Structural Stability of Spiking Neural Networks

Gao Zhang[1], Shao-Qun Zhang[2*]

[1] *Institute of Mathematics, Nanjing Normal University, Nanjing, 210046, China*
[2] *National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Abstract

The past decades have witnessed an increasing interest in spiking neural networks (SNNs) due to their great potential of modeling time-dependent data. Many algorithms and techniques have been developed; however, theoretical understandings of many aspects of spiking neural networks are still cloudy. A recent work [32] disclosed that typical SNNs could hardly withstand both internal and external perturbations due to their bifurcation dynamics and suggested that self-connection has to be added. In this paper, we investigate the theoretical properties of SNNs with self-connection, and develop an in-depth analysis on structural stability by specifying the lower and upper bounds of the maximum number of bifurcation solutions. Numerical experiments conducted on simulation and practical tasks demonstrate the effectiveness of the proposed results.

Key words: Spiking Neural Networks, Self Connections, Polynomial Bifurcation Fields, Structural Stability, Bifurcation Solutions

1. introduction

Spiking neural networks (SNNs) have gained progressive momentum during the last decades due to the flexibility of mimicking the neuronal plasticity and the potential of modeling time-dependent data [26, 29]. Great progress has been made for SNNs in vision [27], speech recognition [24], reinforcement learning [30], few-short learning [10], neuromorphic computing [21], etc. The computational process of SNNs complies with an integration-and-firing paradigm, which usually is formulated as some first-order parabolic equations. An omnipresent challenge in SNNs is to qualify the behavior of structures under perturbations, that is, structural stability [1]. However, it is difficult to assess whether and to what extent a given SNN is unaffected by exactly $\mathcal{C}^1$-small perturbations.

Currently, if a model's capacity to withstand perturbations is measured at all, it is typically investigated heuristically against some tasks [2, 11], e.g., observing the accuracy change of SNNs after adding perturbations, in

*Corresponding author. Email: zhangsq@lamda.nju.edu.cn

which success is taken as an indicator of stability in an ocular sense. Though undoubtedly helpful, such heuristics are rarely defined concerning an underlying mathematical property of interest, nor do they necessarily have any correspondence to the tasks subsequently confronted by the trained SNN. In this paper, we theoretically investigate the structural stability of SNNs. Our starting point is a recent advance of Zhang et al. [32], who proved that typical SNNs are bifurcation dynamical systems in which small perturbations may cause catastrophic damage, and adding *self-connection* can enhance the stability ability of SNNs, which fully connects the spiking neurons in the same hidden layer.

This work initiates an in-depth analysis by providing the lower and upper bounds of the maximum number of bifurcation solutions in perturbed SNNs with higher-order self-connection; the upper bound indicates the relation between the structural stability and computational configurations of SNNs, and the lower bound specifies the difficult nature of this problem. Consequently, the challenge of qualifying structural stability in SNNs can be re-framed as a mathematical problem with quantitative property.

Our main contributions are summarized as follows:

- We present the spiking neural network with Polynomial bIfuRcATion fiEld (PIRATE), which provides a more general paradigm for SNNs with self-connection.
- We prove the existence of bifurcation solutions in PIRATE with perturbations in Theorem 1.
- We provide an algorithmic approach for calculating the upper bounds of the maximum number of bifurcation solutions in Algorithm 1.
- We prove the lower bound of the maximum number of bifurcation solutions in PIRATE in Theorem 2, which specifies the difficult nature of the problem of qualifying structural stability.
- Numerical experiments conducted on several simulation and practical tasks demonstrate the above theoretical results and the effectiveness of PIRATE, respectively.

The rest of this paper is organized as follows. Section 2 reviews some useful notations. Section 3 presents the PIRATE model with provable and computational bounds for its structural stability. Section 4 conducts experiments to verify the theoretical results and effectiveness of PIRATE. Section 5 concludes this work with discussions and prospects.

2. Notations

We first introduce some useful notations. Let $[N] = \{1, 2, \dots, N\}$ be an integer set for $N \in \mathbb{N}^+$, and $|\cdot|_\#$ denotes the number of elements in a collection, e.g., $|[N]|_\# = N$. Given two functions $g, h \colon \mathbb{N}^+ \to \mathbb{R}$, we denote by $h = \Theta(g)$ if there exist positive constants c_1, c_2, and n_0 such that $c_1 g(n) \leq h(n) \leq c_2 g(n)$ for every $n \geq n_0$; $h = \mathcal{O}(g)$ if there exist positive constants c and n_0 such that $h(n) \leq cg(n)$ for every $n \geq n_0$; $h = \Omega(g)$ if there exist positive constants c and n_0 such that $h(n) \geq cg(n)$ for every $n \geq n_0$; $h = o(g)$ if there exist positive constants c and n_0 such that $h(n) < cg(n)$ for every $n \geq n_0$. Let $\mathcal{C}(\mathbb{K}, \mathbb{R})$ be the set of all scalar functions $f : \mathbb{K} \to \mathbb{R}$ continuous on $\mathbb{K} \subset \mathbb{R}^m$. Given $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^\top$, $\alpha_i \geq 0$ for $i \in [m]$, $|\boldsymbol{\alpha}| = \sum_{i \in [m]} \alpha_i$, and $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in K$, we define an n-tuple partial derivative

$$D^{|\boldsymbol{\alpha}|} f(\boldsymbol{x}) = \frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x^{\alpha_2}} \cdots \frac{\partial^{\alpha_n}}{\partial x^{\alpha_n}} f(\boldsymbol{x}) \ .$$

Further, we define

$$\mathcal{C}^l(\mathbb{K}, \mathbb{R}) = \{ f \mid f \in \mathcal{C}(\mathbb{K}, \mathbb{R}),\ D^r f \in \mathcal{C}(\mathbb{K}, \mathbb{R}),\ r \in [l] \} \ .$$

For $1 \leq p < \infty$, we define

$$\mathcal{L}^p(\mathbb{K}, \mathbb{R}) = \left\{ f \ \middle|\ f \in \mathcal{C}(\mathbb{K}, \mathbb{R}),\ \|f\|_{p,\mathbb{K}} < \infty \right\} \ ,$$

where

$$\|f\|_{p,\mathbb{K}} \stackrel{\text{def}}{=} \left(\int_{\mathbb{K}} |f(\boldsymbol{x})|^p \, \mathrm{d}\boldsymbol{x} \right)^{1/p} \ .$$

This work also considers the Sobolev space $\mathcal{W}^{l,p}_{\mu}(\mathbb{K}, \mathbb{R})$, defined as the collection of all functions $f \in \mathcal{C}^l(\mathbb{K}, \mathbb{R})$ and $D^r f \in \mathcal{L}^p(\mathbb{K}, \mathbb{R})$ for all $|r| \in [l]$, that is,

$$\|D^r f\|_{p,\mathbb{K}} = \left(\int_{\mathbb{K}} |D^r f(\boldsymbol{x})|^p \, \mathrm{d}\boldsymbol{x} \right)^{1/p} < \infty \ .$$

3. PIRATE with Structural Stability

We begin our work with introducing the bifurcation spiking neuron models proposed by Zhang et al. [32].

$$\frac{\partial \boldsymbol{u}(t)}{\partial t} = -\frac{\boldsymbol{u}(t)}{\tau_m} + \boldsymbol{u}^*(\boldsymbol{\lambda}, t) + \frac{R}{\tau_m} \mathbf{W} \mathbf{I}(t) \ , \tag{1}$$

where $\boldsymbol{u}(t) = (u_1(t), \ldots, u_N(t))^\top$ indicates the membrane potential vector of N spiking neurons at timestamp t, $\mathbf{I}(t) = (I_1(t), \ldots, I_M(t))^\top$ denotes the M-dimensional input signals, the vector $\boldsymbol{u}^* = (u_1^*, \ldots, u_N^*)^\top$ portrays the mutual promotion between neurons adjusted by the bifurcation parameters $\boldsymbol{\lambda}$, $\mathbf{W}$ is the learnable connection matrix, τ_m and R are positive-valued hyper-parameters with respect to membrane time and membrane resistance, respectively. The advance [32] provided a linear implementation for the term $\boldsymbol{u}^*(\boldsymbol{\lambda}, t)$ in Eq. (1), i.e., unfolding the k-th variable as $u_k^*(\boldsymbol{\lambda}, t) = \sum_{i \neq k} \lambda_{ki} u_i + o(|u_k|)$ where $o(|u_k|)$ denotes the high-order term of u_k for $k \in [N]$. Hence, Eq. (1) becomes

$$\frac{\mathrm{d}u_k(t)}{\mathrm{d}t} = -\frac{u_k(t)}{\tau_m} + \sum_{i \in [N]} \lambda_{ki} u_i(t) + \frac{R}{\tau_m} \sum_{j \in [M]} \mathbf{W}_{kj} \mathbf{I}_j(t) \ , \tag{2}$$

omitted the high-order term $o(|u_k|)$ for $k \in [N]$. Here, $\sum_{j\in[M]} \mathbf{W}_{kj}\mathbf{I}_j(t)$ indicates the signal received by neuron k at timestamp t, and $\sum_{i\in[N]} \lambda_{ki}u_i(t)$ denotes the effect on the membrane potential of neuron k when neuron i fires a spike.

This work ponders the effect of self-connection from the perspective of structural stability. We first generalize the above self-connection topology with linear implementation and present a more general paradigm with a high-order $o(|u_k|)$.

$$u_k^*(\boldsymbol{\lambda}, t) = \sum \lambda_{ki}^1 u_i(t) + \boldsymbol{\lambda}_k^p \sum_{p=2}^{n} \mathrm{P}_p(\boldsymbol{u}(t), t) \ , \tag{3}$$

abbreviated as $u_k^*(\boldsymbol{\lambda}, t) = \mathrm{Poly}(\boldsymbol{u}(t); n)$, where

$$\mathrm{P}_p(\boldsymbol{u}(t), t) = (u_1)^{\alpha_1}(u_2)^{\alpha_2}\dots(u_N)^{\alpha_N} \ ,$$

in which $\alpha_1 + \alpha_2 + \cdots + \alpha_N = p$. Taking an example of $N = 2$ and $n = 2$, we have

$$\begin{aligned} u_1^*(\boldsymbol{\lambda}, t) &= \mathrm{Poly}(u_1(t), u_2(t); n = 2) \\ &= \lambda_{11}^1 u_1(t) + \lambda_{12}^1 u_2(t) \\ &\quad + \lambda_{11}^2 u_1(t)^2 + \lambda_{22}^2 u_2(t)^2 + \lambda_{12}^2 u_1(t)u_2(t) \ , \end{aligned}$$

and

$$\begin{aligned} u_2^*(\boldsymbol{\lambda}, t) &= \mathrm{Poly}(u_1(t), u_2(t); n = 2) \\ &= \lambda_{21}^1 u_1(t) + \lambda_{22}^1 u_2(t) \\ &\quad + \lambda_{11}^2 u_1(t)^2 + \lambda_{22}^2 u_2(t)^2 + \lambda_{12}^2 u_1(t)u_2(t) \ . \end{aligned}$$

Appendix D records the training procedure of PIRATE.

