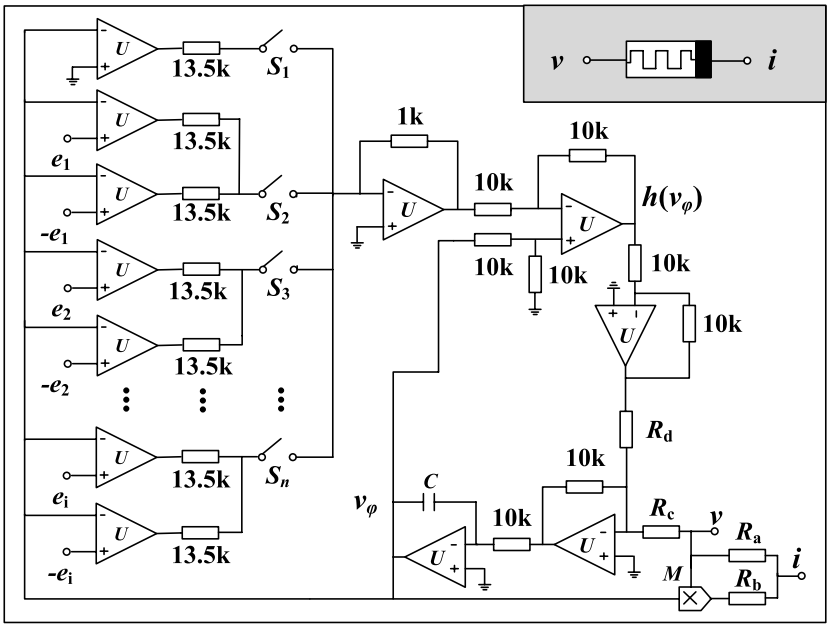


FIGURE 10
Memristor circuit.

proves that the grid multi-structure attractors have complex chaotic characteristics.

4 Circuit implementation and verification

Circuit realization is one of the effective ways to verify the mathematical model [56]. In general, chaotic systems can be implemented by analog or digital circuits to generate chaotic signals [57, 58]. In this section, we first design the physical circuit of system (8) and then verify the existence of grid multi-structure chaotic attractors by Multisim simulation.

4.1 Circuit design

First, a multi-segment non-linear memristor circuit is designed, as shown in Figure 10. Where, the internal state equation $h(\varphi)$ of the memristor is realized by a non-linear function generator, e1, e2,..., e_i is the control voltage, S1, S2,..., Sn is the selection switch. By selecting the corresponding control voltage and switch, the memristor function under different control parameters can be realized, as shown in Table 3.

Then, based on the memristor circuit, the physical circuit of system (8) is realized, as shown in Figure 11. Where the neuron activation function tanh(.) The equivalent circuit is available in Ref. [47]. In addition, the four neuronal membrane voltages are simulated by the four capacitor voltages of the circuit, and the fixed synaptic weight coefficients are simulated using resistors R_1–R_{10}. The memristor synaptic weight is achieved by the memristor. According to Kirchhoff's current law, the circuit equation of system (8) can be expressed as

$$\begin{cases} RC\dot{v}_1 = -v_1 + \frac{R}{R_1}\tanh(v_2) + R\left(\frac{1}{R_{a1}} + g\frac{v_{\varphi 1}}{R_{b1}}\right)\tanh(v_3) - \frac{R}{R_2}\tanh(v_4) \\ RC\dot{v}_2 = -v_2 - \frac{R}{R_3}\tanh(v_1) + \frac{R}{R_4}\tanh(v_2) + \frac{R}{R_5}\tanh(v_3) - R\left(\frac{1}{R_{a2}} + g\frac{v_{\varphi 2}}{R_{b2}}\right)\tanh(v_4) \\ RC\dot{v}_3 = -v_3 - \frac{R}{R_7}\tanh(v_2) + \frac{R}{R_8}\tanh(v_3) + \frac{R}{R_9}\tanh(v_4) \\ RC\dot{v}_4 = -v_4 + \frac{R}{R_{10}}\tanh(v_1) + \frac{R}{R_{11}}\tanh(v_4) \\ RC\dot{v}_{\varphi 1} = \frac{R}{R_c}\tanh(v_3) - \frac{R}{R_d}h(v_{\varphi 1}) \\ RC\dot{v}_{\varphi 2} = \frac{R}{R_c}\tanh(v_3) - \frac{R}{R_d}h(v_{\varphi 2}) \end{cases} \tag{13}$$

4.2 MULTISIM simulation

The physical circuit of system (8) is implemented and simulated on the Multisim14.0 platform. In this section, we first design the physical circuit of system (8) and then verify the existence of grid multi-structure chaotic attractors by Multisim simulation. Set capacitance $C = 1$ nF, $R = 10$ kΩ. Fixed synaptic weight coefficient according to system (8), setting $R_1 = R/w_{12} = 3.3$ kΩ, $R_2 = R/w_{14} = 0.77$ kΩ, $R_3 = R/w_{21} = 10$ kΩ, $R_4 = R/w_{22} = 6.7$ kΩ, $R_5 = R/w_{23} = 1.4$ kΩ, $R_6 = R/w_{32} = 2.5$ kΩ, $R_7 = R/w_{33} = 5.7$ kΩ, $R_8 = R/w_{34} = 2.5$ kΩ, $R_9 = R/w_{41} = 14.3$ kΩ, $R_{10} = R/w_{44} = 5$ kΩ. In addition, the resistance of the memristor circuit $R_a = R/\rho a$, $R_b = gR/\rho b$, $R_c = R/c$, $R_d = R/d$. When the system parameters $a = 1$, $b = 0.01$, $c = 2.2$, $d = 1.2$, $\rho_1 = 3$, $\rho_2 = 5$, $R_{a1} = 3.3$ kΩ, $R_{b1} = 34$ kΩ, $R_{a2} = 2$ kΩ, $R_{b2} = 20$ kΩ, $R_c = 4.5$ kΩ, $R_d = 8.4$ kΩ can be solved. Set the initial voltage of the five capacitor voltages to (0.1V, 0.1V, 0.1V, 0.1V). By setting the corresponding switching and control voltage according to Table 3, the system circuit can generate a grid multi-structure chaotic attractor consistent with the numerical simulation results, as

TABLE 3 Different memristor types realized by selecting the combination of switch and control voltage.

S_1	S_2	S_3	S_4	...	S_n	Memristor
e_1 = 1 V	e_2 = 3 V	e_3 = 5 V	e_4 = 7 V	...	e_i=(2n-1) V	$h_1(\varphi)$
Open	Open	Open	Open	...	Open	N = 0
Open	Off	Open	Open	...	Open	N = 1
Open	Off	Off	Open	...	Open	N = 2
...	...	...		...	...	...
e_1 = 2 V	e_2 = 4 V	e_3 = 6 V	e_4 = 8 V	...	e_i=(2n) V	$h_2(\varphi)$
闭合	断开	断开	断开	...	断开	M = 0
闭合	闭合	断开	断开	...	断开	M = 1
闭合	闭合	闭合	断开	...	断开	M = 2
...	...	...	...	...	...	...

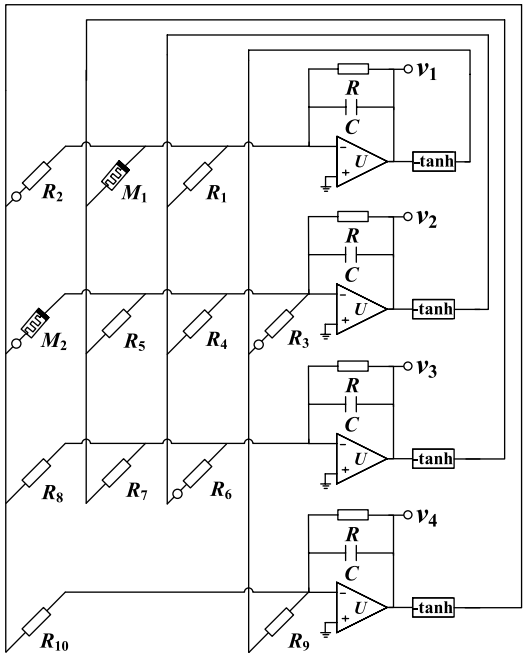


FIGURE 11
Physical circuits of system (8).

shown in Figure 12. By comparing the results of Figure 12 and Figure 13, it can be found that the results obtained by Multisim circuit simulation and Matlab numerical simulation are almost exactly the same, thus verifying the grid multi-structure chaotic attractor characteristics of system (8).

5 Medical image encryption applications

Chaotic signals have the characteristics of ergodic, initial value sensitivity, high randomness, etc., and are mainly used for the key generation of secure communication [59, 60]. Generally, the more chaotic the attractor trajectory of chaotic systems, the more sensitive the initial conditions, the higher the randomness of the key generated, and the better the encryption performance. Based on the above analysis results, the grid multi-structure chaotic attractor proposed in this paper has a complex structure, high initial value sensitivity, and strong chaos randomness, which can greatly improve the communication encryption effect and crack difficulty.

5.1 Image encryption scheme

In this section, a new image encryption scheme is designed based on the grid multi-structure chaotic attractor. The specific encryption steps are as follows:

Step 1: Set $(a, b, c, d, \rho_1, \rho_2, x_{20}, x_{30}, x_{40}, \varphi_{10}, \varphi_{20}, N_1, N_2)$ = (1, 0.01, 2.2, 1.2, 3, 5, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 1, 1), discarded number N_0 = 1000, time step Δt = 0.002, based fourth-order Runge-Kutta algorithm, the system (8) is continuously iterated 256 × 256 times. For each iteration, we can get six values $x_1(i)$, $x_2(i)$, $x_3(i)$, $x_4(i)$, $\varphi_1(i)$, $\varphi_2(i)$.

Step 2: The following preprocessing is performed on the six chaotic sequences to produce two distinct sets of sequences, respectively

$$K_1(i) = x_{1i} + x_{2i} + x_{3i} + x_{4i} \tag{14}$$

$$K_2(i) = \mathrm{mod}\left(\mathrm{floor}\left(\left(\left(x_{1i} + x_{2i} + x_{3i} + x_{4i} + \varphi_{1i} + \varphi_{2i}\right)/6\right)*10^{15}\right), 256\right) \tag{15}$$

Step 3: The index sequence index is obtained by arranging the sequence K1 in ascending order. Then, the index sequence is replaced and encrypted with the original image pixels successively to obtain the replaced image P1 as follows

$$P_1(i) = P(\mathrm{index}(K_1(i))) \tag{16}$$

Step 4: The pixels of the sequence K2 and the diagram P1 to be encrypted are different or encrypted as follows

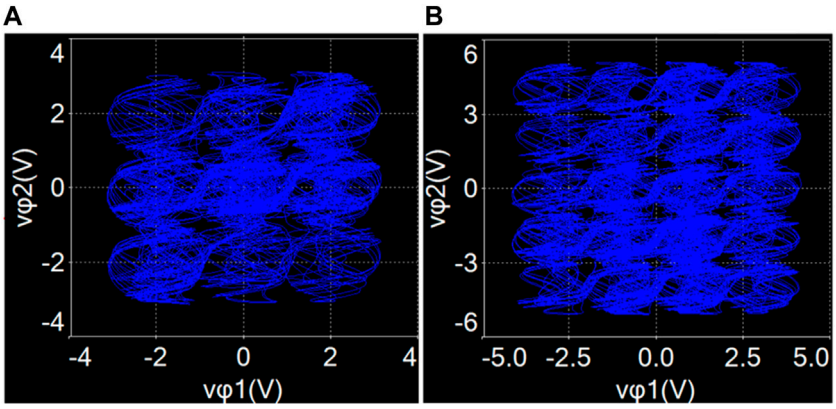


FIGURE 12
Experiment results. **(A)** 3 × 3-structure attractor. **(B)** 4 × 5-structure attractor.

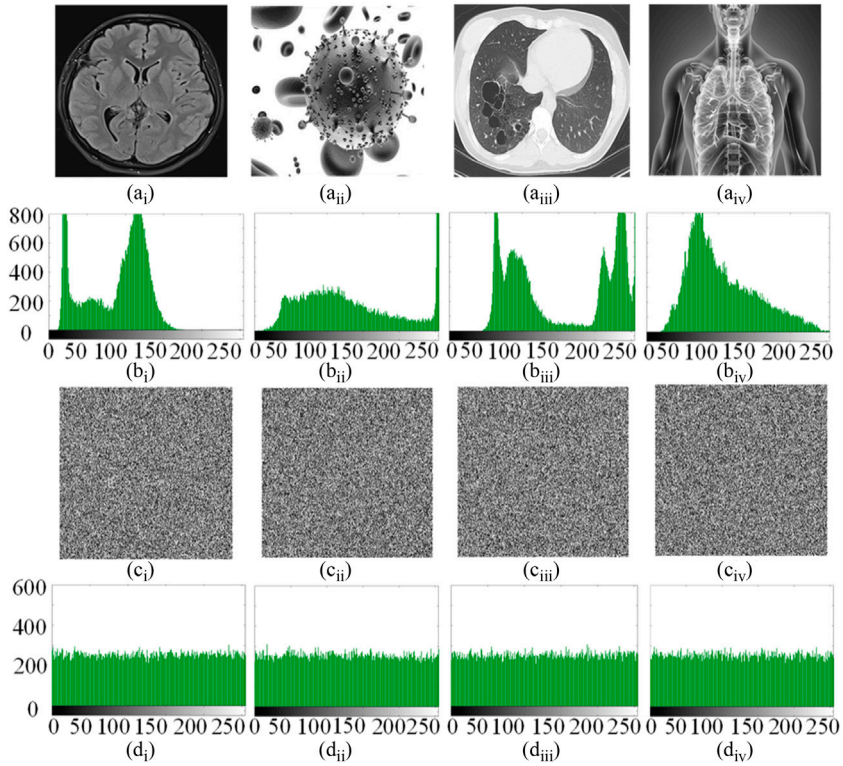


FIGURE 13
Encryption results. **(Ai–iv)** Original images. **(Bi–iv)** Histograms of the original images. **(Ci–iv)** Encrypted images. **(Di–iv)** Histograms of the encrypted images.

$$C(i) = P_1(i) \oplus K_2(i) \tag{17}$$

5.2 Encryption performance analysis

In order to prove the effectiveness and security of the above encryption algorithm, the following tests are carried out on the MATLAB R2017a platform. In the experiment, four 256 × 256 medical images were used as original images, which were brain image, virus image, lung image, and chest image, as shown in Figures 13Ai–Aiv. The key space, histogram, correlation coefficient, information trail, key sensitivity, noise and data loss attacks of the encryption system are analyzed below.

(1) Keyspace analysis. The secret key space is an important index to evaluate the encryption system. Generally speaking, the larger the key space of the encryption system, the stronger the resistance to external brute force attacks. The designed encryption algorithm uses the above 11 parameters as the

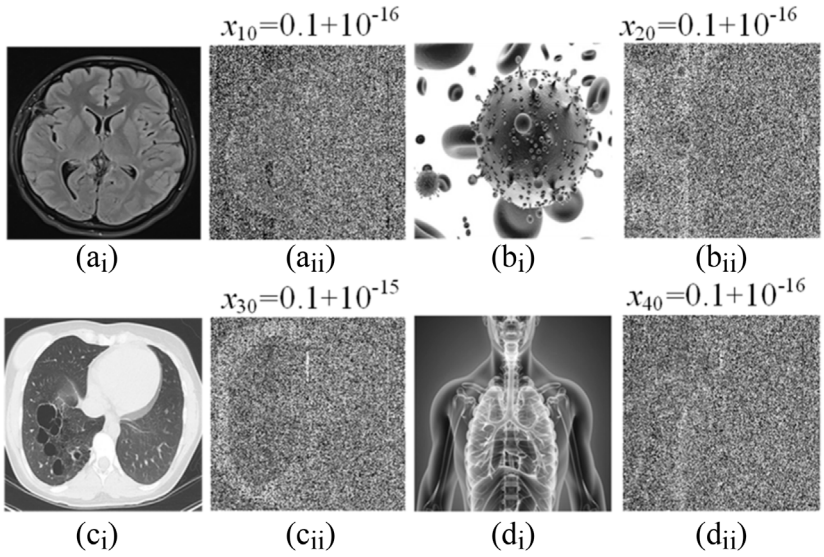


FIGURE 14
Test results of key sensitivity. **(Ai–Di)** Accurate decrypted images with right secret keys. **(Aii–Dii)** Inaccurate decrypted images with wrong secret keys.

secret key, making it difficult to decrypt illegally. In the experiment, all bytes are double-precision data, so the secret key space of the encryption system is $(10^{16})^{11} = 10^{176} \approx 2^{528}$. Obviously, this key space is much larger than the minimum key space value of 2^{100} to resist various brute force attacks.

(2) Histogram analysis. Histogram is a key index to evaluate the intensity distribution of pixel values in an image. The more average the histogram distribution of the image, the stronger the resistance to statistical attacks. Figure 13 shows the original image, the original image histogram, the encrypted image, and the encrypted image histogram of the four medical images respectively. Figures 13Ci–Civ look confusing and completely lose the information of the original images. In addition, by comparing Figures 14Bi–Biv and Figures 13Di–Div, we can see that the histogram of the encrypted image is very different from that of the original images. The histogram distribution of the original image is completely inconsistent and uneven, while the histogram distribution of the encrypted image is very average. Therefore, the designed encryption algorithm has good security against statistical analysis attacks.

(3) Correlation analysis. The correlation coefficient is an important index of image robustness. It represents the correlation between two adjacent pixels in an image. Usually, the correlation coefficient of the original image in all directions is relatively large, close to 1. The correlation of the encrypted image should be as small as possible, close to zero. In general, the correlation coefficient of the image can be calculated by the following formula

$$\rho_{xy} = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - E(y))^2}} \tag{18}$$

TABLE 4 Correlation coefficient and information origin between the original image and the encrypted image.

Images		Horizontal	Vertical	Diagonal	Entropy
Brain	Original	0.963966	0.964300	0.936475	6.3463
	Encrypted	0.009911	0.000687	−0.009079	7.9980
Virus	Original	0.965515	0.961770	0.940575	6.0599
	Encrypted	−0.007801	0.003794	0.000667	7.9974
Lung	Original	0.943080	0.954554	0.917790	7.1008
	Encrypted	−0.001244	0.000561	−0.002912	7.9975
Chest	Original	0.960183	0.920840	0.899727	7.4130
	Encrypted	−0.000161	0.001065	−0.002258	7.9975

where x and y represent two adjacent pixel values, and N represents the total number of pixels. $E(x)$ and $E(y)$ represent the average values of pixels x_i and y_i, respectively. In order to calculate the correlation coefficient between the original image and the encrypted image, 10,000 pixels are randomly selected for analysis. Table 4 shows the phase relation values of the four original images and the encrypted images. As can be seen from Table 4, although the correlation coefficients of the four original images are close to 1 in the vertical, horizontal, and diagonal directions, after encryption, the correlation coefficients of the four encrypted images are very close to 0. Therefore, the designed encryption algorithm can greatly reduce the correlation of images, so as to effectively resist statistical attacks.

(4) Entropy analysis. Information entropy is an important index to describe the degree of image uncertainty. In general, the larger the information origin value, the higher the randomness of the image information. An ideal value for grayscale images is 8. Therefore, a

TABLE 5 Comparison of information entropy of different encryption schemes.

References	[31]	[34]	[35]	[39]	[61]	This work
Entropy	7.9898	7.9976	7.9979	7.9975	7.9976	7.9980

good encryption algorithm should make the information descent of the encryption graph as close to 8 as possible. Information origin can be calculated by the following formula

$$H(P) = \sum_{i=0}^{2^N-1} P(x_i)\log_2\frac{1}{P(x_i)} \tag{19}$$

where N represents the bit depth of image P, and $P(x_i)$ represents the occurrence probability of pixel xi. Table 5 shows the information origin values of the four original and encrypted graphs. Obviously, compared with the origin of the original graph, the origin value of the encrypted graph is not only greatly improved, but also very close to the ideal value of 8. Therefore, the encryption algorithm has good encryption performance.

(5) Sensitivity analysis. The sensitivity of the key is an important index to evaluate the security performance of encryption algorithms. A good encryption algorithm should be as sensitive to the key as possible. When the keys a, b, c, d, ρ, φ_0, and N are fixed and the values of the keys x_{10}, x_{20}, x_{30}, and x_{40} are changed, the key sensitivity of the four images is measured. The experimental results are shown in Figure 15. For the small change of the secret key, even with the disturbance of 10–16, the decrypted image still cannot decrypt the encryption diagram correctly. As shown in Table 6, compared with the existing similar work, the secret key sensitivity value of the image encryption algorithm based on multi-scroll memristor HNN in literature [34] is 10^{-9}, and the secret key sensitivity value of the image encryption algorithm based on multi-scroll chaotic system in literature [35] is 10^{-12}. The designed image encryption algorithm based on multi-structure attractor memristor HNN has great advantages in terms of secret key sensitivity.

(6) Data loss and noise attacks. Noise and data loss attacks can be used to evaluate the robustness of cryptosystems. On the one hand, the image is prone to sudden and strong noise interference in the transmission process, which will produce noise and affect the correct decryption of the image. Therefore, for image encryption systems, resisting noise attacks is a very important performance index. Here, different proportions of salt and pepper noise are added to the encrypted image, as shown in Figures 15Ai–Aiii. Then the secret key is used to decrypt the encrypted image, as shown in Figures 15Bi–Biii. It can be observed that although some pixels of the decrypted image are changed after the addition of salt and pepper noise, most of the main information of the original image can still be obtained. Therefore, the noise has very little effect on the decryption process of the designed encryption algorithm. On the other hand, it is easy to lose some data in the process of encrypted image transmission. Therefore, a good encryption algorithm should be robust to partial data loss. Here, the data of different sizes of the encryption graph is cut out, as shown in Figures 15Ci–Ciii. Then, using the secret key to decrypt it, the results are shown in Figures 15Di–Diii. Obviously, although the encrypted graph has lost some data information, it can still recover most of the original image information through the decryption process.

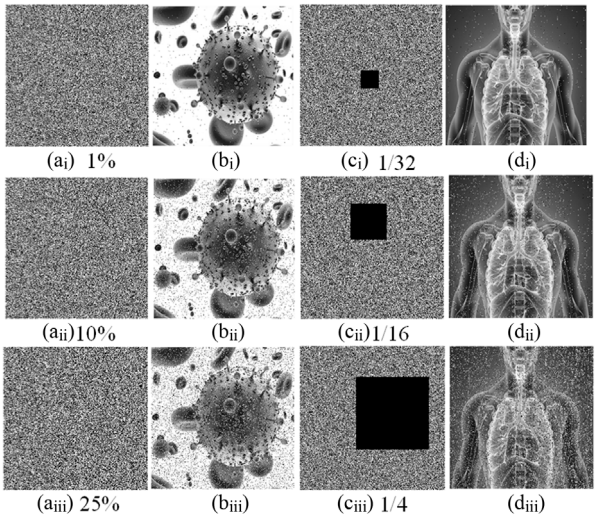


FIGURE 15
Test results of data loss and noise attacks. **(Ai–iii)** The encrypted images with 1%, 10%, and 25% salt and pepper noise, respectively. **(Bi–iii)** Corresponding decrypted images. **(Ci–iii)** The encrypted images with 1/32, 1/16, and 1/4 data loss, respectively. **(Di–iii)** Corresponding decrypted images.

TABLE 6 Comparison of key sensitivity of different encryption schemes.

References	[32]	[34]	[35]	[39]	[62]	This work
Key sensitivity	10^{-12}	10^{-9}	10^{-12}	10^{-6}	10^{-15}	10^{-16}

TABLE 7 Comparison results of the encryption time.

References	Images	Time (s)
This work	Brain (256 × 256)	0.125
	Virus (512 × 512)	0.273
	Lung (768 × 768)	0.854
[32]	Lena (256 × 256)	0.241
[34]	Lena (256 × 256)	0.876
[35]	Lena (256 × 256)	0.546

Therefore, the experimental results show that the designed encryption algorithm has strong resistance to data loss attacks.

(7) Encryption time analysis. Regardless of the security performance, encryption time is also very important, especially in emergency medical care applications. The time test of the scheme is operated using MATLAB R2017a on a PC with Windows 10 64-bit operation system, 2.5 GHz CPU, and 8 GB RAM. All medical images are tested, and the comparison results of one round of encryption execution time are listed in Table 7. It is clear that the execution time of our scheme is shorter than the other schemes.

The above experimental results show that the designed medical image encryption algorithm based on a multi-structure chaotic attractor

has a larger key space and higher key sensitivity, can effectively resist various internal and external attacks, and can be applied to protect image data better in practical information communication.

6 Conclusion

Based on the chaotic property of the Hopfield neural network and the memristor synapse control method, a grid multi-structure attractor chaotic system is proposed in this paper. Its complex and varied chaotic structure provides a new idea for the study of new grid chaotic attractors. The basic dynamic analysis, such as equilibrium point, Lyapunov exponential spectrum, power spectrum, bifurcation diagram, and Poincare cross section, shows that the multi-structure chaotic attractor has the characteristics of simple implementation, few control parameters, complex unit topology, expandable unit structure, and complex chaotic dynamics. At the same time, based on the chaotic circuit design method, the circuit simulation of the chaotic attractor is realized, and the feasibility of the system circuit is verified. Finally, combined with the study of chaotic image encryption theory, an image encryption scheme based on the grid multi-structure attractor is designed. The experimental results show that the new grid multi-structure chaotic attractor has certain advantages in encrypted communication and has a good application prospect. The next step will consider introducing multiple memristor synapses into neural networks to construct n-dimensional grid multi-structure chaos attractors to improve the complexity of chaotic systems.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

ZH: Methodology, Software, Writing–original draft. HL: Data curation, Formal Analysis, Validation, Investigation, Writing–review and editing. CW: Conceptualization, Funding acquisition, Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work is supported by the National Natural Science Foundation of China (62271197, 61971185, and 62201204), the China Postdoctoral Science Foundation (2022M71104), and the Natural Science Foundation of Hunan Province (2023JJ40168).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Pham VT, Volos C, Jafari S, Wei Z, Wang X. Constructing a novel no-equilibrium chaotic system. *Int J Bifurcation Chaos* (2014) 24(05):1450073. doi:10.1142/s0218127414500734

2. Lai Q, Kuate PDK, Liu F, Iu HHC. An extremely simple chaotic system with infinitely many coexisting attractors. *IEEE Trans Circuits Syst Express Briefs* (2019) 67(6):1129–33. doi:10.1109/tcsii.2019.2927371

3. Wen Z, Wang C, Deng Q, Lin H. Regulating memristive neuronal dynamical properties via excitatory or inhibitory magnetic field coupling. *Nonlinear Dyn* (2022) 110(4):3823–35. doi:10.1007/s11071-022-07813-9

4. Lorenz EN. Deterministic nonperiodic flow. *J Atmos Sci* (1963) 20(2):130–41. doi:10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2

5. Fan W, Chen X, Wang Y, Chen B, Wu H, Xu Q. Hidden firing patterns and memristor initial condition-offset boosting behavior in a memristive Hindmarsh-Rose neuron model. *Front Phys* (2023) 11:1160419. doi:10.3389/fphy.2023.1160419

6. Sun J, Li C, Lu T, Akgul A, Min F. A memristive chaotic system with hypermultistability and its application in image encryption. *Ieee Access* (2020) 8: 139289–98. doi:10.1109/access.2020.3012455

7. Lin H, Wang C, Du S, Yao W, Sun Y. A family of memristive multibutterfly chaotic systems with multidirectional initial-based offset boosting. *Chaos, Solitons & Fractals* (2023) 172:113518. doi:10.1016/j.chaos.2023.113518

8. Sahoo S, Roy BK. A new multi-wing chaotic attractor with unusual variation in the number of wings. *Chaos, Solitons & Fractals* (2022) 164:112598. doi:10.1016/j.chaos.2022.112598

9. Chua LO, Lin GN. Canonical realization of Chua's circuit family. *IEEE Trans Circuits Syst* (1990) 37(7):885–902. doi:10.1109/31.55064

10. Chen G, Ueta T. Yet another chaotic attractor. *Int J Bifurcation chaos* (1999) 9(07): 1465–6. doi:10.1142/s0218127499001024

11. Lv J, Chen G. A new chaotic attractor coined. *Int J Bifurcation chaos* (2002) 12(03): 659–61.

12. Wang Q, Tian Z, Wu X, Tan W. Coexistence of multiple attractors in a novel simple jerk chaotic circuit with CFOAs implementation. *Front Phys* (2022) 10:41. doi:10.3389/fphy.2022.835188

13. Njitacke ZT, Sone ME, Fozin TF, Tsafack N, Leutcho GD, Tchapga CT. Control of multistability with selection of chaotic attractor: Application to image encryption. *Eur Phys J Spec Top* (2021) 230(7-8):1839–54. doi:10.1140/epjs/s11734-021-00137-6

14. Yan D, Wang L, Duan S, Chen J, Chen J. Chaotic attractors generated by a memristor-based chaotic system and Julia fractal. *Chaos, Solitons & Fractals* (2021) 146: 110773. doi:10.1016/j.chaos.2021.110773

15. Liu T, Yan H, Banerjee S, Mou J. A fractional-order chaotic system with hidden attractor and self-excited attractor and its DSP implementation. *Chaos, Solitons & Fractals* (2021) 145:110791. doi:10.1016/j.chaos.2021.110791

16. Gong LH, Luo HX, Wu RQ, Zhou NR. New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG. *Physica A: Stat Mech its Appl* (2022) 591:126793. doi:10.1016/j.physa.2021.126793

17. Leonov GA, Kuznetsov NV, Vagaitsev VI. Localization of hidden Chua's attractors. *Phys Lett A* (2011) 375(23):2230–3. doi:10.1016/j.physleta.2011.04.037

18. Zhang S, Wang X, Zeng Z. A simple no-equilibrium chaotic system with only one signum function for generating multidirectional variable hidden attractors and its hardware implementation. *Chaos* (2020) 30(5):053129. doi:10.1063/5.0008875

19. Chen M, Wang C, Bao H, Ren X, Bao B, Xu Q. Reconstitution for interpreting hidden dynamics with stable equilibrium point. *Chaos, Solitons & Fractals* (2020) 140: 110188. doi:10.1016/j.chaos.2020.110188

20. Li C, Peng Y, Tao Z, Sprott JC, Jafari S. Coexisting infinite equilibria and chaos. *Int J Bifurcation Chaos* (2021) 31(05):2130014. doi:10.1142/s0218127421300147

21. Bo-Cheng B, Zhong L, Jian-Ping X. Transient chaos in smooth memristor oscillator. *Chin Phys B* (2010) 19(3):030510. doi:10.1088/1674-1056/19/3/030510

22. Lin H, Wang C, Cui L, Sun Y, Zhang X, Yao W. Hyperchaotic memristive ring neural network and application in medical image encryption. *Nonlinear Dyn* (2022) 110(1):841–55. doi:10.1007/s11071-022-07630-0

23. Wang FP, Wang FQ. Multistability and coexisting transient chaos in a simple memcapacitive system. *Chin Phys B* (2020) 29(5):058502. doi:10.1088/1674-1056/ab7e98

24. Lin H, Wang C, Yu F, Sun J, Du S, Deng Z, et al. A review of chaotic systems based on memristive Hopfield neural networks. *Mathematics* (2023) 11(6):1369. doi:10.3390/math11061369

25. Li S, Wu Y, Zheng G. Adaptive synchronization for hyperchaotic liu system. *Front Phys* (2022) 9:812048. doi:10.3389/fphy.2021.812048

26. Sprott JC. Some simple chaotic flows. *Phys Rev E* (1994) 50(2):647–50. doi:10.1103/physreve.50.r647

27. Sprott JC. A new chaotic jerk circuit. *IEEE Trans Circuits Syst Express Briefs* (2011) 58(4):240–3. doi:10.1109/tcsii.2011.2124490

28. Li C, Yang Y, Yang X, Zi X, Xiao F. A tristable locally active memristor and its application in Hopfield neural network. *Nonlinear Dyn* (2022) 108(2):1697–717. doi:10.1007/s11071-022-07268-y

29. Zhou W, Wang G, Iu HHC, Shen Y, Liang Y. Complex dynamics of a non-volatile memcapacitor-aided hyperchaotic oscillator. *Nonlinear Dyn* (2020) 100:3937–57. doi:10.1007/s11071-020-05722-3

30. Yang Y, Huang L, Xiang J, Guo Q. Three-dimensional sine chaotic system with multistability and multi-scroll attractor. *IEEE Trans Circuits Syst Express Briefs* (2022) 69(3):1792–6. doi:10.1109/tcsii.2021.3126227

31. Wang N, Li C, Bao H, Chen M, Bao B. Generating multi-scroll Chua's attractors via simplified piecewise-linear Chua's diode. *IEEE Trans Circuits Syst Regular Pap* (2019) 66(12):4767–79. doi:10.1109/tcsi.2019.2933365

32. Hong Q, Li Y, Wang X, Zeng Z. A versatile pulse control method to generate arbitrary multidirection multibutterfly chaotic attractors. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2018) 38(8):1480–92. doi:10.1109/tcad.2018.2855121

33. Lai Q, Wan Z, Zhang H, Chen G. Design and analysis of multiscroll memristive hopfield neural network with adjustable memductance and application to image encryption. *IEEE Trans Neural Networks Learn Syst* (2022) 1–14. doi:10.1109/tnnls.2022.3146570

34. Lin H, Wang C, Yu F, Xu C, Hong Q, Yao W, et al. An extremely simple multiwing chaotic system: Dynamics analysis, encryption application, and hardware implementation. *IEEE Trans Ind Electron* (2020) 68(12):12708–19. doi:10.1109/tie.2020.3047012

35. Zhang S, Li C, Zheng J, Wang X, Zeng Z, Peng X. Generating any number of initial offset-boosted coexisting Chua's double-scroll attractors via piecewise-nonlinear memristor. *IEEE Trans Ind Electron* (2021) 69(7):7202–12. doi:10.1109/tie.2021.3099231

36. Escalante RJ, Campos E. A class of piecewise linear systems without equilibria with 3-D grid multiscroll chaotic attractors. *IEEE Trans Circuits Syst Express Briefs* (2018) 66(8):1456–60.

37. Ding PF, Feng XY, Wu CM. Novel two-directional grid multi-scroll chaotic attractors based on the Jerk system. *Chin Phys B* (2020) 29(10):108202. doi:10.1088/1674-1056/ab9dea

38. Azam A, Aqeel M, Sunny DA. Generation of multidirectional mirror symmetric multiscroll chaotic attractors (MSMCA) in double wing satellite chaotic system. *Chaos, Solitons & Fractals* (2022) 155:111715. doi:10.1016/j.chaos.2021.111715

39. Peng X, Zeng Y. Image encryption application in a system for compounding self-excited and hidden attractors. *Chaos, Solitons & Fractals* (2020) 139:110044. doi:10.1016/j.chaos.2020.110044

40. Lin H, Wang C, Sun Y. A universal variable extension method for designing multiscroll/wing chaotic systems. *IEEE Trans Ind Electron* (2023) 1–13. doi:10.1109/TIE.2023.3299020

41. Xu C, Wang C, Jiang J, Sun J, Lin H. Memristive circuit implementation of context-dependent emotional learning network and its application in multitask. *IEEE Trans Computer-Aided Des Integrated Circuits Syst* (2021) 41(9):3052–65. doi:10.1109/tcad.2021.3116463

42. Liao M, Wang C, Sun Y, Lin H, Xu C. Memristor-based affective associative memory neural network circuit with emotional gradual processes. *Neural Comput Appl* (2022) 34(16):13667–82. doi:10.1007/s00521-022-07170-z

43. Jiang D, Njitacke ZT, Nkapkop JDD, Tsafack N, Wang X, Awrejcewicz J. A new cross ring neural network: Dynamic investigations and application to WBAN. *IEEE Internet Things J* (2022) 10(8):7143–52. doi:10.1109/jiot.2022.3228748

44. Zhou C, Wang C, Yao W, Lin H. Observer-based synchronization of memristive neural networks under DoS attacks and actuator saturation and its application to image encryption. *Appl Maths Comput* (2022) 425:127080. doi:10.1016/j.amc.2022.127080

45. Zhou C, Wang C, Sun Y, Yao W, Lin H. Cluster output synchronization for memristive neural networks. *Inf Sci* (2022) 589:459–77. doi:10.1016/j.ins.2021.12.084

46. Hopfield JJ. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci* (1984) 81(10): 3088–92. doi:10.1073/pnas.81.10.3088

47. Bao B, Chen C, Bao H, Zhang X, Xu Q, Chen M. Dynamical effects of neuron activation gradient on Hopfield neural network: Numerical analyses and hardware experiments. *Int J Bifurcation Chaos* (2019) 29(04):1930010. doi:10.1142/s0218127419300106

48. Lin H, Wang C, Chen C, Sun Y, Zhou C, Xu C, et al. Neural bursting and synchronization emulated by neural networks and circuits. *IEEE Trans Circuits Syst Regular Pap* (2021) 68(8):3397–410. doi:10.1109/tcsi.2021.3081150

49. Isaac SD, Njitacke ZT, Tsafack N, Tchapga CT, Kengne J. Novel compressive sensing image encryption using the dynamics of an adjustable gradient Hopfield neural network. *Eur Phys J Spec Top* (2022) 231(10):1995–2016. doi:10.1140/epjs/s11734-022-00472-2

50. Chen C, Min F, Zhang Y, Bao H. ReLU-type Hopfield neural network with analog hardware implementation. *Chaos, Solitons & Fractals* (2023) 167:113068. doi:10.1016/j.chaos.2022.113068

51. Lin H, Wang C, Sun Y, Tang Z, Xie F, Chen D, et al. Analysis of outer membrane vesicles indicates that glycerophospholipid metabolism contributes to early symbiosis between *Sinorhizobium fredii* HH103 and soybean. *IEEE Trans Circuits Syst Express Briefs* (2022) 70(1):311–22. doi:10.1094/MPMI-11-21-0288-R

52. Yu F, Kong X, Mokbel AAM, Yao W, Cai S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive Hopfield neural network with a novel local active memeristor. *IEEE Trans Circuits Syst Express Briefs* (2022) 70(1):326–30. doi:10.1109/tcsii.2022.3218468

53. Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi-scroll memristive Hopfield neural network. *IEEE Trans Netw Sci Eng* (2022) 10(2):845–58. doi:10.1109/tnse.2022.3223930

54. Lai Q, Wan Z, Kuate PDK. Generating grid multi-scroll attractors in memristive neural networks. *IEEE Trans Circuits Syst Regular Pap* (2022) 70(3):1324–36. doi:10.1109/tcsi.2022.3228566

55. Strukov DB, Snider GS, Stewart DR, Williams RS. The missing memristor found. *nature* (2008) 453(7191):80–3. doi:10.1038/nature06932

56. Xia Z, Wang C, Jin J, Du S, Lin H, Yang H. Novel AM/FM/ASK/FSK/PSK/QAM signal generator based on a digitally programmable CDTA. *Circuits, Systems, Signal Process.* (2015) 34(5):1635–53. doi:10.1007/s00034-014-9921-3

57. Liu X, Wang J. The simplest memristor circuit with hyperchaos. *Front Phys* (2022) 10:904200. doi:10.3389/fphy.2022.904200

58. Wang M, An M, He S, Zhang X, Ho-Ching Iu H, Li Z. Two-dimensional memristive hyperchaotic maps with different coupling frames and its hardware implementation. *Chaos* (2023) 33(7):073129. doi:10.1063/5.0154516

59. He S, Liu J, Wang H, Sun K. A discrete memristive neural network and its application for character recognition. *Neurocomputing* (2023) 523:1–8. doi:10.1016/j.neucom.2022.12.014

60. Zhu W, Sun K, He S, Wang H, Liu W. A class of m-dimension grid multi-cavity hyperchaotic maps and its application. *Chaos, Solitons & Fractals* (2023) 170:113370. doi:10.1016/j.chaos.2023.113370

61. Sambas A, Vaidyanathan S, Zhang X, Koyuncu I, Bonny T, Tuna M, et al. A novel 3D chaotic system with line equilibrium: Multistability, integral sliding mode control, electronic circuit, FPGA implementation and its image encryption. *IEEE Access* (2022) 10:68057–74. doi:10.1109/access.2022.3181424

62. Sambas A, Vaidyanathan S, Tlelo-Cuautle E, Abd-El-Atty B, El-Latif AAA, Guillen-Fernandez O, et al. A 3-D multi-stable system with a peanut-shaped equilibrium curve: Circuit design, FPGA realization, and an application to image encryption. *IEEE Access* (2020) 8:137116–32. doi:10.1109/access.2020.3011724

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 12 October 2023
DOI 10.3389/fphy.2023.1292336

OPEN ACCESS

EDITED BY
Chunbiao Li,
Nanjing University of Information Science and Technology, China

REVIEWED BY
Kehui Sun,
Central South University, China
Fuhong Min,
Nanjing Normal University, China

*CORRESPONDENCE
Xiao-Yuan Wang,
youyuan-0213@163.com

RECEIVED 11 September 2023
ACCEPTED 27 September 2023
PUBLISHED 12 October 2023

CITATION
Li X-J, Wang X-Y, Li P, Iu HHC and Cheng Z-Q (2023), Ternary combinational logic gate design based on tri-valued memristors.
Front. Phys. 11:1292336.
doi: 10.3389/fphy.2023.1292336

Ternary combinational logic gate design based on tri-valued memristors

Xiao-Jing Li[1,2], Xiao-Yuan Wang[1,2]*, Pu Li[1], Herbert H. C. Iu[3] and Zhi-Qun Cheng[1]

[1]Wenzhou Institute, Hangzhou Dianzi University, Wenzhou, China, [2]School of Electronics and Information, Hangzhou Dianzi University, Hangzhou, China, [3]School of Electrical and Electronic Engineering, The University of Western Australia, Crawley, WA, Australia

Traditional binary combinational logic circuits are generally obtained by cascading multiple basic logic gate circuits, using more components and complicated wiring. In contrast to the binary logic circuit design in this method, ternary combinational logic circuit implementation is more complicated. In this paper, a ternary circuit design method that does not require cascading basic ternary logic gates is proposed based on a tri-valued memristor, which can directly realize specific logic functions through a series connection of memristors. The ternary encoder, ternary decoder, ternary comparator, and ternary data selector are implemented by this method, and the effectiveness of the circuits is verified by LTspice simulations.