This work theoretically investigates the structural stability of PIRATE. The key idea is to add the perturbations led by a small parameter ϵ to a center point of the algebraic Eq. (4), and observe whether the perturbed system bifurcates either from the center point or from some periodic orbits surrounding the center point. Hence, the upper and lower bounds of the maximum number of bifurcation solutions can be bounded by the number of center points where bifurcation occurs and the concerned periodic orbits (i.e., limit cycles), respectively.

Formally, we define $H(n)$ to denote the maximum number of bifurcation solutions of dynamical systems in Eq. (1) with n-order polynomial bifurcation fields in Eq. (3). We first exhibit the algebraic formation of Eq. (1)

$$\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = -\frac{u_i(t)}{\tau_m} + \mathrm{Poly}(\boldsymbol{u}(t); n) \tag{4}$$

for $i \in [N]$ and the corresponding perturbed system

$$\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = -\frac{u_i(t)}{\tau_m} + \mathrm{Poly}(\boldsymbol{u}(t); n) + \epsilon\, \mathrm{Poly}(\boldsymbol{u}(t); m) \ , \tag{5}$$

where ϵ indicates a small parameter that scales the perturbation magnitude with respect to $m \in \mathbb{N}^+$. Here, we are interested in the small limit cycles of Eq. (5), which bifurcate at ϵ from the center points of Eq. (4) as $|\epsilon| \to 0$.

In the rest section, we prove the existence of bifurcation solutions in perturbed PIRATE system in Subsection 3.1, provide an algorithmic way for calculating the upper bound in Subsection 3.2, and find the lower bound in Subsection 3.3.

3.1. Existence of Bifurcation Solutions

Now, we present the first theorem as follows.

Theorem 1 *Let $\tilde{u}$ be a center point of system* (5)*. For $|\epsilon| > 0$ sufficiently small, there exists a 2π-periodic (bifurcation) solution $f(t,\epsilon)$ of system* (5) *s.t. $f(0,\epsilon) \to \tilde{u}$ as $\epsilon \to 0$.*

Theorem 1 shows the existence of bifurcation solutions (i.e., 2π-periodic bifurcation limit cycles) of the perturbed systems. This result holds based on some useful lemmas as follows.

Lemma 1 *The perturbed system* (5) *induces a planar differential equation with the normal form as follows*

$$\frac{\partial f(t,\epsilon)}{\partial t} = \sum_{k=0}^{K} \epsilon^k F_k(t,f) + \epsilon^{K+1}\mathrm{Rest}(t,\epsilon,f) \;, \tag{6}$$

where $F_k : \mathbb{R} \times \mathbb{K} \to \mathbb{R}$ for $k \in [K]$ and $\mathrm{Rest} : \mathbb{R} \times \mathbb{K} \times [-\epsilon_0, \epsilon_0] \to \mathbb{R}$ are $\mathcal{C}^k$-continuous functions in which $\epsilon_0 \in \mathbb{R}$.

Lemma 1 shows an equivalent formation of the concerned system (5), which contributes to the periodic solutions of recursive formations as shown in the following lemma.

Lemma 2 *Let $\tilde{u}$ and $f(t,\epsilon) : [0,T] \times [-\epsilon_0, \epsilon_0] \to \mathbb{R}$ be the center point and solution of system* (5)*, respectively, which satisfies that $f(0,\epsilon) = \tilde{u}$. Then for $t \in [0,T]$, we have*

$$\begin{aligned} f(t,\epsilon) = \tilde{u} + \int_0^t F_0(s,\tilde{u})\,\mathrm{d}s + \sum_{k=0}^{K} \epsilon^k G_k(t,\tilde{u}) \\ + \epsilon^{K+1}\left[\int_0^t \mathrm{Rest}(s, f(s,\epsilon), \epsilon)\,\mathrm{d}s + \mathcal{O}(1)\right] \;, \end{aligned}$$

where G_k (for $k \in [K]$) is of recursive form as follows

$$G_k(t,u) = \int_0^t \left[F_k(s,u) + \mathcal{G}\left(D^{h(r)}F_r(s,u), G_r(s,u)\right)\right]\mathrm{d}s \;,$$

for all $r \in [k-1]$, in which

$$\mathcal{G} = \sum_{r=1}^{k} \sum_{\mathcal{S}_r} \frac{D^{h(r)}F_{k-r}(s,u)}{\alpha_1!(\alpha_2!2!^{\alpha_2})\dots(\alpha_r!r!^{\alpha_r})} \prod_{l=1}^{r} G_l(s,u)^{\alpha_j} \;,$$

for $\mathcal{S}_r$ denotes the set of all r-tuples of non-negative integers $\{\alpha_j\}_{j\in[r]}$ as noted in Section 2, satisfying

$$\sum_j j\alpha_j = r \quad \textit{and} \quad h(r) = \sum_j \alpha_j \;.$$

Lemma 2 provides the recursive formation of the periodic solutions (i.e., limit cycles) of system (5).

Finishing the Proof of Theorem 1. The complete proof of Theorem 1 can be accessed in Appendix A. Let $f(0,\epsilon)=\tilde{u}$. Let $U \subset \mathbb{K}$ be a neighborhood of the center point $\tilde{u}$ such that $G_k(t,u) \neq 0$ for all $u \in \bar{U}/\tilde{u}$ and the Brouwer degree $d_B(G_k,U,\tilde{u}) \neq 0$ [14]. For each $u \in \bar{U}$, there exists $\epsilon_0 > 0$ such that the function $f(t,\epsilon)$ is defined on $[0,T] \times [-\epsilon_0,\epsilon_0]$ once $\epsilon \in [-\epsilon_0,\epsilon_0]$. Thus, $f(t,\epsilon) : [0,T] \times [-\epsilon_0,\epsilon_0] \to \mathbb{R}$ indicates the solution of system (5), as defined in Lemma 2. From the Existence and Uniqueness Theorem [22, Theorem 1.2.4], the domain of function $f(\cdot,\epsilon)$ can be bounded according to

$$t \leq \inf(T, d/M(\epsilon)) ,$$

where

$$M(\epsilon) \geq \left| \sum_{k=1}^{K} \epsilon^k F_k(t,f) + \epsilon^{K+1}\mathrm{Rest}(t,\epsilon,u) \right| .$$

Obviously, we can enable $\inf(T, d/M(\epsilon)) = T$ by taking a sufficiently large $d/M(\epsilon)$ as ϵ is sufficiently small.

On the one hand, based on the continuity of the solution $f(t,\epsilon)$ and the compactness of the set $[0,T] \times [-\epsilon_0,\epsilon_0]$, there exists an image set $\mathbb{K}$ such that $f(t,\epsilon) \in \mathbb{K}$, that is, $f(t,\epsilon) : [0,T] \times [-\epsilon_0,\epsilon_0] \to \mathbb{K}$. Informally, we can re-formulate the solution function f as $f(t,\epsilon,u) : [0,T] \times [-\epsilon_0,\epsilon_0] \times \mathbb{K} \to \mathbb{K}$ throughout this proof. On the other hand, based on the continuity of the function Rest, we have

$$|\mathrm{Rest}(s,\epsilon,f)| \leq \max\{|\mathrm{Rest}(t,\epsilon,u)|\} = N .$$

for all $(t,\epsilon,u) \in [0,T] \times [-\epsilon_0,\epsilon_0] \times \mathbb{K}$. Further, we have

$$\left| \int_0^T \mathrm{Rest}(s,\epsilon,f)\,\mathrm{d}s \right| \leq \int_0^T |\mathrm{Rest}(s,\epsilon,u)|\,\mathrm{d}s = TN ,$$

which implies that

$$\int_0^T \mathrm{Rest}(s,\epsilon,f)\,\mathrm{d}s = \mathcal{O}(1) . \tag{7}$$

Provided

$$\epsilon g(u,\epsilon) = f(T,\epsilon,u) - u ,$$

then from Lemma 2 and Eq. (7), we have

$$g(u,\epsilon) = \sum_{k=1}^{K} \epsilon^{k-1} G_k(T,u) + \epsilon^K \mathcal{O}(1) ,$$

where $u \in \bar{U}/\tilde{u}$. It is self-evident that when $T = 2\pi$, it holds that $U \subset \mathbb{K}$ is a neighborhood of the center point $\tilde{u}$ satisfying that (1) $G_k(t,u) \neq 0$ for all $u \in \bar{U}/\tilde{u}$ and (2) the Brouwer degree $d_B(G_k,U,\tilde{u}) \neq 0$ [14]. Hence, we have

$$g(u,\epsilon) = \sum_{k=r}^{K} \epsilon^{k-1} G_k(2\pi,u) + \epsilon^K \mathcal{O}(1) ,$$

for the case that $G_l \equiv 0$ for $l \in [r-1]$ and $r \in [k]$ but $G_r \neq 0$. Thus, it is self-evident that $f(t,\epsilon)$ is an 2π-periodic solution if and only if $g(u,\epsilon) = 0$. From [14, Lemma 6], we have

$$d_B(G_r, U, \tilde{u}) = d_B(g(u,\epsilon), U, \tilde{u}) \neq 0 \ ,$$

for $|\epsilon| > 0$ sufficiently small. Further, from [5, Charpter VIII], there exists some $u(\epsilon) \in U$ such that $g(u(\epsilon),\epsilon) = 0$. Therefore, we can conclude that $f(t,\epsilon,u(\epsilon))$ is a periodic solution of system (5), and then, pick up a collection of $u(\epsilon)$ such that $u(\epsilon) \to \tilde{u}$ as $\epsilon \to 0$. This completes this proof. □

3.2. Algorithmic Upper Bound

Zhang et al. [32, Theorem 2] has provided a upper bound, i.e., 2^{n-1} for the bifurcation solutions of Eq. (1), which is obviously a baggy result. However, tightening number of bifurcation solutions of Eq. (1) is a tricky challenge in confronted of dynamical systems, which coincides with the second part of Hilbert's 16$^{\text{th}}$ problem [7]. In the near future, it has not been possible to find uniform upper bounds for $H(n)$, referring to the knowledge of Sanders et al. [23] and Libre et al. [13] for a modern investigation of this problem.

This subsection explores a calculable approach for computing the upper bound rather than finding the explicit one. From Lemma 2, we specify the general solution of which the K^{th} component $G_k(t,\epsilon)$ is of a recursive form for $k \in [K]$. So it is feasible to simulate $G_k(t,\epsilon)$ in an algorithmic way. Further, we can obtain the bifurcation solution $f(t,\epsilon)$, and then, the upper bound of $H(n)$ can be calculable in our hands.

Inspired by this recognition, we present the algorithm for calculating the K^{th} components of bifurcation solutions. The procedure comprises four steps:

- **Step One**. Simulate Eq. (6) with K^{th} order and ϵ from the perturbed system (5) in **Procedure 1-3**;
- **Step Two**. Formulate the exact formula of $F_k(t,\epsilon)$ for $k \in [K]$ in **Procedure 4**.
- **Step Three**. Compute the approximation to $G_k(t,\epsilon)$ relative to $\partial^r F_k/\partial\epsilon^r$ and $\text{Rest}(t,u,\epsilon)$ for $k \in [K]$ and $r \in [k]$ in **Procedure 5-12**.
- **Step Four**. Calculate the upper bound of $H(n)$ using the number of positive simple center points of $G_k(t,\epsilon)$ for $k \in [K]$ in **Procedure 13**.

Notice that according to Eqs. (14) and (15), the numerator for each K^{th} component $G_k(t,\epsilon)$ is a polynomial function with degree $\lfloor N * T = 2 * 2\pi = 4\pi \rfloor$. Drawing on the experience of Huang et al. [8], we can greatly improve the calculation speed by updating Eq. (6) along with forcing $G_1 \equiv G_2 \equiv \cdots \equiv G_{K-1} \equiv 0$.

Algorithm 1 Algorithmic Calculation for Upper Bounds.

Input: $N = 2$ (i and i'), n, m, K, apposite perturbation ϵ

Output: K^{th} component $G_k(t, \epsilon)$ for $k \in [K]$,

bifurcation solution $f(t, \epsilon)$

Procedure:

1: Generate Poly$(\boldsymbol{u}(t); n)$ and Poly$(\boldsymbol{u}(t); K)$ in a feed-forward way;

2: Compute $\frac{\mathrm{d}u_i(t)}{\mathrm{d}t}$ and $\frac{\mathrm{d}u_{i'}(t)}{\mathrm{d}t}$ from Eq. (5);

3: Convert the perturbed system (5) into $\frac{\partial f(t,\epsilon)}{\partial t}$ in Eq. (6);

4: Compute functions F_k for $k = 0$ or $k \in [K]$ from the Taylor expansion (refer to Eq. (11)) of $\frac{\partial f(t,\epsilon)}{\partial t}$ in Eq. (6);

5: Let $\delta F = 0$;

6: **for** k **from** 1 **to** $K - 1$ **do**

7: **for** r **from** 1 **to** k **do**

8: $\delta F \leftarrow \delta F + \frac{\partial^r F_k(t, f(t,\epsilon,u))}{\partial \epsilon^r}\big|_{\epsilon=0}$;

9: Re-compute F_k provided δF from Eq. (11);

10: Compute G_k provided δF and F_k from Eq. (13);

11: $\delta R \leftarrow \int_0^{2\pi} \mathrm{Rest}(s, u, \epsilon)\, \mathrm{d}s$ by sampling $u \in \mathbf{K}$;

12: Compute f provided G_k, F_k, and δR from Eq. (16);

13: **return** f.

3.3. Provable Lower Bound

Now, we present the second theorem as follows.

Theorem 2 *Let $H(n)$ denote the maximum possible number of bifurcation solutions of dynamical systems in Eq.* (1) *with n-order polynomial bifurcation fields in Eq.* (3)*. Then $H(n)$ is calculable, and we have*

$$H(n) \in \Omega((n+1)^2 \ln(n+1)) \ .$$

Theorem 2 shows the lower bound of the maximum number of bifurcation solutions of the dynamical system led by Eq. (1). This theoretical result provides solid support for the bifurcation solutions proposed by [32, Theorem 2], which only sheds the "at most" (upper) bound for spiking neural networks.

The complete proof of Theorem 2 can be accessed in Appendix B, and its proof idea can be summarized as follows. According to Section 3, the lower bound of $H(n)$ coincides with the maximum possible number of limit cycles of the dynamical system led by Eq. (1). Hence, a key intuition of bounding $H(n)$ is reformulate the lower bound into a recursive formation. It is observed that equipped with n-order polynomial bifurcation fields in Eq. (3), Eq. (1) leads to a Hamiltonian system with perturbation ϵ,

$$\begin{cases} \dfrac{\mathrm{d}u_k(t)}{\mathrm{d}t} = -\dfrac{\partial\mathcal{H}(u_k, u_{k'})}{\partial u_{k'}} + \epsilon f_\epsilon(u_k, u_{k'}) \ , \\ \dfrac{\mathrm{d}u_{k'}(t)}{\mathrm{d}t} = \dfrac{\partial\mathcal{H}(u_k, u_{k'})}{\partial u_k} + \epsilon g_\epsilon(u_k, u_{k'}) \ , \end{cases} \tag{8}$$

where $\mathcal{H}(u_k, u_{k'}) = \mathrm{Poly}(u_k)^2 + \mathrm{Poly}(u_{k'})^2$ indicates the Lyapunov energy function [32], $0 < \epsilon < \infty$ denotes the noise amplitude, $f(u_k, \cdot)$ and $g(u_k, \cdot)$ are two polynomial functions of degree $2^n - 1$ for $k' \in [N]$. Thus, the lower bound of $H(n)$ meets the following recursive formation

$$H(n+1) = 4H(n) + (2^n - 2)^2 + (2^n - 1)^2 \ .$$

With straightforward computation, we can obtain that there exists a constant C such that

$$H(n) \geq C(n+1)^2 \ln(n+1) \ .$$

We begin our proof with some useful lemmas.

Lemma 3 *The PIRATE model in Eq.* (1) *with n-order polynomial bifurcation fields coincides with a Hamiltonian system of degree free*$(n) = 2^n - 1$.

Lemma 4 *The Hamiltonian system led by Eq.* (1) *with n-order polynomial bifurcation fields has at least $P(n)$ limit cycles in which*

$$P(n+1) = P(n) + (\mathit{free}(n) - 1)^2 + \mathit{free}(n)^2 \ . \tag{9}$$

Lemma 4 provides a recursive sequence $P(n)$ relative to the freedom degree $2^n - 1$ of system (8), which contributes to the lower bound of $H(n)$.

Finishing the Proof of Theorem 2. From Lemma 4, the recursive formation of sequence $P(n)$ indicates the lower bound of $H(n)$. Let $P(n) = 2^{2n}Q(n)$. Then Eq. (9) becomes

$$Q(n+1) = Q(n) + \frac{1}{2} - \frac{3}{2^{n+1}} + \frac{5}{4^{n+1}} \ .$$

Further, we have $Q(2) = 3/16$ and

$$\begin{aligned} Q(n) &= Q(n-1) + \frac{1}{2} - \frac{3}{2^n} + \frac{5}{4^n} \\ &= Q(2) + \frac{n-2}{2} - \frac{3}{4}(1 - 2^{-n+2}) + \frac{5}{48}(1 - 4^{-(n+2)}) \\ &= Q(2) + \frac{n}{2} - \left(\frac{16}{5}\right)^{-n} - \frac{5 \cdot 4^{-n}}{3} - \frac{79}{48} \\ &= \frac{n}{2} - \left(\frac{16}{5}\right)^{-n} - \frac{5 \cdot 4^{-n}}{3} - \frac{35}{24} \end{aligned}$$

for $n \geq 3$. Since $H(n) \geq P(n) \geq 4^n Q(n)$ and the degree of Eq. (8) is $\mathrm{free}(n) = 2^{n-1}$, we have

$$H(2^n - 1) \geq 4^{n-1}\left(2n - \frac{35}{6}\right) + \left(\frac{16}{5}\right)^n - \frac{5}{3} \,.$$

Re-substituting the variable n, the above inequality becomes

$$H(n) \geq \frac{(n+1)^2}{2}\left(\log_2(n+1) - 3\right) + 3n \,.$$

Therefore, there exists some constant C such that

$$H(n) \geq C(n+1)^2 \ln(n+1) \,.$$

It is observed that

$$H(2) \geq 0, \quad H(3) \geq 1, \quad H(7) \geq 25, \quad H(15) \geq 185, \quad \text{and} \quad H(31) \geq 1262 \,.$$

This completes this proof. $\square$

4. Experiments

This section conducts experiments to evaluate the functional performance of PIRATE, where Subsection 4.1 verifies the explicit bounds in Theorem 2 and Subsection 4.2 demonstrates the comparative performance on Image Recognition.

4.1. Simulation Experiments on Algorithmic Upper Bounds

This subsection illustrate the algorithmic upper bound of $H(n)$ in Subsection 3.2. For convenience, we consider a simple case of $N = 2$, $n = 2$, $m = 3$, and $K = 5$, as follows:

$$\begin{cases} \dfrac{\mathrm{d}u_1(t)}{\mathrm{d}t} = -\dfrac{u_1(t)}{\tau_m} + u_1(t)^2 u_2(t) + \epsilon\, \mathrm{Poly}^1(\boldsymbol{u}(t); m) \,, \\ \dfrac{\mathrm{d}u_2(t)}{\mathrm{d}t} = -\dfrac{u_2(t)}{\tau_m} + u_1(t) u_2(t)^2 + \epsilon\, \mathrm{Poly}^2(\boldsymbol{u}(t); m) \,, \end{cases} \tag{10}$$

where

$$\begin{aligned} \mathrm{Poly}^i(\boldsymbol{u}; 3) = \ & \beta^i_{k,1} u_1 + \beta^i_{k,2} u_2 \\ & + \beta^i_{k,3} u_1^2 + \beta^i_{k,4} u_1 u_2 + \beta^i_{k,5} u_2^2 \\ & + \beta^i_{k,6} u_1^3 + \beta^i_{k,7} u_1^2 u_2 + \beta^i_{k,8} u_1 u_2^2 + \beta^i_{k,9} u_2^3 \,, \end{aligned}$$

for $i \in [N = 2]$ and $k \in [K = 5]$. Obviously, it is known as a cubic system with $m = 3^{\text{th}}$ polynomial perturbations. Here, we employ $K = 5^{\text{th}}$ component to estimate the upper bounds of $H(n)$, and have the following conclusion.

Corollary 3 *The maximum number of bifurcation solutions of the concerned system* (10) *is at most 3, which can be calculated by the* 5^{th} *components.*

Corollary 3 shows that using Algorithm 1 with $m = 3$ and $K = 5$ enables the upper bound of $H(n)$ with $N = 2$ to be calculable. Combining with the lower bound from Theorem 2, we can conclude that

$$0 \leq H(2) \leq 3 \ ,$$

which is a tighter bound than that of Zhang et al. [32]. The detailed materials can be accessed in Appendix C.

4.2. Comparative Performance on Image Recognition

This subsection aims to demonstrate the performance of the PIRATE model from an experimental point of view. Hence, we compare PIRATE with some typical SNNs on several image recognition tasks to verify its performance, especially led by an increasing n and $N > 2$. Limited to space, we move the training procedure of PIRATE to Appendix D.