KEYWORDS

tri-valued memristor, ternary encoder, ternary decoder, ternary comparator, ternary data selector

1 Introduction

Traditional digital systems are built on binary numbers, where only two levels are considered, namely, “0” and “1.” However, with the rapid development of modern information technology and the increasing integration of digital systems, interconnection limitations have become one of the main challenges in implementing the binary logic in the nano-scale circuit design [1]. Interconnect lines cause increased latency, noise, and power consumption in the system. In multi-valued logic (MVL), a single signal line carries more information, which can effectively reduce the number of interconnecting lines and solve the interconnection problem in binary digital systems [2]. After the calculation demonstration in [3], it can be seen that when the base is e, the complexity and cost of the multi-valued system are the lowest and 3 is the integer closest to e. Therefore, compared with binary logic, ternary logic has significant advantages. In 1840, the British mathematician and inventor Thomas Fowler first proposed the concept of a ternary computer, but the relevant details have long been lost. Glusker studied and sorted it, and elaborated the relevant concepts of ternary logic and ternary computer in [4] in 2005. Although the same number of binary signals are easier to process than ternary signals, ternary signals can carry more information, which can effectively improve the efficiency of information transmission and storage. Therefore, compared with binary logic, ternary logic has stronger information processing capabilities and can solve some complex and cumbersome problems in binary logic.

Memristor is a nonlinear nano-component with many excellent properties such as stable resistive performance, low power consumption, and compatibility with traditional CMOS technology. In particular, the circuit realized by the memristor can have both operation and

storage functions, so a memristor is considered to be a strong competitor to replace traditional silicon chips and continue Moore's law. Compared with traditional digital logic circuits using CMOS, memristor-based digital logic circuits can effectively reduce the area and power consumption of the circuit [5–9]. At present, there are still few research studies on ternary logic based on the memristor, and most of them need to realize corresponding functions by memristors combined with complementary metal–oxide semiconductor (CMOS). In 2016, Khalid designed a basic ternary logic gate using a circuit structure similar to MRL (hybrid CMOS/memristive logic gate) [10], which reduced the number of components. In 2020, Zhang designed flexible logic circuits based on spintronic memristors and CMOS switches to implement basic unbalanced logic gates with non-volatility, good load capacity, and constant voltage input and output without signal degradation [11]. In the same year, Wang designed ternary AND gates, OR gates, NOT gates, and maximum and minimum circuits by utilizing the compatibility of memristors and CMOS, and achieved an order of magnitude improvement in data density; the switching speed of the memristor is reduced by a factor of about 13 [12]. However, the aforementioned circuits require CMOS devices to work together, which leads to more circuit area and power consumption, as well as more complex operation steps and the amount of running power supply. In addition, some of the ternary logic circuits are designed by the memristor combined with the carbon nanotube field-effect transistor (CNTFET). In 2018, basic ternary logic gates, ternary decoders, 2-bit adders, and standard ternary inverters based on memristors and CNTFETs were proposed [13,14]. In 2019, Soliman proposed a systematic method for constructing a 2-bit ternary function based on the concept of the memristive threshold logic [15]. A 2-bit ternary adder and multiplier are implemented using VTEAM memristors and Stanford CNTFET transistor models. In the same year, Chen fabricated a memristor device based on nano-columnar crystalline ZnO thin films and used it to realize a complete set of ternary logic and a ternary multiplier unit [16]. Nevertheless, the logic variables of these circuits are voltage, and the memristors are only used as a computing unit rather than a storage unit. Therefore, these designs have the problems of signal degradation and loss of power-off information.

Due to the lack of actual ternary components corresponding to ternary logic, the promotion of ternary logic in practical applications is not smooth. Based on the asymmetric piecewise linear memristive mathematical model [17] extended by Chua on the basis of the voltage-controlled odd-symmetric piecewise linear memristive model in 2015, Wang's team first proposed the concept of the tri-valued memristor in 2019, obtained tri-valued and multi-valued memristor models on this asymmetric piecewise linear memristor model, and realized a new chaotic circuit based on the tri-valued memristor model [12]. In [18], a method for realizing ternary basic logic gates based on tri-valued memristors is proposed. The designed logic gates do not need to use other CMOS devices, and the power consumption of the circuit is lower. As a multi-valued memristor, the tri-valued memristor can exhibit three different resistance states without using any additional devices to represent "0," "1," and "2" in ternary logic, and the application of tri-valued memristors to digital logic circuits can further reduce circuit power consumption and the circuit area, which improves the storage density. In addition, a tri-valued memristor provides a non-traditional computing architecture, that is, combines information storage and processing, which create favorable conditions for the realization of ternary logic circuits.

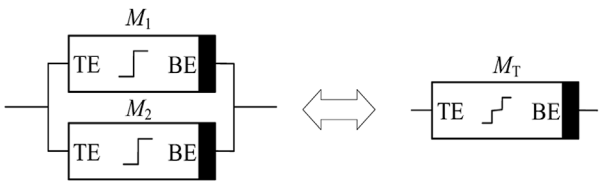


FIGURE 1
Constructing a tri-valued memristor in parallel with binary memristors.

The rest of this paper is as follows: Section 2 explains the process of constructing a voltage-controlled tri-valued memristor, and the threshold characteristics and pinched hysteresis loop of this memristor are analyzed. Section 3 proposes a tri-value circuit design method using the resistance state of the memristor to represent the logic value. This method does not require cascading the basic tri-value logic gates and allows specific logic functions to be implemented directly through series and parallel connections of the memristors. Concretely, the ternary combinational logic gate circuits based on a tri-valued memristor include a ternary encoder, a decoder, a comparator, and a data selector, and the designed circuit is verified by LTspice. Section 4 gives the summary of this paper.

2 A tri-valued memristor built by two common binary memristors

Ternary logic is a multivalued logic with three different logic states. The tri-valued memristor is a crude element which exhibits three different states (resistances) matching "0," "1," and "2" in the ternary logic without using any additional devices. In this paper, the resistances of the tri-valued memristor are used as the logic state variables, where R_H, R_M, and R_L are used to represent logics "0," "1," and "2". The voltage threshold tri-valued memristor explained in this paper and its threshold characteristics are introduced as follows.

2.1 Modeling of a tri-valued memristor

In 2021, Wang proposed a voltage threshold tri-valued memristor model that has the characteristics of simple structure and clear principle. The model parameters can be modified according to the actual application requirements [18]. In [19], an implementation method of connecting two tri-valued memristors in series and parallel to obtain a multi-valued memristor was proposed. As a result, a voltage-controlled tri-valued memristor model is obtained in this paper by connecting two voltage-controlled binary memristors in parallel, and the resulting tri-valued memristor model is used to design ternary combinational logic circuits. It is also proved that the tri-valued memristor model in [18] can be realized using the binary memristor through the circuit structure.

TABLE 1 Parameter values of the two binary memristors.

Parameter	M_1	M_2
R_{on}	500Ω	125Ω
R_{off}	(4000/3)Ω	2000Ω
V_{on}	1 V	1.2 V
V_{off}	1.2 V	1 V

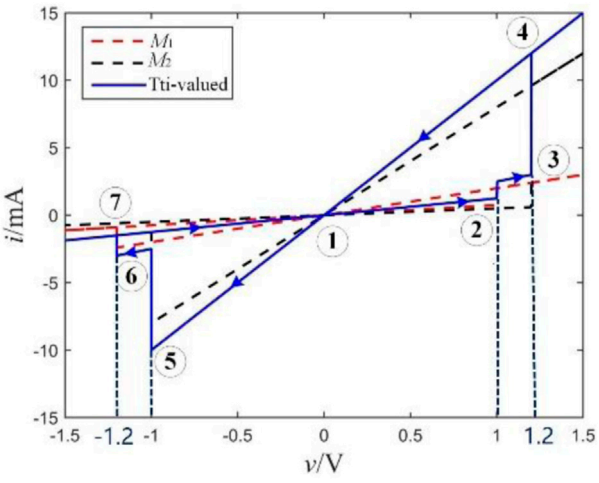


FIGURE 2
v-i curve of the tri-valued, M_1 and M_2 memristors.

In 2015, Knowm Inc. company designed and produced a voltage threshold binary memristor called the Knowm memristor. In this paper, by modifying the expression of $G(v)$ of the Knowm memristor from $G(v) = v/R_{on} + (1- v)/R_{off}$ to $G(v) = \text{IF}\ (v < 0.5, 1/R_{off}, 1/R_{on})$, a better threshold binary memristor model is obtained. Based on this model, we obtained a tri-valued memristor by putting two Knowm memristors in parallel, as shown in Figure 1, with parameters shown in Table 1. The *v-i* curves of two single memristors (M_1 and M_2) and the tri-valued memristor with three distinct resistance states are shown in Figure 2.

2.2 Characteristics of the tri-valued memristor

The tri-valued memristor mentioned previously has voltage threshold characteristics, and its two threshold voltages are $v_{th1} = 1$ V and $v_{th2} = 1.2$ V, respectively. R_L, R_M, and R_H correspond to three different resistance states of the model. According to the characteristics of the tri-valued memristor, voltages with different amplitudes (applied to the positive electrode of the memristor) will be used to initialize the memristor and implement the update of the memristor resistance value in the subsequent logic gate design. Specifically, when the voltage $v \geq 1.2$ V, the memristor switches from any resistance state to R_L. When $1\ \text{V} \leq v < 1.2$ V, if the resistance state of the memristor is R_H at this time, it will switch to R_M; otherwise, it will remain unchanged. When $-1\ \text{V} < v < 1$ V, the memristor will continue to maintain its original state. When $-1.2\ \text{V} < v \leq -1$ V, if the resistance state of the memristor is R_L at this time, it will switch to R_M; otherwise, it will not change. When $v \leq -1.2$ V, the memristor switches from any resistance state to R_H. Table 2 summarizes the voltage range required for the resistance state switching of the tri-valued memristor. The symbol "→" indicates that the resistance state on the left side is switched to that on the right side of the symbol.

TABLE 2 Voltage range required for resistance state switching.

Resistance state switching	Voltage range
$R_L \rightarrow R_L$	$v > -1$ V
$R_L \rightarrow R_M$	$-1.2\ \text{V} < v \leq -1$ V
$R_L \rightarrow R_H$	$v \leq -1.2$ V
$R_M \rightarrow R_L$	$v \geq 1.2$ V
$R_M \rightarrow R_M$	$-1.2\ \text{V} < v < 1.2$ V
$R_M \rightarrow R_H$	$v \leq -1.2$ V
$R_H \rightarrow R_L$	$v \geq 1.2$ V
$R_H \rightarrow R_M$	$1\ \text{V} \leq v < 1.2$ V
$R_H \rightarrow R_H$	$v < 1$ V

In order to verify the threshold characteristics of the tri-valued memristor model, the voltage signals $v = t$ and $v = -t$ are applied to the memristor models with initial states of R_H and R_L, respectively, and the LTspice simulation results, as shown in Figure 3, are obtained, which indicate that the resistance switching process of the memristor is consistent with Table 2, proving that the tri-valued memristor built exhibit the threshold characteristics.

2.3 Analysis of the pinched hysteresis loop of the tri-valued memristor

Figure 2 shows the trajectory of the pinched hysteresis loop obtained by applying the voltage v = 2sin (2πt) to the tri-valued memristor. The starting point of the pinched hysteresis loop, as shown in Figure 2, is ①; at this time, the resistances of M_1 and M_2 are both R_{off}, so the state of the tri-valued memristor at point ① is the initial state R_H. Before reaching point ②, the voltage value across the memristor will not exceed the threshold voltage 1V, so the state of the memristor will not change. When point ② is reached, since the input voltage exceeds the threshold voltage 1V, it exceeds the threshold voltage V_{on} of M_1, and the resistance of M_1 is switched to R_{on}, while the resistance of M_2 remains unchanged. At this time, the resistance of the tri-valued memristor is set to R_M.

At point ③, the voltage across the memristor reaches the threshold voltage of 1.2 V; at this time, the threshold voltages V_{on} of M_1 and M_2 are exceeded at the same time, and the resistances of M_1 and M_2 are both switched to R_{on}; the resistance of the tri-valued memristor is set to R_L. On the trajectory before reaching point ④, the voltage continues to increase, but the resistance of the tri-valued memristor will remain at R_L. Along with the applied voltage decrease, from points ④ to ⑤, although the threshold voltage 1 V is crossed at this stage, the state of the memristor will not change because the resistance of the tri-valued memristor is R_L

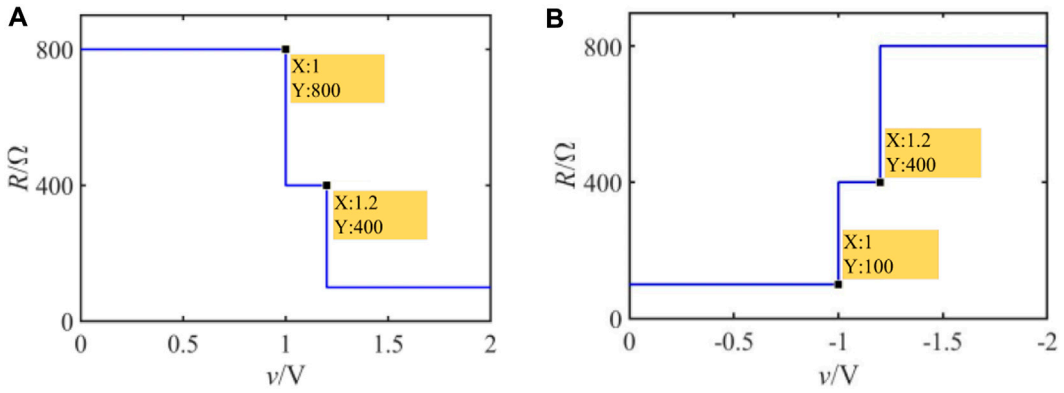


FIGURE 3
Relationship between the input voltage and the resistance of the tri-valued memristor; **(A)** the initial resistance is R_H, and the input voltage is $v = t$; **(B)** the initial resistance value is R_L, and the input voltage is $v = -t$.

during this period. When point ⑤ is reached, the voltage across the memristor reaches the threshold voltage −1 V, it exceeds the threshold voltage V_{off} of M_2, and the resistance of M_2 is switched to R_{off}, while the resistance of M_1 remains unchanged, making the tri-valued memristor set to R_M again. As the input voltage further decreases, reaching point ⑥, the voltage across the memristor exceeds the threshold voltage −1.2 V; at this time, the threshold voltages V_{off} of M_1 and M_2 are exceeded at the same time, and the resistances of M_1 and M_2 are both switched to R_{off}, making the tri-valued memristor set to R_H again. Then, the input voltage reaches peak point ⑦ and begins to increase, during which the resistance of the tri-valued memristor will not change until it returns to the initial point ① to start the next cycle.

Overall, the pinched hysteresis loop of the tri-valued memristor model shows the characteristics that the resistance decreases with the increase in the forward voltage, and the resistance increases with the increase in the negative voltage. It is worth noting that the change law of the resistance value of this tri-valued memristor model is consistent with the change law of the resistance value shown by the HP memristor. So this tri-valued memristor model can be used to research on the application of multivalued memristors manufactured in the method of an HP memristor.

TABLE 3 Truth table of the 3-line to 1-line ternary encoder.

X_0	X_1	X_2	Y
1	0	0	0
0	1	0	1
0	0	1	2

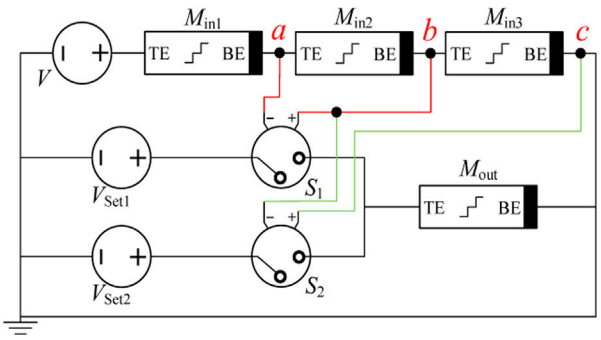


FIGURE 4
3-line–1-line ternary encoder circuit.

3 Ternary combination logic gates design

In this section, a series of ternary combinational logic circuits with memristor resistance as the logic state variable will be proposed. Specifically, it includes a ternary encoder, ternary decoder, ternary comparator, and ternary data selector, and the effectiveness of the designed ternary combinational logic gate will be verified by LTspice simulation.

3.1 Ternary encoder

In the traditional binary or multivalued encoder, the circuit structure of the encoder is composed of cascaded basic logic gates, and the circuit structure is relatively complex. In this paper, a ternary encoder circuit based on the tri-valued memristor is proposed, which does not require cascading basic ternary logic gates. The circuit consists of three input memristors, one output memristor, and a corresponding number of voltage-controlled switches, and can realize the function of converting three channels of binary signals into one channel of ternary signals. The truth table of the designed 3-line–1-line ternary encoder is shown in Table 3, where X_0, X_1, and X_2 are input signals, and Y is the output signal.

Figure 4 shows the circuit structure of a 3-line to 1-line ternary encoder based on tri-valued memristors. Among them, M_{in1}, M_{in2}, and M_{in3} are input memristors, M_{out} is the output memristor, and the initial resistance value of M_{out} is R_H. The magnitudes of the output voltages of the DC voltage sources V_{set1} and V_{set2} are 1.1 and

1.3 V, respectively, which are used to complete the operations of setting "1" and setting "2"to M_{out}. S_1 and S_2 are voltage-controlled switches, which are turned on when the applied control voltages V_{ba} and V_{cb} across M_{in2} and M_{in3} exceed their threshold voltages. After evaluation, the threshold voltages of S_1 and S_2 are set to satisfy that only one of these two switches is turned on or none of the switches is turned on (the output memristor maintains the initial value) under different inputs to realize the encoding function of the ternary encoder.