The conducted data sets comprise: (1) The MNIST handwritten digit data set[1] comprises a training set of 60,000 examples and a testing set of 10,000 examples in 10 classes, where each example is centered in a 28×28 image. Using Poisson encoding, we produce a list of spike signals with a formation of $784 \times T$ binary matrices, where T denotes the encoding length and each row represents a spike sequence at each pixel. (2) The Neuromorphic-MNIST (N-MNIST) data set[2] [18] is a spiking version of the original frame-based MNIST data set. Each example in N-MNIST was converted into a spike sequence by mounting the ATIS sensor on a motorized pan-tilt unit and having the sensor move while it views MNIST examples on an LCD monitor. It consists of the same 60,000 training and 10,000 testing samples as the original MNIST data set, and is captured at the same visual scale as the original MNIST data set (28×28 pixels) with both "on" and "off" spikes. (3) The Fashion-MNIST data set[3] consists of a training set of 60,000 examples and a testing set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes. (4) The Extended MNIST-Balanced (EMNIST) [4] data set is an extension of MNIST to handwritten, which contains handwritten upper & lower case letters of the English alphabet in addition to the digits, and comprises 112,800 training and 18,800 testing samples for 47 classes.

The pre-processing steps for these experiments are the same as those used by [19, 32]. Each static image of (1) MNIST, (3) Fashion-MNIST, and (4) EMNIST is transformed as a spike sequence using Poisson Encoding, while each instance in N-MNIST was encoded by a Dynamic Audio / Vision Sensor (DAS / DVS). For these

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.garrickorchard.com/datasets/n-mnist

[3] https://www.kaggle.com/zalando-research/fashionmnist

Table 1: Parameter Setting of PIRATE on Image Recognition.

Parameters Value	MNIST	N-MNIST	Fashion-MNIST	EMNIST
Batch Size	32	32	32	64
Encoding Length T	300	300	400	400
Expect Spike Count (True)	100	80	100	140
Expect Spike Count (False)	10	5	10	0
Firing Threshold	10	10	10	10
Learning Rate η	0.01	0.01	0.01	0.01
Maximum Time	300 ms	300 ms	400 ms	400 ms
Membrane Time τ_m	0.2	0.1	0.2	0.2
Refractory Period	16 ms	16 ms	16 ms	16 ms
Time Constant of Synapse τ_s	8 ms	8 ms	8 ms	8 ms
Time Step τ_s	1 ms	1 ms	1 ms	1 ms

image classification tasks, we set 10 (47 for EMNIST) output spiking neurons corresponding to the classification labels. The output label of SNNs is the one with the greatest spike count. Notice that all SNN models are without any convolution structure, and the self-connection element λ is randomly sampled from $[0, 1]$ with bias $= 1/4$. Table 1 lists the typical configuration values in PIRATE. All the experiments were made in Python, running on Intel Core i7-6500U CPU @2.50 GHz.

Table 2 lists the comparative performance (accuracy) and configurations (setting) of the contenders and PIRATE on four digit data sets. Notice that BSNN proposed by [32] is a special case of the proposed PIRATE model with ($n = 1$). It is observed that the self-connection SNN models (PIRATE with $n = 1, 2$) perform best against other competing approaches, achieving very superior testing accuracy (i.e., more than 99.00% on MNIST, 99.20% on NMNIST, 91.00% on Fashion-MNIST, and 87.50 % on E-MNIST). It is a laudable result, which shows the power and potential of self-connection in SNNs. On the other hand, we observe that the accuracy of PIRATE with $n = 3$ is lower than other models yet with a larger variance, which hints a conspicuous symbol for over-fitting. Thus, we conjecture that unilaterally increasing the higher-order terms on the self-connection mutual promotion will not achieve an ideal improvement on accuracy of SNNs. Besides, a higher-order $u_k^*(\boldsymbol{\lambda}, t)$ tends to cause evidently larger computation consumption. Therefore, we recommend that the self-connection term $u_k^*(\boldsymbol{\lambda}, t)$ should not exceed order 2 in practical applications.

5. Conclusions, Discussions, and Prospects

In this paper, we present the theoretical understandings of the structural stability of SNNs with polynomial self-connection, that is, PIRATE. We provided three main results to show the existence, algorithmic upper bound, and explicit lower bound of the bifurcation solutions of PIRATE with internal and external perturbations. Numerical experiments conducted on several simulation systems and image recognition tasks demonstrate the above theoretical results and the effectiveness of PIRATE, respectively.

Solid Support for SNNs Theory. Despite an increasing focus on the potential of handling spatio-temporal data, the theoretical understandings of many aspects (e.g., approximation power, computational efficiency, and stability) of SNNs are still cloudy. To the best of our knowledge, few theoretical guarantees on the universality [15, 25], approximation complexity [33], and computational efficiency [28, 3] of SNNs have been provided. This work initiates an in-depth analysis for theoretically investigating the structural stability. In contrast to the Lyapunov stability that considers perturbations of initial conditions for a fixed system and the algorithmic stability that characterizes perturbations of training sets for some learning algorithms, structural stability is a qualitative property that qualifies the behavior of structures under perturbations, dealing with perturbations of the SNN system itself. We succeed in producing provable and algorithmic bounds for qualifying the structural stability of SNNs. Besides, our theoretical results show the effect of self-connection on withstanding perturbations (since using higher-order implementation can be regarded as a Taylor approximation to the complex self-connection function), which disclose the significance and power of self connections for enhancing the stability of SNNs and may shed some insights on developing provable and sound SNNs. Prospectively, our work provides solid support for the theory development of SNNs.

Future Issues. In light of the preceding merits, many issues are worthy of being studied in the future. One important future issue is to explore some practical techniques for scSNNs. Indeed, the current implementation, i.e., the Taylor expansion of the self-connection function from the k-th neuron to the i-th neuron equals to the last spike of neuron k as shown in Eq. (3), inevitably leads to a larger memory consumption when the input spike sequences are high-dimensional and high-frequency. Thus, it is prospective to explore some more practical techniques or modules for scSNNs. An alternative way is to train SNNs with adversarial examples, that is, trick SNNs by providing deceptive input or protect their performance from malicious attack signals. Another important future issue is to develop in-depth theoretical understandings of SNNs, especially explore the theoretical advantages of SNNs with self connections over SNNs without ones from the perspectives of approximation, optimization, and generalization.

Table 2: The comparative performance of the contenders and PIRATE.

Data Sets	**Contenders**	**Accuracy** (%)	**Setting**
MNIST	Deep SNN-BP [12]	98.71	28×28-800-10 ♠
	SNN-EP [17] ♡	97.63	28×28-500-10
	HM2-BP [9]	98.84 ± 0.02	28×28-800-10
	SLAYER [27]	98.39 ± 0.04	28×28-500-500-10
	SNN-L [20]	98.23 ± 0.07	28×28-1000-R28-10 ♢
	BSNN [32]	99.02 ± 0.04	28×28-500-500-10
	PIRATE ($n = 2$)	**99.09** ± **0.07**	28×28-500-500-10
	PIRATE ($n = 3$)	98.53 ± 0.13	28×28-500-500-10
N-MNIST	Deep SNN-BP	98.78	2*28×28-800-10
	SNN-EP	97.74	2*28×28-500-10
	HM2-BP	98.84 ± 0.02	2*28×28-800-10
	SLAYER	98.89 ± 0.06	2*28×28-500-500-10
	SNN-L	98.33 ± 0.11	2*28×28-1000-R28-10
	BSNN	**99.24** ± **0.12**	2*28×28-500-500-10
	PIRATE ($n = 2$)	99.21 ± 0.16	2*28×28-500-500-10
	PIRATE ($n = 3$)	99.07 ± 0.22	2*28×28-500-500-10
Fashion-MNIST	Deep SNN-BP	87.34	28×28-800-10
	ST-RSBP [34]	90.00 ± 0.13	28×28-400-R400-10
	HM2-BP	88.99	28×28-400-400-10
	SLAYER	88.61 ± 0.17	28×28-500-500-10
	SNN-L	89.61 ± 0.09	28×28-1000-R28-10
	BSNN	91.22 ± 0.06	28×28-500-500-10
	PIRATE ($n = 2$)	**91.78** ± **0.12**	28×28-500-500-10
	PIRATE ($n = 3$)	88.56 ± 0.23	28×28-500-500-10
EMNIST	Deep SNN-BP	80.51	28×28-800-47
	eRBP [16]	78.17	28×28-200-200-47
	HM2-BP	84.43 ± 0.10	28×28-400-400-47
	SLAYER	85.73 ± 0.16	28×28-500-500-47
	SNN-L	83.75 ± 0.15	28×28-1000-R28-47
	BSNN	87.51 ± 0.23	28×28-500-500-47
	PIRATE ($n = 2$)	**87.62** ± **0.14**	28×28-500-500-47
	PIRATE ($n = 3$)	84.78 ± 0.34	28×28-500-500-47

♠ : -800- denotes one hidden layer with 800 spiking neurons, while -300-300- is two hidden layers with 300 spiking neurons.

♡ : SNN-EP indicates an implementation for training SNN with equilibrium propagation.

♢ : R28 represents a recurrent layer of 28 spiking neurons.

Supplementary Materials of "Structural Stability of Spiking Neural Networks" (Appendix)

This Appendix provides the supplementary materials for our work "Structural Stability of Spiking Neural Networks", constructed according to the corresponding sections therein. Before that, we review the useful notations as follows.

Let $[N]=\{1,2,\dots,N\}$ be an integer set for $N\in\mathbb{N}^+$, and $|\cdot|_\#$ denotes the number of elements in a collection, e.g., $|[N]|_\#=N$. Given two functions $g,h\colon\mathbb{N}^+\to\mathbb{R}$, we denote by $h=\Theta(g)$ if there exist positive constants c_1,c_2 and n_0 such that $c_1g(n)\le h(n)\le c_2g(n)$ for every $n\ge n_0$; $h=\mathcal{O}(g)$ if there exist positive constants c and n_0 such that $h(n)\le cg(n)$ for every $n\ge n_0$; $h=\Omega(g)$ if there exist positive constants c and n_0 such that $h(n)\ge cg(n)$ for every $n\ge n_0$; $h=o(g)$ if there exist positive constants c and n_0 such that $h(n)<cg(n)$ for every $n\ge n_0$.

Let $\mathcal{C}(\mathbb{K},\mathbb{R})$ be the set of all scalar functions $f:\mathbb{K}\to\mathbb{R}$ continuous on $\mathbb{K}\subset\mathbb{R}^m$. Given $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\dots,\alpha_m)^\top$, $\alpha_i\ge 0$ for $i\in[m]$, $|\boldsymbol{\alpha}|=\sum_{i\in[m]}\alpha_i$, and $\boldsymbol{x}=(x_1,x_2,\dots,x_m)\in K$, we define an n-tuple partial derivative

$$D^{|\boldsymbol{\alpha}|}f(\boldsymbol{x})=\frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}}\frac{\partial^{\alpha_2}}{\partial x^{\alpha_2}}\cdots\frac{\partial^{\alpha_n}}{\partial x^{\alpha_n}}f(\boldsymbol{x})\ .$$

Further, we define

$$\mathcal{C}^l(\mathbb{K},\mathbb{R})=\{f\mid f\in\mathcal{C}(\mathbb{K},\mathbb{R}),\ D^rf\in\mathcal{C}(\mathbb{K},\mathbb{R}),\ r\in[l]\}\ .$$

For $1\le p<\infty$, we define

$$\mathcal{L}^p(\mathbb{K},\mathbb{R})=\left\{f\ \middle|\ f\in\mathcal{C}(\mathbb{K},\mathbb{R}),\ \|f\|_{p,\mathbb{K}}<\infty\right\}\ ,$$

where

$$\|f\|_{p,\mathbb{K}}\overset{\text{def}}{=}\left(\int_{\mathbb{K}}|f(\boldsymbol{x})|^p\,\mathrm{d}\boldsymbol{x}\right)^{1/p}\ .$$

This work considers the Sobolev space $\mathcal{W}_\mu^{l,p}(\mathbb{K},\mathbb{R})$, defined as the collection of all functions $f\in\mathcal{C}^l(\mathbb{K},\mathbb{R})$ and $D^rf\in\mathcal{L}^p(\mathbb{K},\mathbb{R})$ for all $|r|\in[l]$, that is,

$$\|D^rf\|_{p,\mathbb{K}}=\left(\int_{\mathbb{K}}|D^rf(\boldsymbol{x})|^p\,\mathrm{d}\boldsymbol{x}\right)^{1/p}<\infty\ .$$

A. Full Proof for Theorem 1

This section provides the detailed proof for Theorem 1. Lemma 1 establishes based on the perturbation structure between original system (4) and perturbed system (5). Lemma 2 generalizes the Faa di Bruno's Formula in [31],

$$\frac{\mathrm{d}g(f)(t)}{\mathrm{d}t}=\sum_{\mathcal{S}_r}C_{\mathcal{S}}\frac{\mathrm{d}^{h(r)}g(f)(t)}{\mathrm{d}t^{h(r)}}\prod_{l=1}^{r}\left(\frac{\mathrm{d}^lf(t)}{\mathrm{d}t^l}\right)^{\alpha_j}\ ,$$

where $g, f \in \mathcal{C}^K(\mathbb{R}, \mathbb{R})$ and

$$C_{\mathcal{S}} = \frac{r!}{\alpha_1!(\alpha_2!2!^{\alpha_2})\dots(\alpha_r!r!^{\alpha_r})} ,$$

for $\mathcal{S}_r$ denotes the set of all r-tuples of non-negative integers $\{\alpha_j\}_{j\in[r]}$ as noted in Section 2, satisfying

$$\sum_j j\alpha_j = r \quad \text{and} \quad h(r) = \sum_j \alpha_j .$$

Informally, we re-formulate the solution function f as $f(t,\epsilon,u) : [0,T] \times [-\epsilon_0, \epsilon_0] \times \mathbb{K} \to \mathbb{K}$. Hence, we have

$$f(t,\epsilon,u) = u + \int_0^t F_0(s,\tilde{u})\,\mathrm{d}s + \sum_{k=0}^{K} \epsilon^k G_k(t,u) + \epsilon^{K+1}\left[\int_0^t \mathrm{Rest}(s, f(s,\epsilon),\epsilon)\,\mathrm{d}s + \mathcal{O}(1)\right]$$

for $f(0,\epsilon,\tilde{u}) = \tilde{u}$, especially,

$$f(t,\epsilon) = \tilde{u} + \int_0^t F_0(s,\tilde{u})\,\mathrm{d}s + \sum_{k=0}^{K} \epsilon^k G_k(t,\tilde{u}) + \epsilon^{K+1}\left[\int_0^t \mathrm{Rest}(s, f(s,\epsilon),\epsilon)\,\mathrm{d}s + \mathcal{O}(1)\right] .$$

where G_k (for $k \in [K]$) is of recursive form as follows

$$G_k(t,u) = \int_0^t \left[F_k(s,u) + \mathcal{G}\left(D^{h(r)}F_r(s,u), G_r(s,u)\right)\right] \mathrm{d}s ,$$

for all $r \in [k-1]$. The Taylor expansion of $F_k(t, f(t,\epsilon,u))$ for $k \in [K-1]$ around $\epsilon = 0$ is given by

$$F_k(t, f(t,\epsilon,u)) = F_k(t, f(t,\epsilon,0)) + \epsilon^{K-k+1}\mathcal{O}(1) + \sum_{r=1}^{K-k} \frac{\epsilon^r}{r!}\left(\frac{\partial^r F_k(t, f(t,\epsilon,u))}{\partial \epsilon^r}\right)\Bigg|_{\epsilon=0} . \tag{11}$$

From the Faa di Bruno's Formula, we calculate the r-derivatives of $F_k(t, f(t,\epsilon,u))$ for $k \in [K-1]$ in ϵ

$$\frac{\partial^r F_k(t, f(t,\epsilon,u))}{\partial \epsilon^r}\bigg|_{\epsilon=0} = \sum_{\mathcal{S}_r} C_{\mathcal{S}} r! \frac{\mathrm{d}^{h(r)} F_k(t, f(t,\epsilon,u))}{\mathrm{d}t^{h(r)}}\bigg|_{\epsilon=0} \prod_{l=1}^{r} G_k(t,u)^{\alpha_l} , \tag{12}$$

where

$$G_k(t,u) = \frac{1}{r!}\left(\frac{\partial^r f(t,\epsilon,u)}{\partial \epsilon^r}\right)\bigg|_{\epsilon=0} = \int_0^t \left[F_k(s,u) + \mathcal{G}\left(D^{h(r)}F_r(s,u), G_r(s,u)\right)\right] \mathrm{d}s \tag{13}$$

with

$$\mathcal{G} = \sum_{r=1}^{k}\sum_{\mathcal{S}_r} \frac{D^{h(r)}F_{k-r}(s,u)}{\alpha_1!(\alpha_2!2!^{\alpha_2})\dots(\alpha_r!r!^{\alpha_r})} \prod_{l=1}^{r} G_l(s,u)^{\alpha_l}$$

for all $r \in [k-1]$.

Substituting Eq. (12) into Eq. (11), the Taylor expansion of $F_k(t, f(t,\epsilon,u))$ at $\epsilon = 0$ becomes

$$F_k(t, f(t,\epsilon,u)) = F_k(t,u) + \epsilon^{K-k+1}\mathcal{O}(1) + \sum_{r=1}^{K-k}\sum_{\mathcal{S}_r} C_{\mathcal{S}}\epsilon^r \frac{\mathrm{d}^{h(r)} F_k(t, f(t,\epsilon,u))}{\mathrm{d}t^{h(r)}}\bigg|_{\epsilon=0} \prod_{l=1}^{r} G_k(t,u)^{\alpha_j}$$

for $k \in [K-1]$. Moreover, the above result hold for the case $k = 0$. Further, for $k = K$, we have

$$F_k(t, f(t,\epsilon,u)) = F_k(t,u) + \epsilon\mathcal{O}(1) . \tag{14}$$

Thus, we have

$$|F_k(t, f(t,\epsilon,u)) - F_k(t,u)| \le L|f(t,\epsilon,u) - u| = \mathcal{O}(1) , \tag{15}$$

since the set $[0,T] \times [-\epsilon_0, \epsilon_0] \times \bar{U}$ is compact and $F_k(t,u)$ is locally Lipschitz in $\bar{U}$ with scale L.

Summing up the above results, we can conclude that

$$f(t,\epsilon,u) = u + \int_0^t F_0(s,\tilde{u})\,\mathrm{d}s + \sum_{k=0}^{K} \epsilon^k G_k(t,u) + \epsilon^{K+1}\left[\int_0^t \mathrm{Rest}(s, f(s,\epsilon), \epsilon)\,\mathrm{d}s + \mathcal{O}(1)\right] \,, \tag{16}$$

where G_k (for $k \in [K]$) is of recursive form as follows

$$G_k(t,u) = \int_0^t \left[F_k(s,u) + \mathcal{G}\left(D^{h(r)} F_r(s,u), G_r(s,u)\right)\right] \mathrm{d}s \,,$$

for all $r \in [k-1]$, in which

$$\mathcal{G} = \sum_{r=1}^{k} \sum_{\mathcal{S}_r} \frac{D^{h(r)} F_{k-r}(s,u)}{\alpha_1!(\alpha_2! 2!^{\alpha_2}) \ldots (\alpha_r! r!^{\alpha_r})} \prod_{l=1}^{r} G_l(s,u)^{\alpha_l} \,,$$

for $\mathcal{S}_r$ denotes the set of all r-tuples of non-negative integers $\{\alpha_j\}_{j\in[r]}$ as noted in Section 2, satisfying

$$\sum_j j\alpha_j = r \quad \text{and} \quad h(r) = \sum_j \alpha_j \,.$$

This completes this proof. $\square$

B. Full Proof for Theorem 2

This section provides the detailed proof for Theorem 2. The basic theory of the perturbation of planar Hamiltonian systems is well known. In general, we can reload Eq. (8) as

$$\begin{cases} \dfrac{\mathrm{d}u_k(t)}{\mathrm{d}t} = -\dfrac{\partial \mathcal{H}(u_k, u_{k'})}{\partial u_{k'}} + \epsilon f_\epsilon(u_k, u_{k'}) \,, \\ \dfrac{\mathrm{d}u_{k'}(t)}{\mathrm{d}t} = \dfrac{\partial \mathcal{H}(u_k, u_{k'})}{\partial u_k} + \epsilon g_\epsilon(u_k, u_{k'}) \,. \end{cases}$$

We are going to show the degree of system Eq. (8). We starting this proof with an example of $n = 2$

$$\mathcal{H}(u_k, u_k') = (u_k^2 - 1)^2 + (u_k'^2 - 1)^2 \,.$$

Thus, the unperturbed system has 9 critical points, that is, (x, y) for $x, y \in \{-1, 0, 1\}$, of which 5 points are non-degenerate, that is, $(\pm 1, \pm 1)$ and $(0, 0)$. Therefore, we can claim that there is a polynomial f_ϵ of degree 3, which meets the degree result $2^2 - 1 = 3$ of Lemma 3, such that

$$\begin{cases} \dfrac{\mathrm{d}u_k(t)}{\mathrm{d}t} = -\dfrac{\partial \mathcal{H}(u_k, u_{k'})}{\partial u_{k'}} + \epsilon f_\epsilon(u_k, u_{k'}) \\ \dfrac{\mathrm{d}u_{k'}(t)}{\mathrm{d}t} = \dfrac{\partial \mathcal{H}(u_k, u_{k'})}{\partial u_k} \end{cases}$$

has limit cycles around center points $(-1,-1)$, $(0,0)$, or $(1,1)$, if ϵ is sufficiently small but $\epsilon \neq 0$. This claim is self-evident if provided

$$f_\epsilon(u_k, u_k') = \frac{1}{3}(u_k - u_k')^2 - \epsilon(u_k - u_k') \,.$$

Next, it suffices to develop the above result to the case of n via mathematical induction. Then we have the following proposition.

Proposition 1 *The system Eq.* (8) *has non-degenerate center points at the origin and at* $2^n - 2$ *other points on each axis, all of which lie within* $\{(x, y) \mid |x| \leq 2^{n-1}$ *and* $|y| \leq 2^{n-1}\}$.