The operation of the encoder is driven by an excitation voltage source V, and its working process can be divided into two stages: the first stage is the initial stage where V outputs initial voltage V_{Init}, which is used to measure the initial resistance state of each memristor. The second stage is the running stage, and in this stage, V will output running voltage V_{Run} to complete the encoding operation. The initial voltage V_{Init} must meet two conditions to complete the operation of measuring the initial state of the memristor: the first condition is to ensure that the voltage division of each input memristor does not exceed the threshold voltages v_{th1} and v_{th2} of the memristor when V_{Init} is input. Otherwise, the resistance state of the input memristor will be changed; The second condition is that when V_{Init} is input, the divided voltages V_{ba} and V_{cb} on M_{in2} and M_{in3} would not exceed the threshold voltages of S_1 and S_2; otherwise, the resistance of the output memristor will change in the initial stage, which will cause the encoding result not correct under the running voltage V_{Run}. Here, the only demand on V_{Run} is to ensure that the voltage division of each input memristor does not exceed the threshold voltages v_{th1} and v_{th2} of the memristor when V_{Run} works as an input. Based on the aforementioned rules, the initial voltage V_{Init} and the operating voltage V_{Run} of the ternary encoder are determined as 1.5 and 1.2 V, respectively. The operation stage of the 3-line–1-line ternary encoder designed in this paper can be divided into the following three situations:

(1) When the input logic is "100," namely, M_{in1} = 400Ω, M_{in2} = 800Ω, and M_{in3} = 800Ω, according to the input voltage division calculation, $V_{ba} = V_{cb}$ = −0.48 V can be obtained. In the circuit, switches S_1 and S_2 need not be turned on at the same time to ensure the output logic "0," so the threshold voltages of switches S_1 and S_2 must be greater than −0.48 V.
(2) When the input logic is "010," namely, M_{in1} = 800Ω, M_{in2} = 400Ω, and M_{in3} = 800Ω, V_{ba} = −0.24V and V_{cb} = −0.48 V can be produced at this moment based on the input voltage division computation. To assure the output logic "1," switches S_1 must be switched on and S_2 need to turn off, so the threshold voltage of switches S_1 must be less than −0.24 V and S_2 must be more than −0.48 V.
(3) When the input logic is "001," namely, M_{in1} = 800Ω, M_{in2} = 800Ω, and M_{in3} = 400Ω, the corresponding output logic should be "2." In this case, the input voltage yields V_{ba} = −0.48V and V_{cb} = −0.24 V, so the switch S_1 need to be turned off and S_2 must be switched on, which force the threshold voltage of switches S_1 must be greater than −0.48 V and S_2 must be less than −0.24 V.

Through the aforementioned analysis, it can be obtained that the range of the threshold voltages of S_1 and S_2 should be between −0.48, and −0.24 V to complete the function of the encoding circuit, so −0.3 V is chosen as the threshold voltages of both S_1 and S_2 at the end.

The ternary encoder circuit is built using LTspice, and the simulation results are shown in Figure 5. It can be observed that when the inputs are "100," "010," and "001", the logic values of the output memristor are "0," "1," and "2," which is consistent with the truth table of the ternary encoder, which proves the rationality of the designed ternary encoder circuit.

3.2 Ternary decoder

Decoding is the reverse operation of encoding, and the function of the ternary decoder is to convert one ternary signal into three binary signals. Table 4 shows the truth table of the ternary decoder, where X represents the input of the ternary decoder, and Y_0, Y_1, and Y_2 represent the outputs of the ternary decoder. It is worth noting that the valid logic values output by Y_0, Y_1, and Y_2 can be either logic "1" or logic "2" according to actual needs, and in this paper, we take logic "1" as an example to design the ternary decoder.

Figure 6 shows the 1-line to 3-line decoder circuit based on tri-valued memristors and voltage-controlled switches. The resistance state of the memristor M_{in} is used to represent the logic value of the input variable X. The initial resistance values of M_{out1}, M_{out2}, and M_{out3} are all R_H, which represent the logic values of the output variables Y_0, Y_1, and Y_2, respectively. The voltage source V_{set1} outputs 1.1 V to set each of the output memristor as logic "1." The auxiliary resistor R is a key component in the circuit, and it helps realize more voltage division cases in the circuit. Without this resistor, the negative pole of M_{in} would be grounded directly so that the voltage source V will be applied directly to M_{in}. During the resistance state of M_{in} changing, the divided voltage on it will always be equal to the power voltage V, which means the three input conditions cannot be distinguished, resulting in the inability to complete the decoding function. The value of R needs to be between the high resistance value and the low resistance value of the tri-valued memristor; in this part, R is selected as 400Ω. Similar to the 3–1 encoder circuit, the ternary decoder work in two stages, which are the initial stage with V=V_{Init} = 0.5 V and running stage V = V_{Run} = 1.2 V, respectively.

In the decoder design, if only one voltage-controlled switch is connected to each output memristor in the circuit, there is no guarantee that only one branch of each output is selected through. For example, each output memristor is connected to only one voltage-controlled switch, whose control voltage is the voltage across M_{in}. In Case 1, when M_{in} = R_L, the switch with the smallest threshold voltage will be turned on. In Case 2, when M_{in} = R_M, the switch with the minimum and middle threshold voltages will be both turned on. In Case 3, when the resistance value of M_{in} is the maximum value, that is, M_{in} = R_H, all three switches will be turned on, which do not guarantee that only one switch is turned on at each case. Therefore, in this design, we connect two voltage-controlled switches to each output memristor to ensure that under different input conditions, only one output memristor is connected to the voltage source V_{set1} so that the corresponding output memristor can be set to R_M.

In Figure 6, the control voltage of switches S_1, S_3, and S_5 is the divided voltage V_{bc} on the resistor R. The control voltages of S_2, S_4,

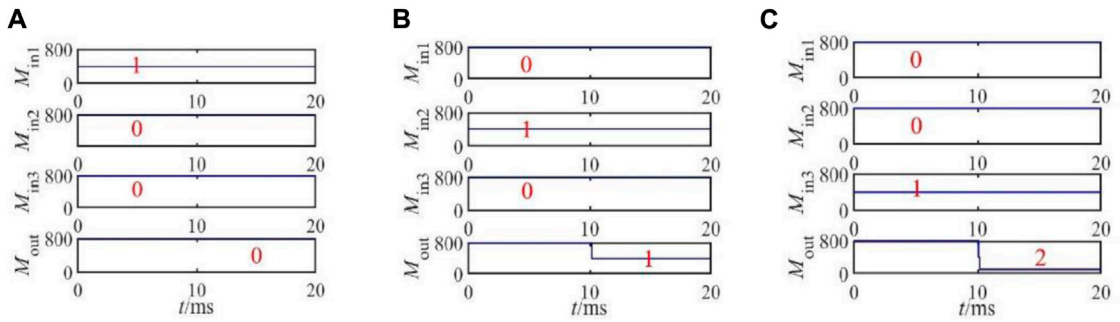


FIGURE 5
LTspice simulation results of the ternary encoder: **(A)** input as (1,0,0), **(B)** input as (0,1,0), and **(C)** input as (0,0,1).

TABLE 4 Truth table of the 1-line to 3-line ternary decoder.

X	Y_0	Y_1	Y_2
0	1	0	0
1	0	1	0
2	0	0	1

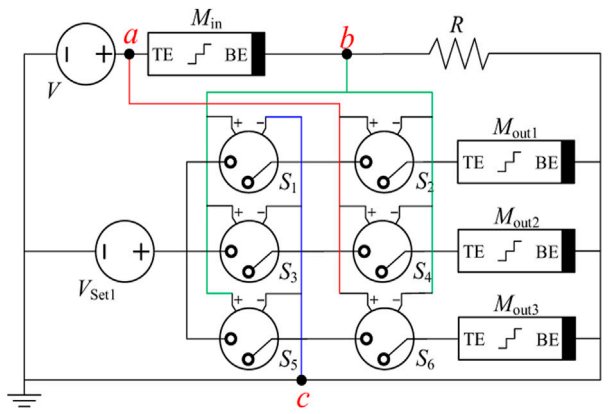


FIGURE 6
1-line to 3-line decoder circuit.

and S_6 are the divided voltage V_{ab} on the memristor M_{in}. In the ternary decoder circuit design process, an operating voltage is first selected; then, by analyzing the voltage division under different inputs, the threshold voltage of switches S_1–S_6 is set to meet the command above as 0.3, 0.7, 0.5, 0.5, 0.9, and 0.2 V, respectively.

According to the specific resistance state of M_{in}, the circuit operation stage of the ternary decoder can be summarized into the following three situations under the threshold voltage of switches S_1–S_6 set previously:

(1) When the input is logic "0," that is, M_{in} = 800Ω, at this time, V_{bc} = 0.4V and V_{ab} = 0.8V, all of the switches S_1, S_2, S_4, and S_6 are turned on, and S_3 and S_5 are turned off, so only the output memristor M_{out1} is set to 400Ω. The other two output memristors remain in their initial status. The logic gate output is "100."

(2) When the input is logic "1," M_{in} = 400Ω. V_{bc} = V_{ab} = 0.6 V. Switches S_1, S_3, S_4, and S_6 are all on, S_3 and S_5 are turned off, and only the output memristor M_{out2} can be set to 400Ω. So the output of the logic gate is "010."

(3) When the input is logic "2," that is, M_{in} = 100Ω, then V_{bc} = 0.96V and V_{ab} = 0.24 V. Switches S_1, S_3, S_5, and S_6 are all on, and S_2 and S_4 are turned off, while the output memristor M_{out3} is set to 400Ω; the other memristors will not change, so we obtain the outputs of the logic gate as "001."

Similarly, the ternary decoder is simulated and verified using SPICE, as shown in Figure 7. When the inputs are "0," "1," and "2", the corresponding outputs are "100," "010," and "001," respectively. The function of converting one ternary signal into three binary signals is realized, which confirms the effectiveness of the designed 1-line–3-line ternary decoder circuit.

3.3 Ternary comparator

In digital logic circuits, it is often necessary to compare the magnitude of two numbers, and the circuit that completes this logic function is called a numerical comparator. For a one-bit ternary comparator, its input is two ternary numbers, and the output is the result of the comparison of the two numbers. The truth table of the one-bit ternary comparator is shown in Table 5, where A and B are the two input ternary numbers, and L, E, and G represent the three output binary numbers. L, E and G are valid when output logic "1," representing $A < B$, $A = B$, and $A > B$, respectively.

As shown in Figure 8, the one-bit ternary comparator consists of two input memristors, three output memristors, two voltage sources, and six voltage-controlled switches. Input variables A and B are represented by the resistances of M_{in1} and M_{in2}, and the resistances of M_{out1}, M_{out2}, and M_{out3} are used to represent variables L, E, and G. The initial resistances of M_{out1}, M_{out2}, and M_{out3} are all R_H. Similar to the decoder circuit, each output memristor in the circuit is connected with two switches to ensure that only one output memristor is connected to the voltage source V_{set1} during the operation phase of the circuit. The two output voltages V_{Init} and V_{Run} of the voltage source V are 0.5V and 1 V, respectively, and V_{set1} outputs 1.1 V for the operation of "1."

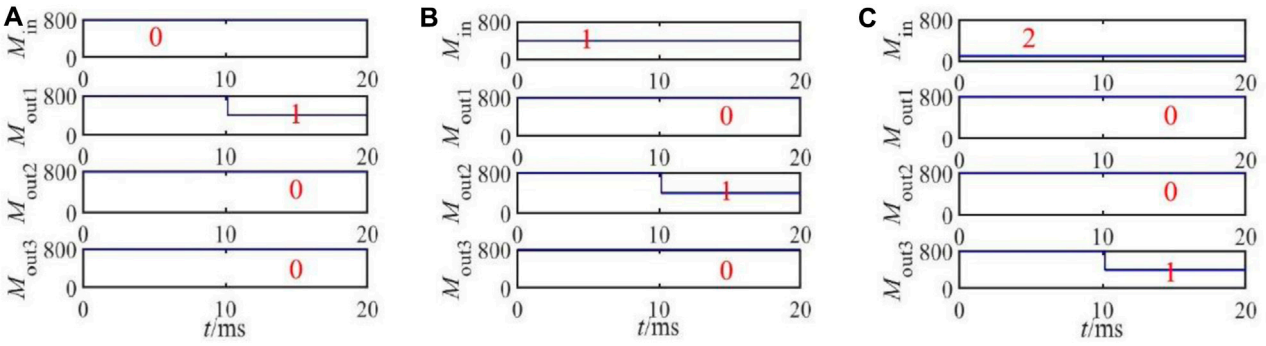


FIGURE 7
LTspice simulation results of the ternary decoder, **(A)** input "0," **(B)** input "1," and **(C)** input "2."

TABLE 5 Truth table of the one-bit ternary comparator.

A	B	L	E	G
0	0	0	1	0
0	1	1	0	0
0	2	1	0	0
1	0	0	0	1
1	1	0	1	0
1	2	1	0	0
2	0	0	0	1
2	1	0	0	1
2	2	0	1	0

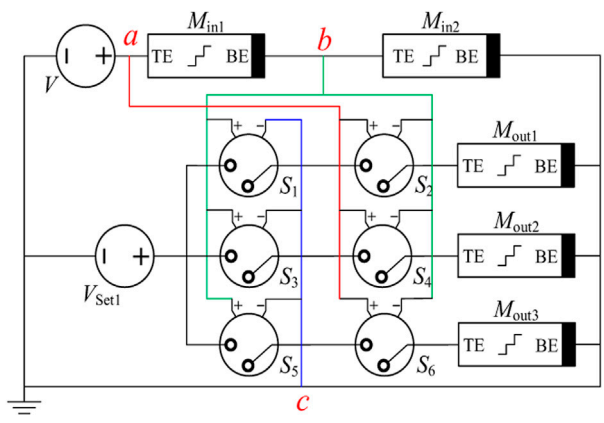


FIGURE 8
One-bit ternary comparator circuit.

In Figure 8, the control voltages of switches S_1, S_3, and S_5 are V_{bc} across M_{in2}, and V_{ab} cross M_{in1} is used to control S_2, S_4, and S_6. The threshold voltages of S_1–S_6 are 0.1, 0.6, 0.4, 0.4, 0.6, and 0.1 V, respectively. According to the specific logic states of M_{in1} and M_{in2}, the specific operation of the ternary comparator in the running phase can be summarized as follows:

(1) If A < B, there are three cases of input resistances of M_{in1} and M_{in2}, namely, (800Ω, 400Ω), (800Ω, 100Ω), and (400Ω, 100Ω). The divided voltages V_{ab} and V_{bc} of M_{in1} and M_{in2} are (0.67 V, 0.33 V), (0.89 V, 0.11 V), and (0.8 V, 0.2 V). In the aforementioned three cases, all the switches S_1, S_2, S_4, and S_6 are turned on, and S_3 and S_5 are turned off. Therefore, only the output memristor M_{out1} is set to 400Ω. The resistance of the other two output memristors remains unchanged, and the ternary comparator output is "100."

(2) If A = B, the input resistances of M_{in1} and M_{in2} are the same, that is, (800Ω, 800Ω), (400Ω, 400Ω), and (100Ω, 100Ω). If V_{ab} = V_{bc} = 0.5 V, at this time, switches S_1, S_3, S_4, and S_6 are all on, and S_2 and S_5 are turned off. Thus, only M_{out2} is set to 400Ω, and the corresponding logic gate output is "010."

(3) If A > B, there are also three cases of input resistance of M_{in1} and M_{in2}, namely, (400Ω, 800Ω), (100Ω, 800Ω), and (100Ω, 400Ω). V_{ab} and V_{bc} are (0.33 V, 0.67 V), (0.11 V, 0.89 V), and (0.2 V, 0.8 V). Switches S_1, S_3, S_5, and S_6 are all on, and S_2 and S_4 are turned off, which corresponds to the output memristor of M_{out3}, which is set to 400Ω; M_{out1} and M_{out2} remain R_H. At this point, the output of the logic gate is "001."

Figure 9 shows the LTspice simulation results of the ternary comparator. When the logic value of M_{in1} is less than that of M_{in2}, M_{out1}, which is corresponding to the variable "*L*," is set to logic "1." Similarly, when the logic value of M_{in1} is equal to that of M_{in2}, M_{out2} corresponding to the variable "*E*" will output "1." When the logic value represented by M_{in1} is greater than M_{in2}, the output memristor M_{out3}, which represents the variable *G*, outputs the effective resistance value "1." The simulation results are consistent with the truth table of the ternary comparator, which verifies the effectiveness of the designed one-bit ternary comparator circuit.

3.4 Ternary data selector

In the process of digital signal transmission, sometimes, it is necessary to select one from a group of data, and then, a logic

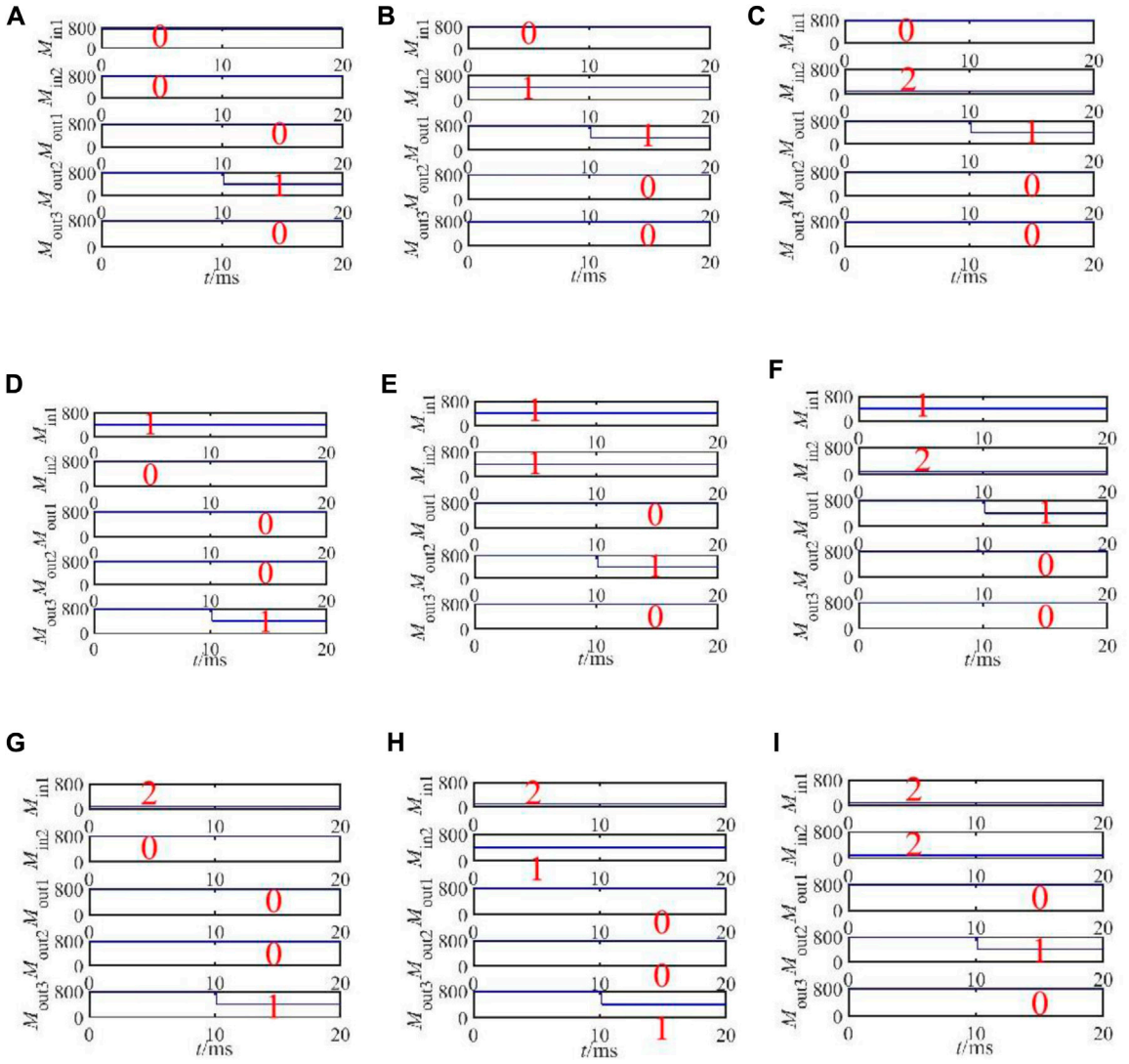


FIGURE 9
LTspice simulation results of the ternary comparator: **(A)** input as (0, 0), **(B)** input as (0, 1), **(C)** input as (0, 2), **(D)** input as (1, 0), **(E)** input as (1, 1), **(F)** input as (1, 2), **(G)** input as (2, 0), **(H)** input as (2, 1), and **(I)** input as (2, 2)

TABLE 6 Truth table of the 9-to-1 ternary data selector.

A	B	Y
0	0	D_0
0	1	D_1
0	2	D_2
1	0	D_3
1	1	D_4
1	2	D_5
2	0	D_6
2	1	D_7
2	2	D_8

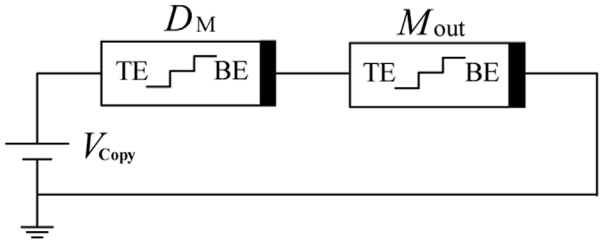


FIGURE 10
Ternary copy circuit.

circuit called data selector or multiplexer is used. The traditional binary data selector can select one data from four data through two address signals, while the ternary data selector based on the

TABLE 7 Specific voltage division of each memristors in the calculation stage of the ternary copy gate.

First stage				Second stage		
Data memristor		Output memristor		Data memristor	Output memristor	
Memristor (logical value)	Voltage (V)	Memristor (logical value)	Voltage (V)	Voltage (V)	Memristor (logical value)	Voltage (V)
800Ω(0)	0.8	800Ω(0)	0.8	No change in status		
400Ω(1)	0.53	800Ω(0)	1.07	0.8	400Ω(1)	0.8
100Ω(2)	0.18	800Ω(0)	1.42	0.8	100Ω(2)	0.8

tri-valued memristor can select one output from nine data through two address signals. The truth table of the 9-out-of-1 ternary data selector is shown in Table 6, where A and B are two address signals, and Y outputs the data selected by the address signal.

Before designing the ternary data selector circuit, a ternary copy gate needs to be introduced, whose function is to copy the information from the data memristor to the output memristor. The circuit structure is shown in Figure 10, which consists of an operating voltage source V, a data memristor D_M, and an output memristor M_{out}. Among them, V outputs the operating voltage V_{copy}, and the initial resistance of M_{out} is R_H. According to the different input logic, it can be divided into the following three cases:

(1) If $D_M = 800\Omega$ and the logic is "0," the voltage of M_{out} should be less than 1 V to ensure M_{out} remain its initial resistance, which demand the divided voltage of M_{out} meets $(800/1600)*V_{copy} < 1V$, that is, $V_{copy} < 2$ V.
(2) If $D_M = 400\Omega$ and the logic is "1," to ensure M_{out} change from R_H to R_M, the divided voltage of M_{out} should be between 1 V and 1.2 V, so the divided voltage of M_{out} should satisfy the condition of '1 V< $(800/1200)*V_{copy}$<1.2 V', that is, 1.5 V ≤ V_{copy}<1.8 V.
(3) If $D_M = 100\Omega$ and the logic is "2," M_{out} need to change from R_H to R_L, which needs to meet the condition of $V_{copy} \geq 1.35$ V.

Combining the three aforementioned situations, the operating voltage of the ternary copy gate circuit must satisfy 1.5 V ≤ V_{copy}<1.8 V. So we choose 1.6 V as the operating voltage of the copy gate, and the specific voltage division between the data memristor and the output memristor is shown in Table 7.

The circuit structure of the 9-out-of-1 ternary data selector based on tri-valued memristors is shown in Figure 11, where the input memristors M_{in1} and M_{in2} store two address signals A and B, and the data memristors D_0–D_8 store, respectively, nine channels of known ternary data. M_{out} outputs the data D_i (i = 0, 1, 2, . . ., 8) selected by the address signals, and all of the initial resistance values of M_{out} are R_H. In this design, D_i forms a ternary copy gate with the output memristor M_{out} and the voltage source V_{copy}, and its function is to copy the data on D_i to M_{out}. Switches S_1–S_{18} are used to control the access conditions of the memristors D_0–D_8, ensuring that only one data memristor D_i is connected to the circuit during the circuit operation stage. It is worth mentioning that the auxiliary resistor R is also added to the data selector circuit, and its function is to make the circuit have more voltage divider cases. If there is no auxiliary resistor, when the values of M_{in1} and M_{in2} are same, there will be three different cases ($M_{in1} = M_{in2} = 800\Omega$, $M_{in1} = M_{in2} = 400\Omega$, and $M_{in1} = M_{in2} = 100\Omega$), but the divided voltages on M_{in1} and M_{in2} in these three cases will always be the same, which will cause the switch groups that work for these cases to be the same. However, with the help of an auxiliary resistor R, the voltage division in these three cases will easily be distinguished, and different switch groups can be turned on according to different cases. To meet this demand, R is calculated as 500Ω, and the other parameters are V_{copy} = 1.6 V, V_{Init} = 0.5 V, and V_{Run} = 1.2 V.

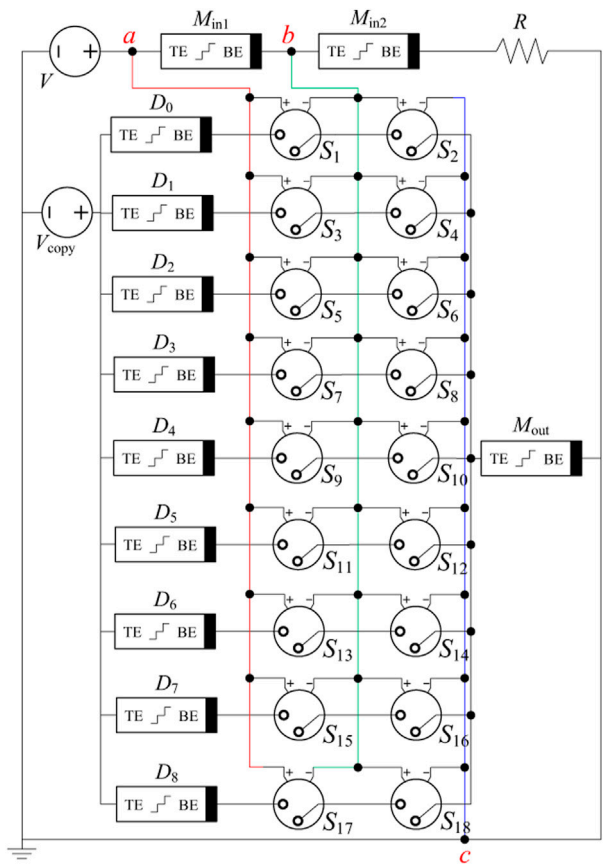


FIGURE 11
9-out-of-1 ternary data selector circuit.

TABLE 8 Conduction of the 9-to-1 ternary data selector under different inputs.

Input resistance (logical value)		Control voltage (V)		Conductive switch group	M_{out}
M_{in1}	M_{in2}	V_{ab}	V_{bc}		
800Ω(0)	800Ω(0)	0.457	0.743	(S1,S2)	D_0
800Ω(0)	400Ω(1)	0.565	0.635	(S3,S4)	D_1
800Ω(0)	100Ω(2)	0.686	0.514	(S5,S6)	D_2
400Ω(1)	800Ω(0)	0.282	0.918	(S7,S8)	D_3
400Ω(1)	400Ω(1)	0.369	0.831	(S9,S10)	D_4
400Ω(1)	100Ω(2)	0.48	0.72	(S11,S12)	D_5
100Ω(2)	800Ω(0)	0.086	1.114	(S13,S14)	D_6
100Ω(2)	400Ω(1)	0.12	1.08	(S15,S16)	D_7
100Ω(2)	100Ω(2)	0.171	1.029	(S17,S18)	D_8

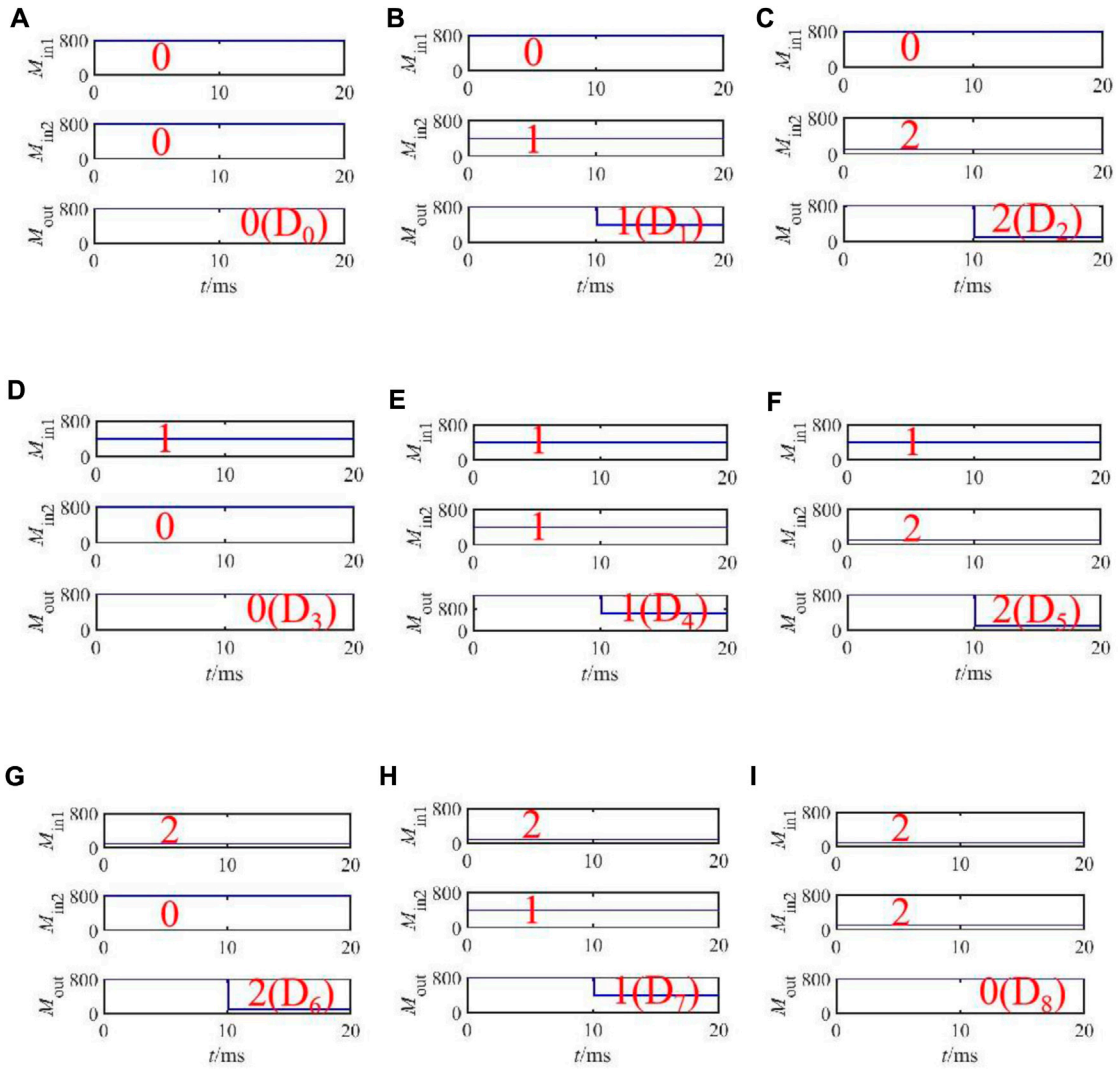


FIGURE 12
LTspice simulation result of the ternary data selector: **(A)** input as (0, 0), **(B)** input as (0, 1), **(C)** input as (0, 2), **(D)** input as (1, 0), **(E)** input as (1, 1), **(F)** input as (1, 2), **(G)** input as (2, 0), **(H)** input as (2, 1), and **(I)** input as (2, 2).

For convenience, switches S_1–S_{18} are divided into nine groups. Each group is called a switch group, which is expressed as (S_1, S_2) ~ (S_{17}, S_{18}), and each switch group is turned on only when two switches in the switch group are turned on at the same time. The threshold voltages of S_1–S_{18} are set to 0.4, 0.73, 0.5, 0.6,0.6, 0.5, 0.2, 0.9, 0.3, 0.8, 0.47, 0.7, 0.05, 1.1, 0.1, 1.05, 0.15, and 1 V. Table 8 lists the situation of conductive switch groups and the output of M_{out} under different inputs of the 9-out-of-1 ternary data selector. According to the specific logic states of M_{in1} and M_{in2}, the specific operation of the ternary data selector in the running phase can be summarized as follows. In this paper, the input logic "00" and "01" are taken as examples for detailed analysis. The analysis process of other inputs is the same, and the description will not be repeated here.

(1) The input logic is "00," that is, M_{in1} = 800Ω and M_{in2} = 800Ω. V_{ab} = 0.457V and V_{bc} = 0.743 V can be obtained by voltage division calculation. At this time, only S_1 and S_2 will be closed at the same time among the nine switch groups. Then, memristors D_0 and M_{out} are connected in series for the copy operation, and the logic gate outputs the data stored in D_0.
(2) The input logic is "01," that is, M_{in1} = 800Ω, M_{in2} = 400Ω, V_{ab} = 0.565V, and V_{bc} = 0.635 V. Only S_3 and S_4 are turned on at the same time among the nine switch groups. So the memristor pair D_1 and M_{out} will be connected in series for the copy operation, and the logic gate outputs the data stored in D_1.

The 9-out-of-1 ternary data selector is simulated using LTspice software, in which the data memristors D_0–D_8 store nine channels of ternary signals, and its resistance value and corresponding logic value are D_0 = 800Ω ("0"), D_1 = 400Ω ("1"), D_2 = 100Ω ("2"), D_3 = 800Ω ("0"), D_4 = 400Ω ("1"), D_5 = 100Ω ("2"), D_6 = 100Ω ("2"), D_7 = 400Ω ("1"), and D_8 = 800Ω ("0"). The simulation results under different combinations of address signals are shown in Figure 12, which shows the data selection results of the 9-out-of-1 ternary data selector. The resistance values of M_{in1} and M_{in2} are the input address signals, and the resistance value of M_{out} is the output signal. When the inputs are "00," "01," "02," "10," "11," "12," "20," "21," and "22," the ternary signals stored in D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7, and D_8 are outputs, which are consistent with the output of the ternary data selector in Table 6, and verify the effectiveness of the designed ternary data selector circuit.

4 Conclusion

This paper presents a design method of the ternary combinational logic gate circuit based on a tri-valued memristor and realizes the ternary encoder, ternary decoder, ternary comparator, and ternary data selector with the resistance of the tri-valued memristor as the logic state variable. First, a voltage threshold tri-valued memristor is introduced, and the threshold characteristics of the tri-valued memristor are analyzed in detail. Second, using the tri-valued memristor, a series of complex combinational logic circuits with the resistance of the memristor as the logic state variable are designed, respectively, and the implementation principle of each gate is analyzed in detail. Finally, the effectiveness of the designed combinational logic gate circuits is verified by LTspice circuit simulation. Compared with the existing binary or ternary combinational logic gate circuits, the ternary combinational logic gate circuit based on the tri-valued memristor proposed in this paper does not need to be realized by cascading basic logic gates and can be realized only by a small number of tri-valued memristors and voltage-controlled switches. In addition, the logic gate designed by this method can not only perform logic operations but also store logic values.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

X-JL: validation, writing–review and editing, methodology, and writing–review and editing. X-YW: writing–review and editing, data curation, conceptualization, funding acquisition, investigation, methodology, project administration, and supervision. PL: writing–review and editing, data curation, and investigation. HI: visualization, and writing–review and editing. Z-QC: visualization, and writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (Grant No. 61871429), the Natural Science Foundation of Zhejiang Province, China (Grant No. LY18F010012), and the Project of Ministry of Science and Technology of China (Grant No. D20011).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Hamedani SG, Moaiyeri MH. Impacts of process and temperature variations on the crosstalk effects in sub-10 nm multilayer graphene nanoribbon interconnects. *IEEE Trans Device Mater Reliability* (2019) 19(4):630–41. doi:10.1109/TDMR.2019.2937789

2. Khezeli MR, Moaiyeri MH, Jalali A. Analysis of crosstalk effects for multiwalled carbon nanotube bundle interconnects in ternary logic and comparison with Cu interconnects. *IEEE Trans Nanotechnology* (2016) 16(1):107–17. doi:10.1109/TNANO.2016.2633460

3. Hurst . Multiple-valued logic—Its status and its future. *IEEE Trans Comput* (1984) 100(12):1160–79. doi:10.1109/TC.1984.1676392

4. Glusker M, Hogan DM, Vass P. The ternary calculating machine of Thomas Fowler. *IEEE Ann Hist Comput* (2005) 27(3):4–22. doi:10.1109/MAHC.2005.49

5. Talati N, Gupta S, Mane P, Kvatinsky S. Logic design within memristive memories using memristor-aided loGIC (MAGIC). *IEEE Trans Nanotechnology* (2016) 15(4): 635–50. doi:10.1109/TNANO.2016.2570248

6. Thangkhiew PL, Datta K. Scalable in-memory mapping of Boolean functions in memristive crossbar array using simulated annealing. *J Syst Architecture* (2018) 89: 49–59. doi:10.1016/j.sysarc.2018.07.002

7. Thangkhiew PL, Gharpinde R, Datta K. Efficient mapping of Boolean functions to memristor crossbar using MAGIC NOR gates. *IEEE Trans Circuits Syst Regular Pap* (2018) 65(8):2466–76. doi:10.1109/TCSI.2018.2792474

8. Wald N, Kvatinsky S. Understanding the influence of device, circuit and environmental variations on real processing in memristive memory using Memristor Aided Logic. *Microelectronics J* (2019) 86:22–33. doi:10.1016/j.mejo.2019.02.013

9. Min F, Xue L. Routes toward chaos in a memristor-based Shinriki circuit. *Chaos: Interdiscip J Nonlinear Sci* (2023) 33(2):023122. doi:10.1063/5.0126900

10. Khalid M, Singh J. Memristor based unbalanced ternary logic gates. *Analog Integrated Circuits Signal Process.* (2016) 87:399–406. doi:10.1007/s10470-016-0733-1

11. Zhang H, Zhang Z, Gao M, Luo L, Duan S, Dong Z, et al. Implementation of unbalanced ternary logic gates with the combination of spintronic memristor and CMOS. *Electronics* (2020) 9(4):542. doi:10.3390/electronics9040542

12. Wang XY, Zhou PF, Eshraghian JK, Lin CY, Iu HHC, Chang TC, et al. High-density memristor-CMOS ternary logic family. *IEEE Trans Circuits Syst Regular Pap* (2020) 68(1):264–74. doi:10.1109/TCSI.2020.3027693

13. Soliman NS, Fouda ME, Radwan AG. Memristor-CNTFET based ternary logic gates. *Microelectronics J* (2018) 72:74–85. doi:10.1016/j.mejo.2017.12.008

14. Mohammed MU, Vijjapuram R, Chowdhury MH. "Novel CNTFET and memristor based unbalanced ternary logic gate", In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS) (2018). IEEE, 1106–9. doi:10.1109/MWSCAS.2018.8623845

15. Soliman N, Fouda ME, Alhurbi AG, Said LA, Madian AH, Radwan AG. Ternary functions design using memristive threshold logic. *Ieee Access* (2019) 7:48371–81. doi:10.1109/ACCESS.2019.2909500

16. Chen QL, Liu G, Tang MH, Chen XH, Zhang YJ, Zheng XJ, et al. A univariate ternary logic and three-valued multiplier implemented in a nano-columnar crystalline zinc oxide memristor. *RSC Adv* (2019) 9(42):24595–602. doi:10.1039/c9ra04119b

17. Chua L. Everything you wish to know about memristors but are afraid to ask. *Handbook of Memristor Networks* (2019) 89–157. doi:10.1007/978-3-319-76375-0_3

18. Wang X, Zhang X, Gao M. A novel voltage-controlled tri-valued memristor and its application in chaotic system. *Complexity* (2020) 2020:1–8. doi:10.1155/2020/6949703

19. Li P, Wang X, Zhang X, Eshraghian JK, Lu HHC. Spice modelling of a tri-state memristor and analysis of its series and parallel characteristics. *IET Circuits, Devices Syst* (2022) 16(1):81–91. doi:10.1049/cds2.12086

20. Wang X, Li P, Jin C, Dong Z, Iu HH. General modeling method of threshold-type multivalued memristor and its application in digital logic circuits. *Int J Bifurcation Chaos* (2021) 31(16):2150248. doi:10.1142/S0218127421502485

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 18 October 2023
DOI 10.3389/fphy.2023.1290880

OPEN ACCESS

EDITED BY
Hairong Lin,
Hunan University, China

REVIEWED BY
Fuhong Min,
Nanjing Normal University, China
Wei Yao,
Changsha University of Science and Technology, China

*CORRESPONDENCE
Yuanpeng He,
he.yuanpeng@outlook.sg

RECEIVED 08 September 2023
ACCEPTED 09 October 2023
PUBLISHED 18 October 2023

CITATION
Zhou X and He Y (2023), Dynamic displacement estimation of structures using one-dimensional convolutional neural network.
Front. Phys. 11:1290880.
doi: 10.3389/fphy.2023.1290880

Dynamic displacement estimation of structures using one-dimensional convolutional neural network

Xin Zhou[1] and Yuanpeng He[2]*

[1]School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai, China, [2]Faculty of Geosciences and Ground Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China

For large infrastructures, dynamic displacement measurement in structures is an essential topic. However, limitations imposed by the installation location of the displacement sensor can lead to measurement difficulties. Accelerometers are characterized by easy installation, good stability and high sensitivity. For this regard, this paper proposes a structural dynamic displacement estimation method based on a one-dimensional convolutional neural network and acceleration data. It models the complex relationship between acceleration signals and dynamic displacement information. In order to verify the reliability of the proposed method, a finite element-based frame structure was created. Accelerations and displacements were collected for each node of the frame model under seismic response. Then, a dynamic displacement estimation dataset is constructed using the acceleration time series signal as features and the displacement signal at a certain moment as target. In addition, a typical neural network was used for a comparative study. The results indicated that the error of the neural network model in the dynamic displacement estimation task was 9.52 times higher than that of the one-dimensional convolutional neural network model. Meanwhile, the proposed modelling scheme has stronger noise immunity. In order to validate the utility of the proposed method, data from a real frame structure was collected. The test results showed that the proposed method has a mean square error of only 5.097 in the real dynamic displacement estimation task, which meets the engineering needs. Afterwards, the outputs of each layer in the dynamic displacement estimation model are visualized to emphasize the displacement calculation process of the convolutional neural network.