Based on this proposition, we can conclude that

$$\text{free}(n) = 2^{n-1} + 1 = 2^n - 1 \ .$$

Consider a clear-cut case that

$$\begin{cases} \text{Poly}(u_k; 1) = u_k \\ \text{Poly}(u_k; n) = \text{Poly}(u_k^2 - 2^{n-2}; n-1), \quad \text{for} \quad n \geq 2 \ , \end{cases}$$

then system Eq. (8) induces a singular transformation

$$(u_k, u_k') \mapsto (u_k^2 - 2^{n-2}, u_k'^2 - 2^{n-2})$$

that is,

$$(u_k, u_k') \mapsto \left[u_k^2 - (\text{free}(n) - 1), u_k'^2 - (\text{free}(n) - 1)\right]$$

for $n \geq 2$. Further, it is easy to calculate the recursive sequence of Eq. (9) as follows

$$P(n+1) = P(n) + (\text{free}(n) - 1)^2 + \text{free}(n)^2 \ .$$

This completes this proof. □

C. Full Proof for Corollary 3

We begin our analysis with a recall for the concerned example system. We consider a simple case of $N = 2$, $n = 2$, $m = 3$, and $K = 5$, as follows:

$$\begin{cases} \dfrac{\mathrm{d}u_1(t)}{\mathrm{d}t} = -\dfrac{u_1(t)}{\tau_m} + u_1(t)^2 u_2(t) + \epsilon \, \text{Poly}^1(\boldsymbol{u}(t); m) \ , \\ \dfrac{\mathrm{d}u_2(t)}{\mathrm{d}t} = -\dfrac{u_2(t)}{\tau_m} + u_1(t) u_2(t)^2 + \epsilon \, \text{Poly}^2(\boldsymbol{u}(t); m) \ , \end{cases}$$

where

$$\begin{aligned} \text{Poly}^i(\boldsymbol{u}; 3) = {} & \beta_{k,1}^i u_1 + \beta_{k,2}^i u_2 \\ & + \beta_{k,3}^i u_1^2 + \beta_{k,4}^i u_1 u_2 + \beta_{k,5}^i u_2^2 \\ & + \beta_{k,6}^i u_1^3 + \beta_{k,7}^i u_1^2 u_2 + \beta_{k,8}^i u_1 u_2^2 + \beta_{k,9}^i u_2^3 \ , \end{aligned}$$

for $i \in [N = 2]$ and $k \in [K = 5]$. Obviously, it is known as a cubic system with $m = 3^{\text{th}}$ polynomial perturbations.

Here, we employ $K = 5^{\text{th}}$ component to estimate the upper bounds of $H(n)$. From **Procedure 1-3**, we first convert Eq. (10) into

$$\frac{\partial f(t,\epsilon)}{\partial t} = \sum_{k=0}^{5} \epsilon^k F_k(t,f) + \epsilon^6 \mathrm{Rest}(t,\epsilon,f) \ ,$$

that is,

$$\frac{\partial f(t,\epsilon)}{\partial t} = F_0 + \sum_{k=1}^{5} \epsilon^k F_k(t,f) + \mathcal{O}(\epsilon^6) \ .$$

Next, we provides the detailed calculation paradigms for f provided G_k and F_k ($k \in [K=5]$). For convenience, we consider two cases either $F_0 \equiv 0$ or $F_0 \not\equiv 0$.

Let $\odot$ denote the Hadamard product. We have the sets $\mathcal{S}_r$ for $r \in [K=5]$

$$\begin{cases} \mathcal{S}_1 = \{1\}, \\ \mathcal{S}_2 = \{(0,1),(2,0)\} \\ \mathcal{S}_3 = \{(0,0,1),(1,1,0),(3,0,0)\}, \\ \mathcal{S}_4 = \{(0,0,0,1),(1,0,1,0),(2,1,0,0),(0,2,0,0),(4,0,0,0)\}, \\ \mathcal{S}_5 = \{(0,0,0,0,1),(1,0,0,1,0),(0,1,1,0,0),(2,0,1,0,0),(3,1,0,0,0),(1,2,0,0,0),(5,0,0,0,0)\}. \end{cases}$$

For the case of $F_0 \equiv 0$, we have

$$\begin{aligned}
G_0(u) &= 0 \\
G_1(u) &= \int_0^T F_1(t,u)\,\mathrm{d}t \\
G_2(u) &= \int_0^T F_2(t,u)\,\mathrm{d}s + \frac{\partial F_1}{\partial u}(t,u) y_1(t,u) dt \\
G_3(u) &= \int_0^T \left(F_3(t,u) + \frac{\partial F_2}{\partial u}(t,u) y_1(t,u) \right) \mathrm{d}t + \int_0^T \left(\frac{\partial^2 F_1}{\partial u^2}(t,u) y_1(t,z)^2 + \frac{\partial F_1}{\partial u}(t,u) y_2(t,u) \right) \mathrm{d}t \\
G_4(u) &= \int_0^T \left(F_4(t,u) + \frac{\partial F_3}{\partial u}(t,u) y_1(t,u) \right) \mathrm{d}t + \int_0^T \left(\frac{\partial^2 F_2}{\partial u^2}(t,u) y_1(t,u)^2 + \frac{\partial F_2}{\partial u}(t,u) y_2(t,u) \right) \mathrm{d}t \\
&\quad + \int_0^T \frac{\partial^2 F_1}{\partial u^2}(t,u) y_1(t,u) \odot y_2(t,u)\,\mathrm{d}t + \int_0^T \left(\frac{\partial^3 F_1}{\partial u^3}(t,u) y_1(t,u)^3 + \frac{\partial F_1}{\partial u}(t,u) y_3(t,u) \right) \mathrm{d}t \\
G_5(u) &= \int_0^T \left(F_5(t,u) + \frac{\partial F_4}{\partial u}(t,u) y_1(t,u) \right) \mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^2 F_3}{\partial u^2}(t,u) y_1(t,u)^2 + \frac{\partial F_3}{\partial u}(t,u) y_2(t,u) + \frac{\partial^2 F_2}{\partial u^2}(t,u) y_1(t,u) \odot y_2(t,u) \right) \mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^3 F_2}{\partial u^3}(t,u) y_1(t,u)^3 + \frac{\partial F_2}{\partial u}(t,u) y_3(t,u) + \frac{\partial^2 F_1}{\partial u^2}(t,z) y_1(t,u) \odot y_3(t,z) \right) \mathrm{d}t \\
&\quad + \int_0^T \frac{\partial^2 F_1}{\partial u^2}(t,u) y_2(t,u)^2\,\mathrm{d}t + \int_0^T \frac{\partial^3 F_1}{\partial u^3}(t,u) y_1(t,u)^2 \odot y_2(t,u)\,\mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^4 F_1}{\partial x^4}(t,u) y_1(t,u)^4 + \frac{\partial F_1}{\partial u}(t,u) y_4(t,u) \right) \mathrm{d}t \ ,
\end{aligned}$$

where

$$
\begin{aligned}
y_1(t,u) &= \int_0^s F_1(s,u)\,\mathrm{d}s \\
y_2(t,u) &= \int_0^s F_2(s,u) + \frac{\partial F_1}{\partial u}(s,u)y_1(s,u)\,\mathrm{d}s \\
y_3(t,u) &= \int_0^s \left(F_3(s,u) + \frac{\partial F_2}{\partial u}(s,u)y_1(t,z) + \frac{\partial^2 F_1}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_1}{\partial u}(s,u)y_2(s,u)\right)\mathrm{d}s \\
y_4(t,u) &= \int_0^s \left(F_4(s,u) + \frac{\partial F_3}{\partial x}(s,u)y_1(s,u)\right)\mathrm{d}s + \int_0^s \left(\frac{\partial^2 F_2}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_2}{\partial u}(s,u)y_2(s,u)\right)\mathrm{d}s \\
&\quad + \int_0^s \frac{\partial^2 F_1}{\partial u^2}(s,u)y_1(s,u)\odot y_2(s,u)\,\mathrm{d}s + \int_0^s \left(\frac{\partial^3 F_1}{\partial u^3}(s,u)y_1(s,u)^3 + \frac{\partial F_1}{\partial u}(s,u)y_3(s,u)\right)\mathrm{d}s \\
y_5(t,u) &= \int_0^t \left(F_5(s,u) + \frac{\partial F_4}{\partial u}(s,u)y_1(s,u)\right)d\,\mathrm{d}s \\
&\quad + \int_0^t \left(\frac{\partial^2 F_3}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_3}{\partial u}(s,u)y_2(s,u) + \frac{\partial^2 F_2}{\partial u^2}(s,u)y_1(s,u)\odot y_2(s,u)\right)\mathrm{d}s \\
&\quad + \int_0^t \left(\frac{\partial^3 F_2}{\partial u^3}(s,u)y_1(s,u)^3 + \frac{\partial F_2}{\partial u}(s,u)y_3(s,u) + \frac{\partial^2 F_1}{\partial x^2}(s,u)y_1(s,u)\odot y_3(s,z)\right)\mathrm{d}s \\
&\quad + \int_0^t \frac{\partial^2 F_1}{\partial u^2}(s,u)y_2(s,u)^2\,\mathrm{d}s + y^t\frac{\partial^3 F_1}{\partial u^3}(s,u)y_1(s,u)^2\odot y_2(s,u)\,\mathrm{d}s \\
&\quad + 5\int_0^t \left(\frac{\partial^4 F_1}{\partial u^4}(s,u)y_1(s,u)^4 + \frac{\partial F_1}{\partial u}(s,u)y_4(s,u)\right)\mathrm{d}s\,.
\end{aligned}
$$