KEYWORDS

convolutional neural network, displacement estimation, acceleration, visualization, portable measurement

1 Introduction

With the development of structural health monitoring, the safety of some structures, such as high-rise buildings [1, 2], bridges [3, 4], and rapid transit [5, 6], has gradually attracted public attention. These structures may be affected by natural disasters such as typhoons and earthquakes. These natural disasters could cause structural damage and even lead to major accidents. Therefore, these structures are usually equipped with structural health monitoring systems and a large number of sensors [7–10] are placed to monitor the

safety of the structure. The most common sensors used in structural safety monitoring are accelerometers, fiber-optic gratings, strain gauges, displacement gauges, and so on. However, structural displacement monitoring has been a challenge in the monitoring field. Due to the limitation of structural space, it is sometimes difficult to find suitable locations for sensor installation, such as dynamic displacement detection in bridges. Even if there is enough space in the structure to install sensors, only relative displacement can be measured. Acceleration is easier to monitor than displacement. Furthermore, acceleration sensors can be connected directly to the test point. In addition, acceleration sensors are small and convenient and do not require a large installation space, so acceleration monitoring is easy to implement in engineering applications.

Despite the many difficulties in displacement monitoring, many innovative monitoring methods based on certain sensors have been proposed, such as lasers [11, 12], cameras [13, 14], radar [15], and global positioning systems [16]. These methods have been widely used in real structures, and even some sophisticated measurement devices such as laser displacement sensors, total stations, and millimeter wave radars have emerged. However, these displacement measurement devices still require a large installation space. They can only measure the relative displacement of the structure, and the actual displacement of the structure is difficult to measure. For bridge deflection measurement, it is impossible to build scaffolding on both sides of the bridge to install the sensors. Therefore, the installation space is a key issue that restricts the displacement measurement. In addition, there are many indirect monitoring methods based on acceleration data. Theoretically, the displacement signal can be obtained by double integration of the acceleration signal. The monitoring of acceleration signals is very easy to achieve, so some dynamic displacement measurement methods based on acceleration signals have been proposed. However, these methods may result in a continuous error trend in the displacement signal. Various algorithms for eliminating the error trend [17–20] have been extensively studied. The methods for removing the error trend are mainly classified into three types: time-domain integration, filtering, and frequency-domain integration. These integration methods are fixed signal processing methods that are not able to adapt to changes in the environment and uncertainty in the data. In complex and dynamic environments, these methods may not be able to adapt and process data efficiently. They are mostly used for linear signal processing and have limited ability to deal with nonlinear problems. However, many phenomena in engineering are non-linear issues [21]. Machine learning methods are better able to cope with nonlinear problems and improve performance by learning the nonlinear relationships of the data. Compared to machine learning, they have some limitations in terms of data dependency, feature design, adaptivity and non-linear problem handling. In addition, some neural networks are devoted to address multimodal functional synchronization [22, 23] and communication security [24]. Machine learning methods are more flexible in dealing with different types of data and can automatically learn features and patterns from data, making them more applicable in dealing with complex and uncertainty-prone problems.

In recent years, Convolutional Neural Network (CNN) has made great achievements in the field of object recognition. It is widely used in various fields, such as image recognition [25–28], medical diagnosis [29, 30], traffic safety [31, 32], crack detection [33, 34], pedestrian identification [35], and bolt loosening monitoring [36, 37]. These research results show that convolutional neural networks can accurately model numerous complex systems relying on big data. Li et al. accurately identified concrete surface cracks using a semantic segmentation algorithm based on convolutional neural networks [38, 39]. According to the semantic recognition results, they extracted the crack parameters and analyzed the fractal characteristics of the surface cracks from different specimens using image processing techniques. Zhang et al. used a convolutional neural network to process the time-frequency features of the seismic response, which were subsequently input into a dynamic network to complete the signal classification [40]. The convolutional neural network can deeply analyze the two-dimensional time-frequency features, and the dynamic network further improves the efficiency of signal processing [41]. All of these monitoring methods are based on two-dimensional convolutional neural networks, but one-dimensional convolutional neural networks [42, 43] also have great advantages in data processing. In recent years, convolutional neural networks have achieved remarkable results in the field of image processing. Consequently, the integration of CNNs and machine vision has become widely prevalent. However, when it comes to extracting localized features in image processing, the commonly employed approach is the utilization of two-dimensional CNNs. One-dimensional convolutional neural networks are commonly used to process one-dimensional signals, such as acceleration signals. Two-dimensional convolutional neural networks cannot directly process one-dimensional signals. Of course, one-dimensional signals can be converted into two-dimensional features, such as time-frequency features for processing by two-dimensional convolutional neural networks. However, one-dimensional convolutional neural networks can directly extract features from one-dimensional signals without additional conversion steps to achieve good recognition results. In addition, convolutional neural networks have the ability of autonomous learning. There is a simple integral relationship between acceleration and displacement at the same point. Although there is a trend term, the convolutional neural network has a strong learning ability to learn how to remove the trend term. In addition, the convolutional neural network has the ability to extract data features and optimize them without human intervention. Convolutional neural networks are very effective in processing data.

In this paper, a new method for estimating dynamic displacements of structures using one-dimensional convolutional neural networks and acceleration is presented. The method is used to estimate the dynamic displacement of a three-layer finite element model and a three-layer steel frame. A typical neural network algorithm provides a reference for the proposed CNN method. Section 2 describes the neural network and convolutional neural network used in this paper. A finite element model is designed in Section 3. Under the effect of Wenchuan earthquake wave, the acceleration signal and displacement signal of each node of the model are collected to form a dataset. The data set is divided into training set, validation set and test set. The training and validation sets are fed into the neural network and the proposed convolutional neural network. To validate the noise resistance of the proposed method, four noise levels (i.e., 10%, 20%, 30%, and 40%) are added to the acceleration signals to examine the robustness of the CNN to noisy data. These data were directly fed into the training model,

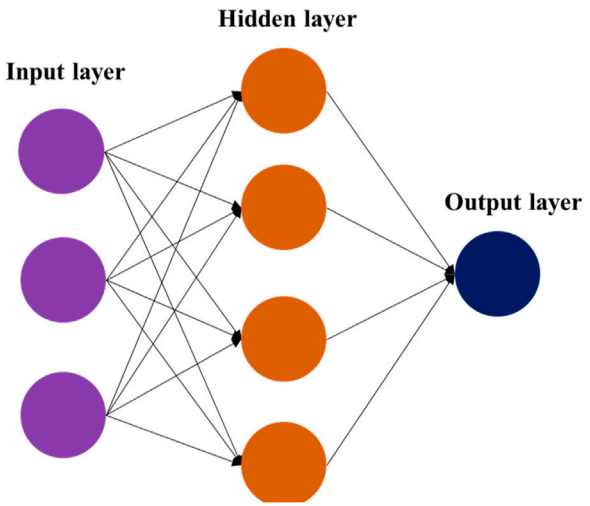


FIGURE 1
Neural network diagram.

which was trained by the noise-free dataset. The results show that the convolutional neural network is robust to noise. In Section 4, a three-layer steel frame is constructed and the acceleration signals and displacement signals of the nodes of the frame are collected by acceleration sensors and displacement sensors. The training and validation sets are fed into the proposed convolutional neural network. The results show that the mean square error (MSE) of the displacement estimation is 5.097, which can meet the engineering needs. Subsequently, the output of each layer is visualized to understand how the convolutional neural network processes the data. Section 5 discusses the article. Section 6 summarizes the article.

2 Methodology

This paper presents a dynamic displacement estimation method based on a one-dimensional convolutional neural network and acceleration signals. The method uses acceleration signals and an estimation model trained by a convolutional neural network to estimate the dynamic displacement of the structure. The estimation results of the neural network method can be used as a reference for the proposed convolutional neural network method. The following section describes the neural network and convolutional neural network in detail. In this paper, a workstation is used to train the model and computational frameworks such as TENSORFLOW and KERAS are applied.

2.1 Neural network

Neural network algorithms are generated by modelling the working of human neurons. A neural network generally consists of an input layer, a hidden layer, and an output layer, as shown in Figure 1. Sometimes, a neural network may have more than one hidden layer. In a fully connected neural network, neurons between two neighboring layers are connected exactly in pairs. The weights refer to the strength or amplitude of the connection between two neurons. During training, these connection weights are updated until the global error of the network approaches a minimum. In the parameter update calculation, this is typically done using a gradient descent algorithm. The key to the gradient descent algorithm is the calculation of the gradient, which tells us how to update the parameters to minimize the loss function. Gradient descent algorithms generally include Batch Gradient Descent, Stochastic Gradient Descent and Mini-Batch Gradient Descent [44]. They differ in the number of samples used each time the parameters are updated. Gradient descent algorithms have the advantage of being simple to implement and can be used to optimize a variety of loss functions.

The neural network used in this paper consists of one input layer, two hidden layers and one output layer, as shown in Tabel 1. The number of neurons in the input layer is 50, the number of neurons in the first hidden layer is 30, the number of neurons in the second hidden layer is 20, and the number of neurons in the output layer is 1. In general, the raw data usually has highly dense features, and the complex features can be transformed into sparse features. This can enhance the robustness of the features. Therefore, introducing an activation function in the neural network can improve the data sparsity. However, a large proportion of sparsity can destroy the characteristics of the data and affect the learning effect of the neural network. The sparsity ratio of human brain is 95%. Due to the property of Rectified Linear Unit (ReLU) (negative output of x is 0), Rectified Linear Unit (ReLU) can be generated for alternate networks. ReLU is shown in Figure 2. ReLU is a nonlinear activation function that helps the neural network model to learn nonlinear relationships. This is important for solving complex problems and fitting nonlinear data. In addition, the output of ReLU is 0 when the input value is less than 0. This means that there will not be any negative signals passed to the next layer of neurons. This sparse activation can help the network learn more robust feature representations and reduce redundancy between features [45]. Compared to other activation functions such as sigmoid and tanh which have small gradients over positive intervals, ReLU has constant gradients over positive intervals, reducing the problem of gradient vanishing and facilitating the training of the network.

The loss function is mainly used to measure the difference between the predicted value and the actual value. When the predicted value is closer to the actual value, the value of the loss function is smaller. When the predicted value is closer to the actual value, the value of the loss function is smaller. In neural networks, the loss function is the mean square error (MSE), which is often used in regression problems. The MSE can be calculated as the square of the difference between the actual value and the predicted value. The formula is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(actual_i - predicted_i\right)^2 \tag{1}$$

where n is the number of samples; *actual* is the actual value; *predicted* is the predicted value; i is the i-th sample. In order to comprehensively assess the impact of noise on the proposed method,

TABLE 1 Architecture of the neural network.

Layer	Number of neurons	Activation function	Number of trainable parameters
Input	50	ReLU	2550
Hidden layer#1	30	ReLU	1530
Hidden layer#2	20	ReLU	620
Output	1	ReLU	21

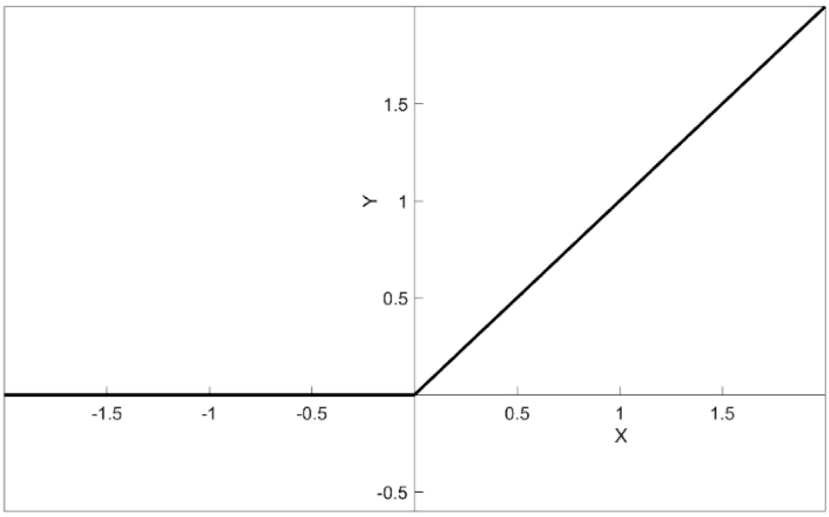


FIGURE 2
Activation function.

three evaluation metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R^2 are also calculated.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|actual_i - predicted_i\right| \tag{2}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{actual_i - predicted_i}{actual_i}\right| \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(actual_i - predicted_i\right)^2}{\sum_{i=1}^{n}\left(actual_i - mean\right)^2} \tag{4}$$

where *mean* represents the mean value of the real sample.

2.2 Convolutional neural network

The major difference between convolutional neural networks and ordinary neural networks is the convolution operation. To illustrate the advantages of convolutional neural networks, the proposed convolutional neural network uses the same activation function and loss function as the fully connected neural network. The architecture of the convolutional neural network is different from the fully connected neural network. Convolutional neural network contains convolutional layer, maximum pooling layer, flat layer and dense (fully connected) layer. Each neuron in a neural network is connected to all the previous layers of neurons, which leads to a large increase in the number of parameters. Whereas in CNN only the inputs that are within the range of the convolutional kernel will be selected for connection, which reduces the number of parameters and increases the computational efficiency. The detailed parameters of the convolutional neural network are shown in Table 2.

2.2.1 Convolutional layer

The convolutional layer is an important part of a convolutional neural network. The convolutional kernel extracts feature from the input data. CNNs use the convolutional kernel to perform sliding window operations on the input data, reducing the number of parameters in the network by means of parameter sharing. This makes CNNs more efficient in processing data with spatial and temporal relationships and allows spatially localized features to be extracted [46]. The convolution operation (⊗) starts from the top left corner of the input data. The parameters of the convolution kernel are multiplied with the parameters of the overlay region. The product values are added together as the output of the convolution operation. The convolution kernel is then shifted one element to the right or down (stride is 1) for the next convolution. The initial parameters of the convolution kernel are randomly generated. During the training process, the parameters of

TABLE 2 The parameters of the convolutional neural network.

Layer	Input shape	Output shape	Kernel size	Kernel number
Convolution 1-D	(50,1)	(46,6)	5	6
Convolution 1-D	(46,6)	(42,12)	5	12
Convolution 1-D	(42,12)	(38,64)	5	64
Convolution 1-D	(38,64)	(34,128)	5	128
Max Pooling 1-D	(34,128)	(17,128)	2	128
Flatten	(17,128)	(2176)	None	None
Dense	2176	(1)	None	None

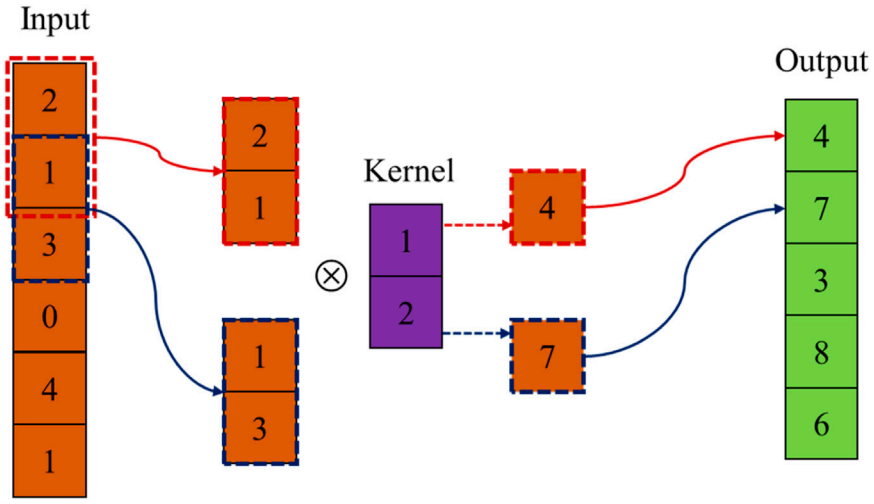


FIGURE 3
Example of convolutional operation.

the convolution kernel are updated until the end of training. An example of a convolution operation is shown in Figure 3.

2.2.2 Pooling layer

The pooling layer, also known as the down sampling layer, is mainly used to reduce the size of the feature data. CNNs use a pooling layer to reduce the spatial size of the feature map, to extract the positional information of the features and at the same time to preserve the most salient features in the feature map. Similar to convolutional layers, pooling layers allow for parameter sharing. This means that pooling operations performed in a local region use the same parameters, which can effectively reduce the number of parameters in the network and improve the computational efficiency of the model. In addition, like convolutional layers, pooling layers have a kernel. However, the pooling kernel does not contain any parameters. The most common pooling operation is maximum pooling. By selecting the maximum value within a local region, maximum pooling captures the most salient features. It is very effective in highlighting the most important and active features in the data [47]. An example of the max pooling operation is shown in Figure 4. For mean pooling, the input features can be smoothed by averaging them to reduce noise and redundant information.

2.2.3 Flatten layer

In this paper, the input data to the network is one-dimensional data. One convolutional kernel in the convolutional layer is convolved with the input data to generate one-dimensional data and multiple convolutional kernels are convolved with the input data to generate multidimensional data. The input data of the dense (fully connected) layer must be one-dimensional data. Therefore, the multidimensional data is transformed into one-dimensional data by the Flatten layer. The Flatten layer transforms the multidimensional input data into one-dimensional vectors, allowing the subsequent fully-connected layer to process the entire input. It is useful for processing multidimensional data, and after converting the input data into a one-dimensional form, the common neural network architecture can be used in the fully connected layer.