For the case of $F_0 \not\equiv 0$, we have

$$
\begin{aligned}
G_0(u) &= \int_0^T F_0(t,u)\,\mathrm{d}t \\
G_1(u) &= \int_0^T F_1(t,u) + \frac{\partial F_0}{\partial u}(t,u)y_1(t,u)\,\mathrm{d}t \\
G_2(u) &= \int_0^T \left(F_2(t,u) + \frac{\partial F_1}{\partial u}(t,u)y_1(t,u) + \frac{\partial^2 F_0}{\partial u^2}(t,u)y_1(t,u)^2 + \frac{\partial F_0}{\partial u}(t,u)y_2(t,u)\right)\mathrm{d}t \\
G_3(u) &= \int_0^T \left(F_3(t,u) + \frac{\partial F_2}{\partial u}(t,u)y_1(t,u) + \frac{\partial^2 F_1}{\partial u^2}(t,u)y_1(t,u)^2 + \frac{\partial F_1}{\partial u}(t,u)y_2(t,u)\right)\mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^2 F_0}{\partial u^2}(t,u)y_1(t,u)\odot y_2(t,u) + \frac{\partial^3 F_0}{\partial u^3}(t,u)y_1(t,u)^3 + \frac{\partial F_0}{\partial u}(t,u)y_3(t,u)\right)\mathrm{d}t \\
G_4(u) &= \int_0^T \left(F_4(t,u) + \frac{\partial F_3}{\partial u}(t,u)y_1(t,u)\right)\mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^2 F_2}{\partial u^2}(t,u)y_1(t,u)^2 + \frac{\partial F_2}{\partial u}(t,u)y_2(t,u)\right)\mathrm{d}t \\
&\quad + \int_0^T \frac{\partial^2 F_1}{\partial x^2}(t,z)y_1(t,z)\odot y_2(t,z)dt \\
&\quad + \int_0^T \left(\frac{\partial^3 F_1}{\partial u^3}(t,u)y_1(t,u)^3 + \frac{\partial F_1}{\partial u}(t,u)y_3(t,u) + \frac{\partial^2 F_0}{\partial u^2}(t,u)y_1(t,u)\odot y_3(t,u)\right)\mathrm{d}t \\
&\quad + \int_0^T \frac{\partial^2 F_0}{\partial u^2}(t,u)y_2(t,u)^2\,\mathrm{d}t + \int_0^T \frac{\partial^3 F_0}{\partial u^3}(t,u)y_1(t,u)^2\odot y_2(t,u)\,\mathrm{d}t \\
&\quad + \int_0^T \left(\frac{\partial^4 F_0}{\partial u^4}(t,u)y_1(t,u)^4 + \frac{\partial F_0}{\partial u}(t,u)y_4(t,u)\right)\mathrm{d}t\,,
\end{aligned}
$$

where

$$
\begin{aligned}
y_1(t,u) &= \int_0^s F_1(s,u) + \frac{\partial F_0}{\partial x}(s,u) y_1(s,u)\,\mathrm{d}t \\
y_2(t,u) &= \int_0^s \left(2F_2(s,u) + \frac{\partial F_1}{\partial u}(s,u)y_1(s,u) + \frac{\partial^2 F_0}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_0}{\partial u}(s,u)y_2(s,u)\right)\mathrm{d}t \\
y_3(t,u) &= \int_0^s \left(F_3(s,u) + \frac{\partial F_2}{\partial u}(s,u)y_1(s,u) + \frac{\partial^2 F_1}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_1}{\partial u}(s,u)y_2(s,u)\right)\mathrm{d}t \\
&\quad + \int_0^s \left(\frac{\partial^2 F_0}{\partial u^2}(s,u)y_1(s,u)\odot y_2(s,u) + \frac{\partial^3 F_0}{\partial u^3}(s,u)y_1(s,u)^3 + \frac{\partial F_0}{\partial u}(s,u)y_3(s,u)\right)\mathrm{d}t \\
y_4(t,u) &= \int_0^s \left(F_4(s,u) + \frac{\partial F_3}{\partial u}(s,u)y_1(s,u)\right)\mathrm{d}t \\
&\quad + \int_0^s \left(\frac{\partial^2 F_2}{\partial u^2}(s,u)y_1(s,u)^2 + \frac{\partial F_2}{\partial u}(s,u)y_2(s,u)\right)\mathrm{d}t \\
&\quad + \int_0^s \frac{\partial^2 F_1}{\partial u^2}(s,u)y_1(s,u)\odot y_2(s,u)\,\mathrm{d}t \\
&\quad + \int_0^s \left(\frac{\partial^3 F_1}{\partial u^3}(s,u)y_1(s,u)^3 + \frac{\partial F_1}{\partial u}(s,u)y_3(s,u) + \frac{\partial^2 F_0}{\partial u^2}(s,u)y_1(s,u)\odot y_3(s,u)\right)\mathrm{d}t \\
&\quad + \int_0^t \frac{\partial^2 F_0}{\partial u^2}(s,u)y_2(s,u)^2\,\mathrm{d}t + \int_0^t \frac{\partial^3 F_0}{\partial u^3}(s,u)y_1(s,u)^2\odot y_2(s,u)\,\mathrm{d}t \\
&\quad + \int_0^t \left(\frac{\partial^4 F_0}{\partial u^4}(s,u)y_1(s,u)^4 + \frac{\partial F_0}{\partial u}(s,u)y_4(s,u)\right)\mathrm{d}t\,.
\end{aligned}
$$

Recall the concerned example system (10), we have

$$
F_1(t,u) = u\left(\beta^1_{1,2} + \beta^2_{1,1}\right)\sin t\cos t + u\left(-\beta^1_{1,1} + \beta^2_{1,2}\right)(\sin t)^2 + u\beta^1_{1,1}\,,
$$

and thus,

$$
G_1(u) = \pi u\left(\beta^1_{1,1} + \beta^2_{1,2}\right)\,.
$$

It is observed that the 1st component $G_1(u)$ has no positive center points, and thus, provides no information about the bifurcation solutions once adding perturbations. Further, it is necessary to compute the higher-order component. From **Procedure 4-12**, we have

$$
\begin{aligned}
G_2(u) &= \frac{\pi u}{2}\left(\pi(\beta^1_{1,1})^2 + 2\pi\beta^1_{1,1}\beta^2_{1,2} + \pi(\beta^2_{1,2})^2 + \beta^1_{1,1}\beta^1_{1,2} - \beta^1_{1,1}\beta^2_{1,1} + \beta^1_{1,2}\beta^2_{1,2} - \beta^2_{1,1}\beta^2_{1,2} + 2\beta^1_{2,1} + 2\beta^2_{2,2}\right) \\
G_3(u) &= \frac{1}{4}\pi u\left[\left(\beta^1_{1,1} + 3\beta^1_{1,6} + \beta^1_{1,8} + \beta_{1,7} + 3\beta^2_{1,9}\right)u^2 + 4\left(\beta^1_{3,1} + \beta^2_{3,2}\right)\right] \quad \text{with} \quad \beta^2_{2,2} = -\beta^1_{2,1} \\
G_4(u) &= \frac{1}{4}\pi u\left[C_1 u^2 + 4\left(\beta^1_{4,1} + \beta^2_{4,2}\right)\right] \\
&\qquad \text{with} \quad \beta^2_{1,7} \leftarrow \beta^1_{1,1} + 3\beta^1_{1,6} + \beta^1_{1,8} + \beta_{1,7} + 3\beta^2_{1,9}\beta^2_{1,7} \quad \text{and} \quad \beta^2_{3,2} \leftarrow -\beta^1_{3,1} - \beta^2_{3,2} + \beta^2_{3,2} \\
G_5(u) &= \frac{1}{4}\pi u\left[\left(2\beta^1_{1,1} + 2\beta^1_{1,6} + \beta^1_{1,8} + \beta^2_{1,9}\right)u^4 + C_2 u^2 + 4\left(\beta^1_{5,1} + \beta^2_{5,2}\right)\right] \\
&\qquad \text{with} \quad \beta^2_{2,7} \leftarrow -C_1 + \beta^2_{2,7} \quad \text{and} \quad \beta^2_{4,2} \leftarrow -\beta^1_{3,1} - \beta^2_{3,2} + \beta^2_{4,2}\,,
\end{aligned}
$$

where

$$
\begin{aligned}
C_1 = {} & 4\beta_{1,1}^1\beta_{1,2}^1 + 2\beta_{1,1}^1\beta_{1,7}^1 + 2\beta_{1,1}^1\beta_{1,8}^2 + \beta_{1,2}^1\beta_{1,8}^1 + 3\beta_{1,2}^1\beta_{1,9}^2 + \beta_{1,3}^1\beta_{1,4}^1 - 2\beta_{1,3}^1\beta_{1,3}^2 + \beta_{1,4}^1\beta_{1,5}^1 + 2\beta_{1,5}^1\beta_{1,5}^2 \\
& + \beta_{1,8}^1\beta_{1,1}^2 + 3\beta_{1,1}^1\beta_{1,9}^2 - \beta_{1,3}^2\beta_{1,4}^2 - \beta_{1,4}^2\beta_{1,5}^2 + 4\beta_{2,1}^1 + 3\beta_{2,6}^1 + \beta_{2,8}^1 + \beta_{2,7}^2 + 3\beta_{2,9}^2
\end{aligned}
$$

$$
\begin{aligned}
C_2 = {} & 4\beta_{1,1}^1(\beta_{1,2}^1)^2 + 2\beta_{1,1}^1\beta_{1,2}^1\beta_{1,7}^1 + 2\beta_{1,1}^1\beta_{1,2}^1\beta_{1,8}^2 + 2\beta_{1,1}^1(\beta_{1,3}^1)^2 + 2\beta_{1,1}^1\beta_{1,3}^1\beta_{1,5}^1 - \beta_{1,1}^1\beta_{1,3}^1\beta_{1,4}^2 + \beta_{1,1}^1(\beta_{1,4}^1)^2 \\
& - \beta_{1,1}^1\beta_{1,4}^1\beta_{1,3}^2 + \beta_{1,1}^1\beta_{1,4}^1\beta_{1,5}^2 + \beta_{1,1}^1\beta_{1,5}^1\beta_{1,4}^2 - 2\beta_{1,1}^1\beta_{1,3}^2\beta_{1,5}^2 - \beta_{1,1}^1(\beta_{1,4}^2)^2 - 2\beta_{1,1}^1(\beta_{1,5}^2)^2 + (\beta^1)_{1,2}^2\beta_{1,8}^1 \\
& + 3(\beta^1)_{1,2}^2\beta_{1,9}^2 + \beta_{1,2}^1\beta_{1,3}^1\beta_{1,4}^1 + 2\beta_{1,2}^1\beta_{1,4}^1\beta_{1,5}^1 + 4\beta_{1,2}^1\beta_{1,5}^1\beta_{1,5}^2 + \beta_{1,2}^1\beta_{1,8}^1\beta_{1,1}^2 + 3\beta_{1,2}^1\beta_{1,1}^2\beta_{1,9}^2 - \beta_{1,2}^1\beta_{1,4}^2\beta_{1,5}^2 \\
& + 2\beta_{1,3}^1\beta_{1,1}^2\beta_{1,3}^2 + \beta_{1,4}^1\beta_{1,5}^1\beta_{1,1}^2 + 2\beta_{1,5}^1\beta_{1,1}^2\beta_{1,5}^2 + \beta_{1,1}^2\beta_{1,3}^2\beta_{1,4}^2 + 4\beta_{1,1}^1\beta_{2,2}^1 + 2\beta_{1,1}^1\beta_{2,7}^1 + 2\beta_{1,1}^1\beta_{2,8}^2 \\
& + 4\beta_{1,2}^1\beta_{2,1}^1 + \beta_{1,2}^1\beta_{2,8}^1 + 3\beta_{1,2}^1\beta_{2,9}^2 + \beta_{1,3}^1\beta_{2,4}^1 - 2\beta_{1,3}^1\beta_{2,3}^2 + \beta_{1,4}^1\beta_{2,3}^1 + \beta_{1,4}^1\beta_{2,5}^1 + \beta_{1,5}^1\beta_{2,4}^1 + 2\beta_{1,5}^1\beta_{2,5}^2 \\
& + 2\beta_{1,7}^1\beta_{2,1}^1 + \beta_{1,8}^1\beta_{2,2}^1 + \beta_{1,8}^1\beta_{2,1}^2 + 2\beta_{2,1}^1\beta_{1,8}^2 + 3\beta_{2,2}^1\beta_{1,9}^2 - 2\beta_{2,3}^1\beta_{1,3}^2 + 2\beta_{2,5}^1\beta_{1,5}^2 + \beta_{2,8}^1\beta_{1,1}^2 + 3\beta_{1,1}^2\beta_{2,9}^2 \\
& - \beta_{1,3}^2\beta_{2,4}^2 - \beta_{1,4}^2\beta_{2,3}^2 - \beta_{1,4}^2\beta_{2,5}^2 - \beta_{1,5}^2\beta_{2,4}^2 + 3\beta_{1,9}^2\beta_{2,1}^2 + 4\beta_{3,1}^1 + 3\beta_{3,6}^1 + \beta_{3,8}^1 + \beta_{3,7}^2 + 3\beta_{3,9}^2 \ .
\end{aligned}
$$

It is observed that the 5^{th} component $G_5(u)$ has at most three positive center points, which provides support for the existence of the upper bound of $H(n)$ in Corollary 3. This completes this proof. □

D. Implementation for PIRATE

Altering to the thought line of bifurcation spiking neural networks [32], we here provide a concrete scheme for implementing PIRATE. This work considers a feed-forward PIRATE with M pre-synaptic input channels and N-dimensional output spiking neurons, and approximate the mutual promotion from the i^{th} neuron to the k^{th} neuron using the last spike of neuron i, noted as $S_i(t_i')$, where $t_i' = \max\{s \mid u_i(s) = u_{firing}, s < t_i\}$. Formally, for the k-th neuron, we have

$$
\frac{\mathrm{d}u_k(t)}{\mathrm{d}t} = -\frac{u_k(t)}{\tau_m} + \text{Poly}(\boldsymbol{u}(t); n, \boldsymbol{\lambda}) + \frac{R}{\tau_m}\sum_{j=1}^{M} \mathbf{W}_{kj}\mathbf{I}_j(t) \ , \tag{17}
$$

which has two types of learnable parameters, i.e., self-connection weights $\boldsymbol{\lambda}$ and connection weights $\mathbf{W}$.

Akin to the spike response model scheme [6], Eq. (17) has a closed-form solution

$$
u_k(t) = \int_{t'}^{t} \exp\left(\frac{t' - s}{\tau_m}\right) Q_k(s)\,\mathrm{d}s \tag{18}
$$

with

$$
Q_k(t) = \sum_{j=1}^{M} \mathbf{W}_{kj}\mathbf{I}_j(t) + \text{Poly}(\boldsymbol{S}(t'); n, \boldsymbol{\lambda}) \ ,
$$

where the vector $\boldsymbol{S}(t') = (S_1(t_1'), \ldots, S_N(t_N'))$ except for S_k, and t' denotes the last firing time $t' = \max\{s \mid \boldsymbol{u}_i(s) = u_{firing},\ s < t\}$ for a pre-given firing threshold $u_{firing} > 0$. Finally, the generated spike is transmitted to the next neuron via the spike excitation function $f_e : u \to S$.

Provided supervised signals, the PIRATE model can be optimized via error back-propagation. In general, we formulate the input (spike) sequence to a spiking neuron as

$$\mathbf{I}_j(t) = \sum_{firing} \delta_j \left(t - t_j^{firing} \right) ,$$

where t_j^{firing} is the spike time of the j^{th} input and $\delta(t)$ is a corresponding Dirac-delta function. Summing up the loss of the k^{th} target supervised signal $\hat{S}_k(t)$ related to $S_k(t)$ in time interval $[0, T]$

$$E_k = \frac{1}{2} \int_0^T E_k(t) \, \mathrm{d}t = \frac{1}{2} \int_0^T \left(S_k(t) - \hat{S}_k(t) \right)^2 \mathrm{d}t \, . \tag{19}$$

So for time t, we have

$$\frac{\partial E_k(t)}{\partial \mathbf{W}_{kj}} = \frac{\partial E_k(t)}{\partial S_k} \frac{\partial S_k}{\partial u_k} \frac{\partial u_k}{\partial \mathbf{W}_{kj}} \, , \tag{20}$$

where the first term is the error back-propagation of the excitatory neurons, the second term is that of the generated spikes with respect to the membrane potential, and the third term denotes the that of basic bifurcation neuron. Plugging Eqs. (18) and (19) into Eq. (20), we have

$$\frac{\partial E_k(t)}{\partial \mathbf{W}_{kj}} = \left(S_k(t) - \hat{S}_k(t) \right) f_e'(u_k) \, \Delta_j^w(t)$$

with

$$\Delta_j^w(t) = \frac{\epsilon_j(t)}{\tau_m} \exp\left(-\frac{t}{\tau_m} \right) \, .$$

However, the derivative of the spike excitation function $f_e'(u)$ is a persistent problem for training SNNs with supervised signals. Recently, there have emerged many seminal approaches for addressing this problem, such as the smoothing derivative via the probability density functions [27] or modified spike excitation functions [33]. Therefore, we obtain the back-propagation pipeline relative to connection weights $\mathbf{W}_{kj}$

$$\nabla_{\mathbf{W}_{kj}} E = \int_0^T \frac{\partial E_k(t)}{\partial \mathbf{W}_{kj}} \, \mathrm{d}t \, .$$

Similarly, the correction with respect to some element λ of $\boldsymbol{\lambda}$ is given by

$$\nabla_\lambda E = \int_0^T \left(S_k(t) - \hat{S}_k(t) \right) f_e'(u_k) \, \Delta^\lambda(t) \, \mathrm{d}t$$

with

$$\Delta^\lambda(t) = \frac{1}{\tau_m} \frac{\partial \mathrm{Poly}(\boldsymbol{S}(t'); n, \boldsymbol{\lambda})}{\partial \lambda} \exp\left(-\frac{t}{\tau_m} \right) \, .$$

Notice that $\partial \mathrm{Poly}(\boldsymbol{S}(t'); n, \boldsymbol{\lambda}) / \partial \lambda$ indicates a polynomial partial derivative, especially we have

$$\nabla_\lambda E = \int_0^T \left(S_k(t) - \hat{S}_k(t) \right) f_e'(u_k) \frac{S_i(t_i')}{\tau_m} \exp\left(-\frac{t}{\tau_m} \right) \mathrm{d}t$$

for $n = 1$. Finally, we can also add a learning rate to help convergence, just like most deep artificial neural networks.

References

[1] Z. Bažant. Structural stability. *International Journal of Solids and Structures*, 37(1-2):55–67, 2000.

[2] K. D. Carlson, M. Richert, N. Dutt, and J. L. Krichmar. Biologically plausible models of homeostasis and stdp: stability and learning in spiking neural networks. In *Proceedings of the 2013 International Joint Conference on Neural Networks*, pages 1–8, 2013.

[3] C.-N. Chou, K.-M. Chung, and C.-J. Lu. On the algorithmic power of spiking neural networks. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference*, pages 26:1–26:20, 2019.

[4] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *Proceesdings of the 2017 International Joint Conference on Neural Networks*, pages 2921–2926, 2017.

[5] J. D. Crawford. Introduction to bifurcation theory. *Reviews of Modern Physics*, 63(4):991, 1991.

[6] W. Gerstner. Time structure of the activity in neural network models. *Physical Review E*, 51(1):738, 1995.

[7] D. Hilbert. Mathematical problems. *Bulletin of the American Mathematical Society*, 8(10):437–479, 1902.

[8] B. Huang and C. Yap. An algorithmic approach to small limit cycles of nonlinear differential systems: The averaging method revisited. *Journal of Symbolic Computation*, 2020.

[9] Y. Jin, W. Zhang, and P. Li. Hybrid macro/micro level backpropagation for training deep spiking neural networks. In *Advances in Neural Information Processing Systems 31*, pages 7005–7015, 2018.

[10] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

[11] R. Kim and T. J. Sejnowski. Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24(1):129–139, 2021.

[12] J. H. Lee, T. Delbruck, and M. Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10:508, 2016.

[13] J. Llibre, R. Moeckel, and C. Simó. *Central configurations, periodic orbits, and Hamiltonian systems*. Springer, 2015.

[14] J. Llibre, D. D. Novaes, and M. A. Teixeira. Higher order averaging theory for finding periodic solutions via brouwer degree. *Nonlinearity*, 27(3):563, 2014.

[15] W. Maass and C. M. Bishop. *Pulsed Neural Networks*. MIT press, 2001.

[16] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in Neuroscience*, 11:324, 2017.

[17] P. O'Connor, E. Gavves, and M. Welling. Training a spiking neural network with equilibrium propagation. In *Proceesdings of the 22nd International Conference on Artificial Intelligence and Statistics*, pages 1516–1523, 2019.

[18] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015.

[19] J. W. Pillow, L. Paninski, V. J. Uzzell, E. P. Simoncelli, and EJ Chichilnisky. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, 2005.

[20] A. Lotfi Rezaabad and S. Vishwanath. Long short-term memory spiking networks and their applications. In *Proceesdings of the 2020 International Conference on Neuromorphic Systems*, pages 1–9, 2020.

[21] K. Roy, A. Jaiswal, and P. Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[22] J. A. Sanders, F. Verhulst, and J. Murdock. *Averaging methods in nonlinear dynamical systems*, volume 59. Springer, 2007.

[23] J. A. Sanders, F. Verhulst, and J. Murdock. *Averaging methods in nonlinear dynamical systems*, volume 59. Springer, 2007.

[24] B. Schrauwen, M. DHaene, D. Verstraeten, and J. Van Campenhout. Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Networks*, 21(2-3):511–523, 2008.

[25] X. She, S. Dash, and S. Mukhopadhyay. Sequence approximation using feedforward spiking neural network for spatiotemporal learning: Theory and optimization methods. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[26] T. Shimokawa, K. Pakdaman, and S. Sato. Time-scale matching in the response of a leaky integrate-and-fire neuron model to periodic stimulus with additive noise. *Physical Review E*, 59(3):3427, 1999.

[27] S. B. Shrestha and G. Orchard. SLAYER: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems 31*, pages 1419–1428, 2018.

[28] P. T. P. Tang, T.-H. Lin, and M. Davies. Sparse coding by spiking neural networks: Convergence theory and computational results. *arXiv:1705.05475*, 2017.

[29] R. VanRullen, R. Guyonneau, and S. J. Thorpe. Spike times make sense. *Trends in Neurosciences*, 28(1):1–4, 2005.

[30] E. Vasilaki, N. Frémaux, R. Urbanczik, W. Senn, and W. Gerstner. Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail. *PLoS Computational Biology*, 5(12):e1000586, 2009.

[31] E. W. Weisstein. Faa di bruno's formula. *https://mathworld. wolfram. com/*, 2003.

[32] S.-Q. Zhang, Z.-Y. Zhang, and Z.-H. Zhou. Bifurcation spiking neural network. *Journal of Machine Learning Research*, 22(253):1–21, 2021.

[33] S.-Q. Zhang and Z.-H. Zhou. Theoretically provable spiking neural networks. *arXiv preprint*, 2022.

[34] W. Zhang and P. Li. Spike-train level backpropagation for training deep recurrent spiking neural networks. In *Advances in Neural Information Processing Systems 32*, pages 7800–7811, 2019.