3 Numerical simulation

3.1 Dataset generation

The finite element model was created by ABAQUS. The model is a three-storey frame structure with a storey height of 4,500 mm and

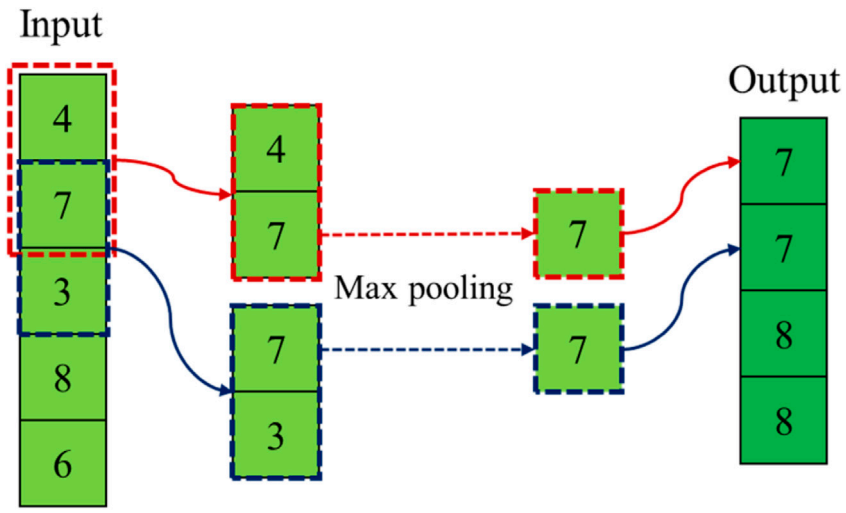


FIGURE 4
Example of pooling operation.

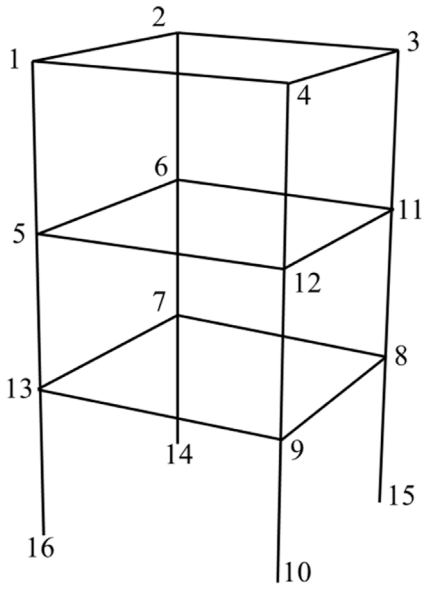


FIGURE 5
The finite element model and node number.

a span of 7,200 mm. The beams and columns are made of Q235 steed. The seismic intensity is 8°. The finite element model and node numbers are shown in Figure 5.

The Wenchuan seismic waves are input into the finite element model and then the acceleration and displacement signals of twelve nodes (1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12 and 13) are collected. The acceleration signals with a duration of 0.5 s at each node will be used as inputs to the neural network and convolutional neural network. The sampling frequency is 100 Hz, so the input data is a 50 × 1 vector. The output of the network is the displacement signal (1 × 1) at the end of the corresponding 0.5 s interval. The acceleration and

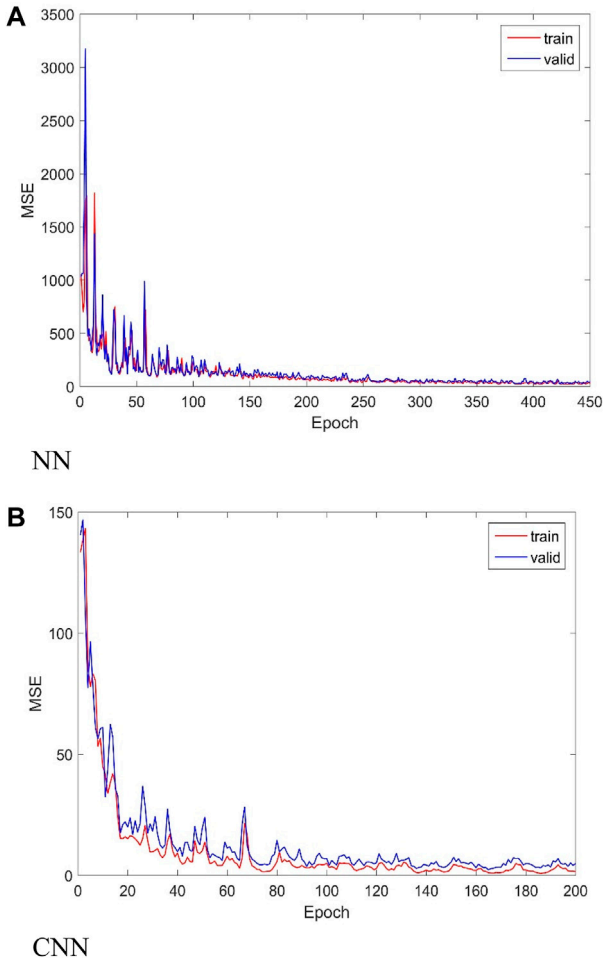


FIGURE 6
The MSE curves of the two methods. **(A)** NN. **(B)** CNN.

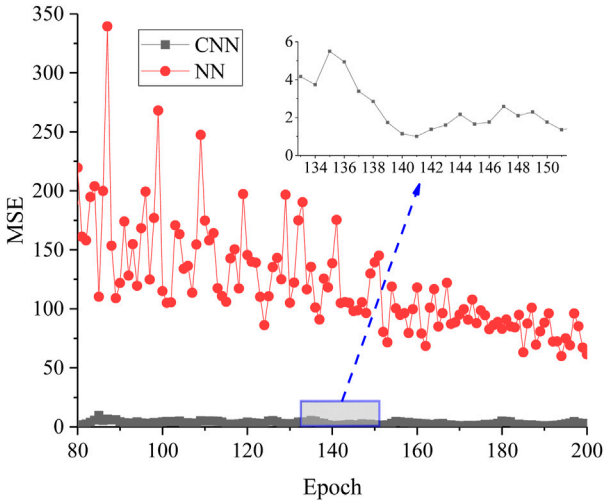


FIGURE 7
Comparison of the training loss curves between the two networks.

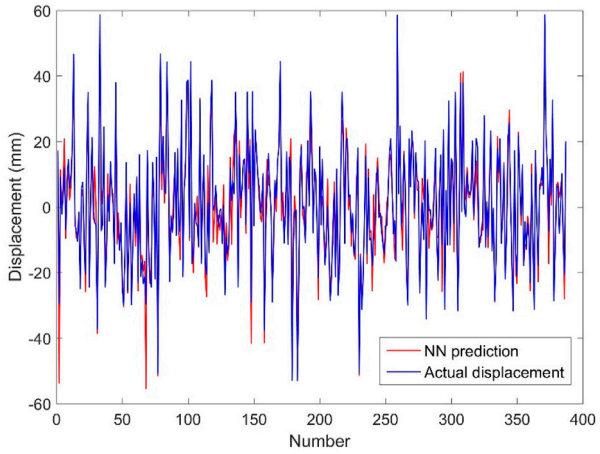


FIGURE 8
The predicted displacement based on NN.

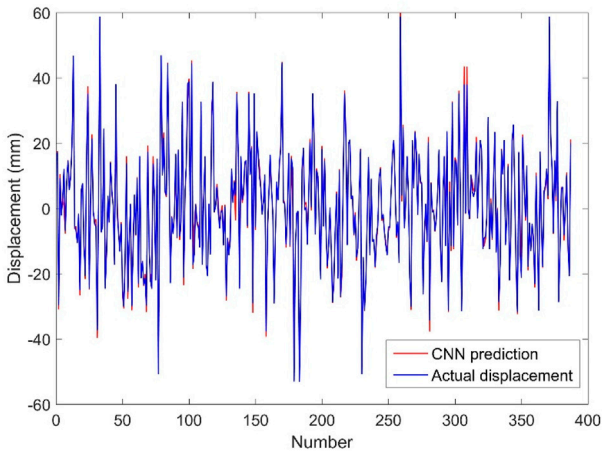


FIGURE 9
The predicted displacement based on CNN.

displacement signals form the dataset. There are a total of 2,387 samples in this data set. The samples in the dataset are then randomly arranged, with 1,600 samples as the training set, 400 samples as the validation set, and 387 samples as the test set.

3.2 Estimation result

In this section, the dataset is trained using a fully connected neural network (NN) and the proposed convolutional neural network (CNN)-based modelling approach, respectively. In order to verify the advantages of convolutional neural network, the proposed convolutional neural network uses the same activation function and loss function as the fully connected neural network. The activation function is ReLU and the loss function is Mean Square Error (MSE). The MSE curves of the two methods are shown in Figure 6. From the figure, it can be seen that when the epoch is 200, the MSE curve of CNN remains basically unchanged. When the epoch is 450, the MSE curve of NN remains basically unchanged. This shows that CNN has stronger feature extraction ability and only 200 epochs are needed to reach the convergence state. The number of epochs for NN to reach the convergence state is 2.25 times more than that of CNN. In CNN, the MSE is 0.899 for the training set and 3.426 for the validation set. In NN, the MSE is 22.386 for the training set and 24.198 for the validation set. When the number of training epochs is less than 80, the MSE of the NN is too large. In order to facilitate the comparison of the effects from the two types of networks, MSE comparisons from 80 epochs to 200 epochs were chosen for this section, as shown in Figure 7. As can be seen from the figure, the MSE of NN gradually decreases from 350 to around 50 whereas the MSE of CNN is very small. As can be seen in the local figure, the MSE of the CNN is only 1.5 when the epoch number is 140. In contrast to NNs, CNNs have significant training efficiency and accuracy in dynamic displacement estimation task.

There were 387 samples in the test set. The predicted displacements are automatically generated by feeding these samples into the displacement estimation model generated by the above two methods. The predicted displacements based on the fully connected neural network are shown in Figure 8. From the figure, it can be seen that the predicted displacement curve based on the fully connected neural network method partially overlaps with the actual displacement curve, and the MSE of the test set is 29.489. It can be seen that the predicted displacements generated by the fully connected neural network have a good degree of overlap with the actual displacements.

The results of displacement prediction based on convolutional neural network are shown in Figure 9. It can be seen that the displacement curve predicted based on convolutional neural network has a great overlap with the actual displacement curve. The MSE of the test set is 3.096, which indicates that the CNN has high accuracy in estimating the dynamic displacement of the structure. For the same test set, the MSE of the convolutional

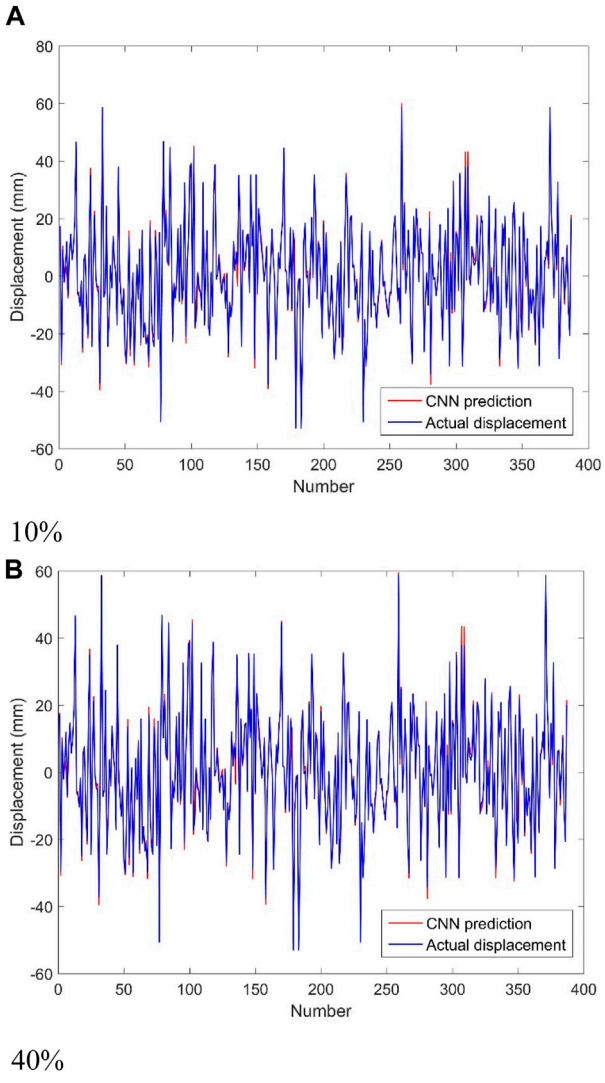


FIGURE 10
The predicted displacements based on test set with noise. **(A)** 10%. **(B)** 40%.

neural network is much smaller than that of the fully connected neural network, and the number of training episodes of the convolutional neural network is also smaller than that of the fully connected neural network. Therefore, compared with the fully connected neural network, the proposed convolutional neural network performs better in displacement estimation with both higher estimation accuracy and higher modelling efficiency.

3.3 The effects of noise

Noise is inevitable in data acquisition systems. Meanwhile, disturbed by the external environment, there are often more noise signals and uncertainties in the sensor-based sensory data. In order to verify the noise resistance of the proposed convolutional neural network, four noise levels (i.e., noise-to-signal ratios of 10%, 20%, 30%, and 40%, respectively) are added to the acceleration signals to investigate the robustness of the CNN to noisy data. These data were directly fed into the trained dynamic displacement estimation model, which was trained and updated from noisy data samples. The dynamic displacement prediction results based on convolutional neural network are shown in Figure 10. It can be seen that the displacements predicted by the convolutional neural network still have good accuracy under the influence of noise. Since the effects of the four noise levels on the convolutional neural network are basically the same, only the predicted displacements based on the test set for two noise levels (10% and 40%) are shown in the figure. Four evaluation indicators (MSE, MAE, MAPE, and R^2) based on the test set for each of the four noise levels are calculated in Table 3. These four-assessment metrics change very little as the noise-to-credit ratio continues to increase. Especially for R^2, its value is always 0.991. The results showed that the convolutional neural network is robust against noise. The noise of data acquisition has little effect on it because the convolutional and pooling layers in the convolutional neural network are similar to filters and are more powerful than normal filters.

TABLE 3 Estimation error for test samples with varying degrees of noise.

Noise-to-signal ratios (%)	MSE	MAE	MAPE (%)	R^2
0	3.096	1.032	19.2	0.991
10	3.115	1.030	18.8	0.991
20	3.106	1.041	19.3	0.991
30	3.121	1.039	19.4	0.991
40	3.124	1.037	19.3	0.991

4 Physical model test

4.1 Dataset generation

In Section 3, acceleration and displacement signals are generated by numerical simulation. The dataset is trained to estimate the dynamic displacements. However, the data generated by numerical simulation is idealized. The acceleration and displacement signals of the actual structure are acquired by sensors. The data collected by the sensors is incomplete and complex due to the noise and missing data characteristics of the sensors. Therefore, it is necessary to verify the effect of the proposed convolutional neural network on the actual structure. The design of the actual frame structure is shown in Figure 11. In the case where the real frame structure is subjected to transient excitation, the acceleration and displacement signals of the nodes are collected by piezoelectric accelerometers and laser displacement sensors. The piezoelectric accelerometers and laser displacement sensors are acquired at a frequency of 100 Hz. This real dynamic displacement estimation dataset is formed in the same way as in Section 3. The dataset contains a total of 2,333 samples, of which 1,600 samples are used as the training set, 400 samples are used as the validation set, and 333 samples are used as the test set.

4.2 Estimation result

The training set is fed into the proposed convolutional neural network. The MSE curve is shown in Figure 12. It can be seen that

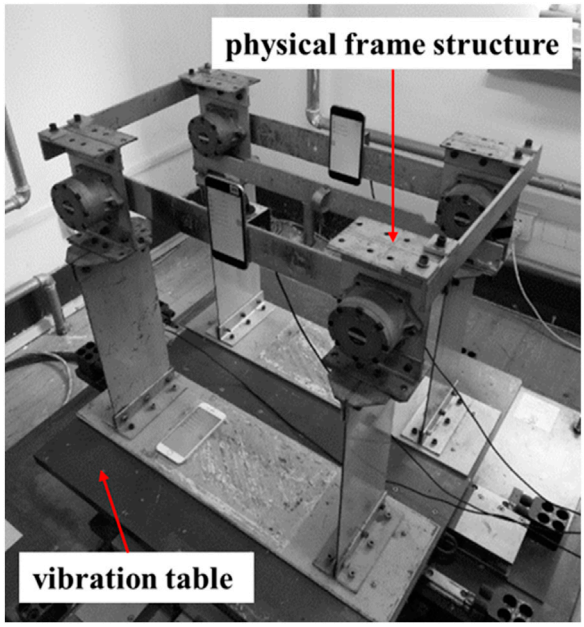


FIGURE 11
Physical frame structure.

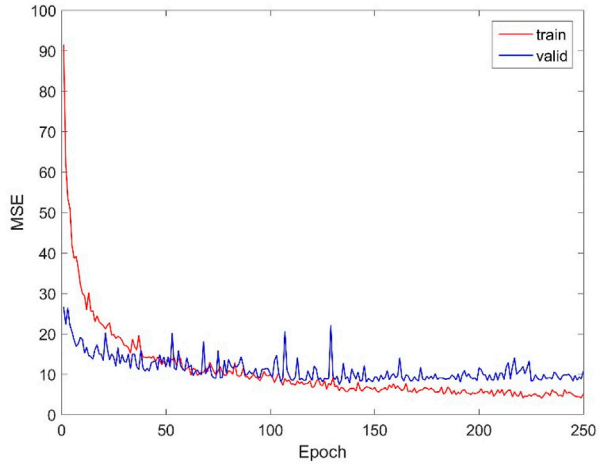


FIGURE 12
The MSE curves.

the MSE curve remains almost constant when the epoch is 250. This indicates that the model has basically reached the convergence state. The MSE based on the training set is 5.097 and the MSE based on the validation set is 10.770. The MSE value can satisfy the need of engineering design.

The test set contains 333 samples and the test set is fed into the dynamic displacement estimation model. The predicted and actual displacements are shown in Figure 13. It can be seen that the predicted displacement curves are in high agreement with the actual displacement curves. The MSE based on the test set is 6.150. The results show that the displacement estimation method

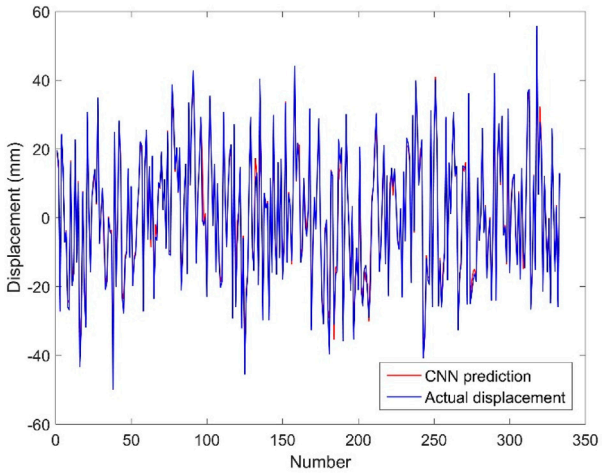


FIGURE 13
The predicted displacements and actual displacements.

based on convolutional neural network can accurately estimate the displacement signal using the acceleration signal.

4.3 Visualization of the output

A neural network is similar to a black box. It is difficult to understand and does not give direct visualization of the output. Data processing in neural networks is invisible. Convolutional neural network is one of the neural networks which also has this characteristic. Therefore, in order to have a clearer understanding of how convolutional neural networks process data, we have visualized the output of the hidden layer of the convolutional neural network. The proposed convolutional neural network consists of an input layer, four convolutional layers, a pooling layer, a flattening layer and an output (dense) layer. In convolutional neural network, convolutional and pooling layers are the core components.

In this section, the output of the convolutional layer and the pooling layer is visualized. The first sample data in the test set is shown in Figure 14A, and the curve fluctuates between 150 and 1000 mm/s^2. The input data is passed to the first convolutional layer. The first convolutional layer has 6 convolutional kernels. Each convolution kernel processes the input data to produce a 46 × 1 vector. Therefore, there are 6 vectors (46 × 1). One of the vectors is shown in Figure 14B, where the curve fluctuates between 50 and 300. The second convolutional layer has 12 convolutional kernels, and there are 12 vectors (42 × 1). One of the vectors is shown in Figure 14C, and the curve fluctuates between 0 and 100. The third convolutional layer has 64 convolutional kernels and a total of 64 vectors (38 × 1). One of the vectors is shown in Figure 14D, and the curve fluctuates between 3 and 16. The fourth convolutional layer has 128 convolutional kernels with 128 vectors (34 × 1). One of the vectors is shown in Figure 14E, and the curve fluctuates between 0 and 2. As the data is processed through the convolutional layers, the value of each data becomes smaller. The pooling layer has 128 convolutional kernels and a total of 128 vectors (17 × 1). One of the vectors is shown in Figure 14F, where the curve fluctuates between 0 and 2. It can be seen that the pooling layer does not change the value of the data, but reduces the amount of data. The data curve

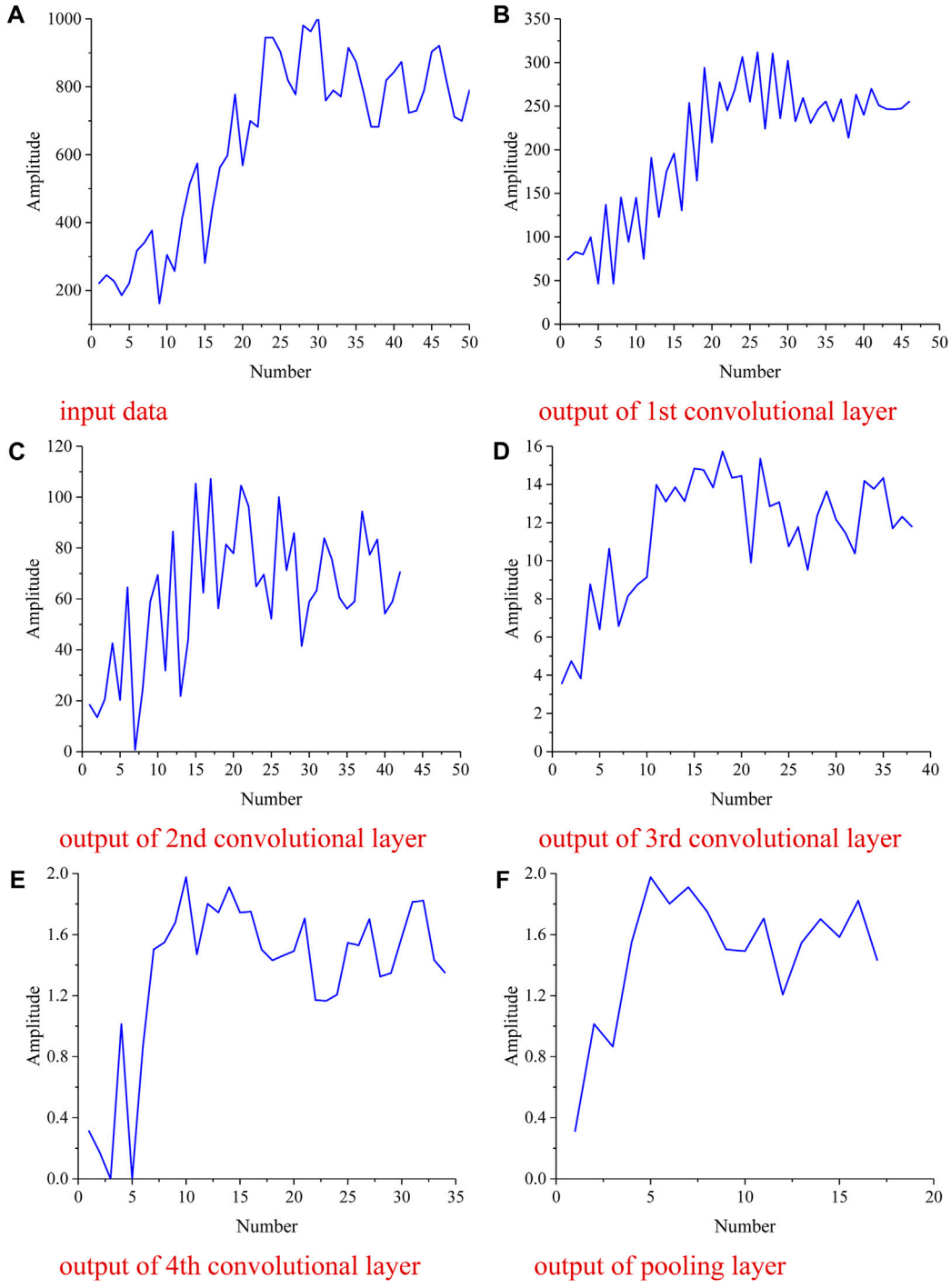


FIGURE 14
The output of hidden layers. **(A)** input data. **(B)** output of 1st convolutional layer. **(C)** output of 2nd convolutional layer. **(D)** output of 3rd convolutional layer. **(E)** output of 4th convolutional layer. **(F)** output of pooling layer.

becomes smoother after the data is processed by the pooling layer. The convolutional and pooling layers can also act as filters, so the convolutional neural network has strong noise immunity. Although the size of the output vector of the hidden layer is decreasing, the overall shape of the original data is still well preserved.

The output of the pooling layer is a multidimensional vector. The Flatten layer then converts the multidimensional vectors into one-dimensional vectors. The Flatten layer rearranges the data into a one-dimensional form, preserving the feature order of the multidimensional vectors. This allows the subsequent Fully Connected layer to learn based on the overall features of the input and to model the relationships between the features. The one-dimensional vectors are fed into the fully connected (dense) layer. Finally, an estimate of the dynamic displacement is output. Table 4 shows the predicted and actual

TABLE 4 The predicted error of test set.

Count	Predicted value	Actual value	Error (%)
1	19.342	19.635	1.49
2	14.212	15.670	9.30
3	−25.742	−27.300	5.71
4	22.301	24.305	8.25
5	14.447	14.425	0.15
6	−5.952	−7.172	17.01
7	−3.662	−3.892	5.91
8	−24.783	−25.935	4.44
9	−23.849	−26.812	11.05
10	16.703	16.120	3.62

values of the first 10 samples in the test set. It can be seen that only the sixth and ninth samples have an error greater than 10%, while the other samples have an error between 0.15% and 9.30%. The average error is 6.69%, so the proposed method can meet the engineering needs.

5 Discussion

In this paper, a 1D convolutional neural network is used for data-driven modelling of the complex relationship between acceleration signals and dynamic displacements. The dynamic displacement estimation method based on convolutional neural network is verified by numerical simulation to be superior to the method based on fully connected neural network in terms of accuracy, stability and convergence efficiency. Noise is added to the test set and these data are directly fed into the dynamic displacement estimation model which is trained from noise-free data samples. It is worth noting that the displacements estimated by the convolutional neural network model still have good accuracy under the influence of noise. Subsequently, a real frame structure is used to verify the feasibility of the proposed method in real engineering applications. Although the root-mean-square error of the predicted displacements of the real structure is larger than the root-mean-square error of the predicted displacements of the finite element model, the predicted displacements of the real structure satisfy the engineering needs. This is mainly due to the fact that the sensors are affected by many factors when sensing the dynamic characteristics of the structure, and the collected signals contain more noise and uncertainty. The results of the output visualization of the convolutional neural network show that the convolution operation not only reduces the amount of data, but also reduces the data values. In addition, the convolutional and pooling layers act as filters, which results in a convolutional neural network with strong noise immunity.

6 Conclusion

In this paper, a dynamic displacement estimation method based on convolutional neural network and acceleration is proposed. The acceleration and displacement signals are trained to generate an estimation model. The acceleration data is input into this estimation model, and the displacement estimate can be output automatically. Numerical simulations and physical model tests verify the feasibility and stability of the method. Some important results are obtained: the 1D convolutional neural network can accurately model the complex relationship between acceleration timing signals and displacement signals; the convergence efficiency of the convolutional neural network is much greater than that of a typical neural network when modelling the complex relationship using the convolutional neural network. The updating efficiency of the former is 2.25 times that of the latter; due to the existence of the convolution kernel in the convolutional neural network, it can filter the data and has a strong anti-noise ability; in this paper, the visualization part at the end fully reflects the filtering effect of the convolution kernel on the noise. The proposed method can easily and quickly estimate the dynamic displacement of structures using acceleration information. It also still shows good results in noisy environments. However, the proposed method is affected by the accuracy of the acceleration signal. To further improve the accuracy of displacement estimation, multi-sensor fusion such as accelerometers and strain gauges can be considered. In addition, deep learning architectures based on multi-sensor fusion need to be further explored.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

XZ: Data curation, Formal Analysis, Methodology, Validation, Visualization, Writing–original draft. YH: Conceptualization, Resources, Supervision, Writing–review and editing.

Funding

The authors declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Park J-W, Lee JJ, Jung HJ, Myung H. Vision-based displacement measurement method for high-rise building structures using partitioning approach. *Ndt E Int* (2010) 43(7):642–7. doi:10.1016/j.ndteint.2010.06.009

2. Zhang Y, Wang T, Yuen K-V. Construction site information decentralized management using blockchain and smart contracts. *Computer-Aided Civil Infrastructure Eng* (2022) 37(11):1450–67. doi:10.1111/mice.12804

3. Chan THT, Yu L, Tam H, Ni Y, Liu S, Chung W, et al. Fiber Bragg grating sensors for structural health monitoring of Tsing Ma bridge: background and experimental observation. *Eng structures* (2006) 28(5):648–59. doi:10.1016/j.engstruct.2005.09.018

4. Zhang Y, Yuen K-V. Review of artificial intelligence-based bridge damage detection. *Adv Mech Eng* (2022) 14(9):168781322211227. doi:10.1177/16878132221122770

5. He Y, Zhou Q, Xu F, Sheng X, He Y, Han J. An investigation into the effect of rubber design parameters of a resilient wheel on wheel-rail noise. *Appl Acoust* (2023) 205: 109259. doi:10.1016/j.apacoust.2023.109259

6. He Y, Zhang Y, Yao Y, He Y, Sheng X. Review on the prediction and control of structural vibration and noise in buildings caused by rail transit. *Buildings* (2023) 13: 2310. doi:10.3390/buildings13092310

7. Lynch JP, Loh KJ. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock Vibration Dig* (2006) 38(2):91–128. doi:10.1177/0583102406061499

8. Giurgiutiu V, Zagrai A, Bao JJ. Piezoelectric wafer embedded active sensors for aging aircraft structural health monitoring. *Struct Health Monit* (2002) 1(1):41–61. doi:10.1177/147592170200100104

9. Farrar CR, Keith W. An introduction to structural health monitoring. *Phil Trans R Soc A: Math Phys Eng Sci* (2006) 365(1851):303–15. doi:10.1098/rsta.2006.1928

10. Park G, Rosing T, Todd MD, Farrar CR, Hodgkiss W. Energy harvesting for structural health monitoring sensor networks. *J Infrastructure Syst* (2008) 14(1):64–79. doi:10.1061/(asce)1076-0342(2008)14:1(64)

11. Amann M-C, Bosch T. M., Lescure M., Myllylae R. A., Rioux M., et al. Laser ranging: a critical review of unusual techniques for distance measurement. *Opt Eng* (2001) 40(1):10–20. doi:10.1117/1.1330700

12. Bitou Y, Schibli TR, Minoshima K. Accurate wide-range displacement measurement using tunable diode laser and optical frequency comb generator. *Opt express* (2006) 14(2):644–54. doi:10.1364/opex.14.000644

13. Lee JJ, Fukuda Y, Shinozuka M, Cho S, Yun CB. Development and application of a vision-based displacement measurement system for structural health monitoring of civil structures. *Smart Structures Syst* (2007) 3(3):373–84. doi:10.12989/sss.2007.3.3.373

14. Feng D, Feng M, Ozer E, Fukuda Y. A vision-based sensor for noncontact structural displacement measurement. *Sensors* (2015) 15(7):16557–75. doi:10.3390/s150716557

15. Rice JA, Li C, Gu C, Hernandez JC. A wireless multifunctional radar-based displacement sensor for structural health monitoring. In: Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems; April 2011; San Diego, California, USA (2011). doi:10.1117/12.879243

16. Yi TH, Li HN, Gu M. Recent research and applications of GPS-based monitoring technology for high-rise structures. *Struct Control Health Monit* (2013) 20(5):649–70. doi:10.1002/stc.1501

17. Sabatini AM, Ligorio G, Mannini A. Fourier-based integration of quasi-periodic gait accelerations for drift-free displacement estimation using inertial sensors. *Biomed Eng Online* (2015) 14(1):106. doi:10.1186/s12938-015-0103-8

18. Park K-T, Kim SH, Park HS, Lee KW. The determination of bridge displacement using measured acceleration. *Eng Structures* (2005) 27(3):371–8. doi:10.1016/j.engstruct.2004.10.013

19. Roberts GW, Meng X, Dodson AH. Integrating a global positioning system and accelerometers to monitor the deflection of bridges. *J Surv Eng* (2004) 130(2):65–72. doi:10.1061/(asce)0733-9453(2004)130:2(65)

20. Gindy M, Vaccaro R, Nassif H, Velde J. A state-space approach for deriving bridge displacement from acceleration. *Computer-Aided Civil Infrastructure Eng* (2008) 23(4): 281–90. doi:10.1111/j.1467-8667.2007.00536.x

21. Rui Z, Min F, Dou Y, Ye B. Switching mechanism and hardware experiment of a non-smooth Rayleigh-Duffing system. *Chin J Phys* (2023) 82:134–48. doi:10.1016/j.cjph.2023.02.001

22. Yao W, Wang C, Sun Y, Gong S, Lin H. Event-triggered control for robust exponential synchronization of inertial memristive neural networks under parameter disturbance. *Neural Networks* (2023) 164:67–80. doi:10.1016/j.neunet.2023.04.024

23. Yao W, Wang C, Sun Y, Zhou C. Robust multimode function synchronization of memristive neural networks with parameter perturbations and time-varying delays. *IEEE Trans Syst Man, Cybernetics: Syst* (2020) 52(1):260–74. doi:10.1109/tsmc.2020.2997930

24. Yao W, Gao K, Zhang Z, Cui L, Zhang J. An image encryption algorithm based on a 3D chaotic Hopfield neural network and random row–column permutation. *Front Phys* (2023) 11:1162887. doi:10.3389/fphy.2023.1162887

25. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* (2012).

26. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition (2014). Availab;e at: https://arxiv.org/abs/1409.1556.

27. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L, et al. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; June 2014; Columbus, OH, USA (2014).

28. He K, Zhang X, Ren S, Sun J, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2016; Las Vegas, NV, USA (2016).

29. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* (2017) 542(7639): 115–8. doi:10.1038/nature21056

30. Moeskops P, Viergever MA, Mendrik AM, de Vries LS, Benders MJNL, Isgum I. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Trans Med Imaging* (2016) 35(5):1252–61. doi:10.1109/tmi.2016.2548501

31. CireşAn D, Meier U, Masci J, Schmidhuber J. Multi-column deep neural network for traffic sign classification. *Neural networks* (2012) 32:333–8. doi:10.1016/j.neunet.2012.02.023

32. Jin J, Fu K, Zhang C. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Trans Intell Transportation Syst* (2014) 15(5):1991–2000. doi:10.1109/tits.2014.2308281

33. Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil Infrastructure Eng* (2017) 32(5):361–78. doi:10.1111/mice.12263

34. Zhang Y, Yuen K-V. Crack detection using fusion features-based broad learning system and image processing. *Computer-Aided Civil Infrastructure Eng* (2021) 36(12): 1568–84. doi:10.1111/mice.12753

35. Ge Y, Zhu F, Chen D, Zhao R, Wang X, Li H. Structured domain adaptation with online relation regularization for unsupervised person re-id. *IEEE Trans Neural Networks Learn Syst* (2022) 1–14. doi:10.1109/tnnls.2022.3173489

36. Du F, Wu S, Xing S, Xu C, Su Z. Temperature compensation to guided wave-based monitoring of bolt loosening using an attention-based multi-task network. *Struct Health Monit* (2023) 22(3):1893–910. doi:10.1177/14759217221113443

37. Zhang Y, Yuen K-V. Bolt damage identification based on orientation-aware center point estimation network. *Struct Health Monit* (2022) 21(2):438–50. doi:10.1177/14759217211004243

38. Li L, Sun HX, Zhang Y, Yu B. Surface cracking and fractal characteristics of bending fractured polypropylene fiber-reinforced geopolymer mortar. *Fractal and Fractional* (2021) 5(4):142. doi:10.3390/fractalfract5040142

39. Li L, Zhang Y, Shi Y, Xue Z, Cao M. Surface cracking and fractal characteristics of cement paste after exposure to high temperatures. *Fractal and Fractional* (2022) 6(9): 465. doi:10.3390/fractalfract6090465

40. Zhang Y, Yuen KV. Time-frequency fusion features-based incremental network for smartphone measured structural seismic response classification. *Eng Structures* (2023) 278:115575. doi:10.1016/j.engstruct.2022.115575

41. Zhang Y, Yuen KV, Mousavi M, Gandomi AH. Timber damage identification using dynamic broad network and ultrasonic signals. *Eng Structures* (2022) 263:114418. doi:10.1016/j.engstruct.2022.114418

42. Malek S, Melgani F, Bazi Y. One-dimensional convolutional neural networks for spectroscopic signal regression. *J Chemometrics* (2018) 32(5):e2977. doi:10.1002/cem.2977

43. Hasegawa C, Iyatomi H. One-dimensional convolutional neural networks for Android malware detection. In: Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA); March 2018; Penang, Malaysia. IEEE (2018).

44. Shrestha A, Mahmood A. Review of deep learning algorithms and architectures. *IEEE access* (2019) 7:53040–53065. doi:10.1109/ACCESS.2019.2912200

45. Glorot X, Antoine B, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS); April 2011; Fort Lauderdale, FL, USA (2011).

46. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern recognition* (2018) 77:354–77. doi:10.1016/j.patcog.2017.10.013

47. Kim P, Kim P. Convolutional neural network. In: *MATLAB deep learning: with machine learning, neural networks and artificial intelligence*. Berlin, Germany: Springer (2017). p. 121–47.

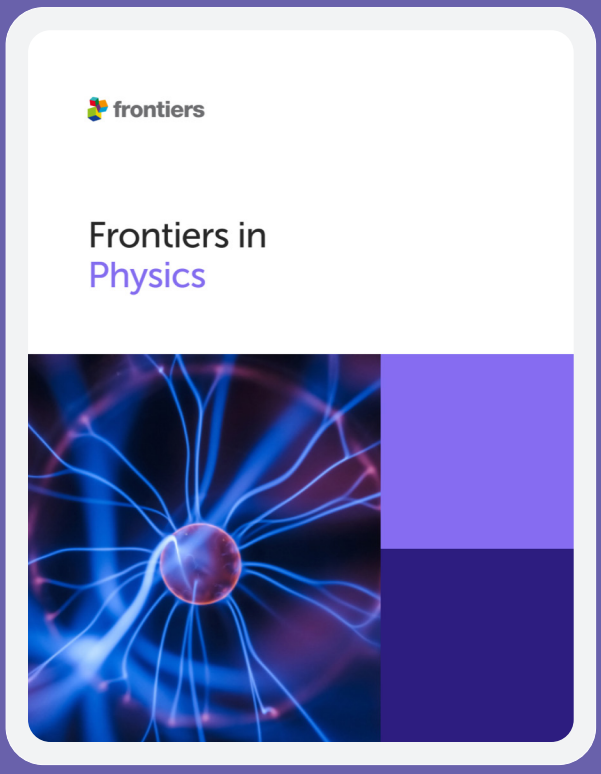
Frontiers in Physics

Investigates complex questions in physics to understand the nature of the physical world

Addresses the biggest questions in physics, from macro to micro, and from theoretical to experimental and applied physics.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact