Neurocomputing 624 (2025) 129392

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A generalized Spiking Locally Competitive Algorithm for multiple optimization problems

Xuexing Du, Zhong-qi K. Tian, Songting Li[1], Douglas Zhou *

School of Mathematical Sciences, MOE-LSC, and Institute of Natural Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai, 200240, China

ARTICLE INFO

Communicated by F. Perez-Pena

Dataset link: https://github.com/XuexingDu/Spiking-LCA

Keywords:
Generalized Spiking LCA
Optimization problems
Signal recovery

ABSTRACT

We introduce a generalized Spiking Locally Competitive Algorithm (LCA) that is biologically plausible and exhibits adaptability to a large variety of neuron models and network connectivity structures. In addition, we provide theoretical evidence demonstrating the algorithm's convergence in optimization problems of signal recovery. Furthermore, our algorithm demonstrates superior performance over traditional optimization methods, such as FISTA, particularly by achieving faster early convergence in practical scenarios including signal denoising, seismic wave detection, and computed tomography reconstruction. Notably, our algorithm is compatible with neuromorphic chips, such as Loihi, facilitating efficient multitasking within the same chip architecture—a capability not present in existing algorithms. These advancements make our generalized Spiking LCA a promising solution for real-world applications, offering significant improvements in execution speed and flexibility for neuromorphic computing systems.

1. Introduction

In diverse areas such as compressed sensing, Bayesian inference, and dictionary learning, the pursuit of sparse representation of signals is critical for enhancing information transfer, reducing complexity, and optimizing resource use [1–6]. These disciplines are often confronted with challenging optimization problems, propelling the advancement of efficient solutions. Traditional methods like gradient descent and greedy algorithms have been effective across a variety of optimization challenges [7–9]. However, their efficiency diminishes in the context of large-scale problems, leading to significant computational demands and resource consumption. This limitation has prompted researchers to explore alternative strategies that can more effectively manage large-scale issues.

An intriguing approach draws inspiration from the human brain, recognized for its exceptional energy efficiency and adaptability. Studies on the primary visual cortex indicate that sensory neurons can encode natural stimuli, such as visual images, with impressive efficiency through sparse coding [10–16]. This has led to the development of neural network models aimed at solving optimization problems in a more energy-efficient manner [17–21]. The Spiking Locally Competitive Algorithm (Spiking LCA), a prominent algorithm in unsupervised learning, stands out in this regard [18]. Yet, Spiking LCA's effectiveness is constrained by its rigidity. Firstly, its reliance on fully inhibitory connections between neurons restricts the scope of optimization problems it can effectively tackle, e.g., the measurement matrix is subject to stringent restrictions. Secondly, distinct neuron network architectures are required for different optimization problems, implying that when we implement the Spiking LCA on neuromorphic chips, network architecture modifications become essential when dealing with varying problems. Therefore, there is an urgent need for an algorithm that can seamlessly adapt to various optimization challenges within a single network framework.

This paper addresses the abovementioned challenges by presenting a new algorithm designed for constructing spiking neural networks, which supports both excitatory and inhibitory neuronal connections. Notably, our approach facilitates the handling of diverse optimization problems within a single network framework by modulating the external input currents to neurons. This adaptability offers significant engineering advantages, particularly the capability to execute multiple tasks within a single chip architecture. Our spiking neural network model is grounded in biologically plausible neuron models, extending from the simple Leaky Integrate-and-Fire (LIF) model to more complex Hodgkin–Huxley type models. We also provide theoretical evidence that our network's firing rates converge to optimal solutions for a variety of optimization problems, such as LASSO and Elastic-Net [22–24]. A notable strength of our proposed model is its considerably faster early convergence when compared to leading optimization algorithms

* Corresponding author.
E-mail addresses: songting@sjtu.edu.cn (S. Li), zdz@sjtu.edu.cn (D. Zhou).
[1] To whom correspondence may be addressed.

https://doi.org/10.1016/j.neucom.2025.129392
Received 16 May 2024; Received in revised form 15 November 2024; Accepted 8 January 2025
Available online 20 January 2025

like FISTA [7]. This enhanced convergence speed enables our model to reach reliable solutions more swiftly, aligning well with practical scenarios where energy efficiency and time are of the essence [25–29].

The rest of this paper is structured as follows: Section 2 presents the optimization problems central to our research. Section 3 provides an overview of the generalized Spiking LCA, along with theoretical demonstrations of its convergence across various optimization scenarios. Section 4 compares our algorithm with a traditional widely used optimization algorithm FISTA [7] in solving practical problems within compressed sensing and signal processing domains. Lastly, we broaden the application of our algorithm to encompass additional types of real-world optimization problems.

1.1. Related works

The Spiking LCA was primarily developed to tackle the constrained LASSO optimization problem [18]. The algorithm's ability to converge to the precise solution of the constrained LASSO problem was theoretically proven in Ref. [19]. Moreover, a rigorous analysis of the convergence rate, which enhanced our understanding of the computational capabilities of SNNs, was provided in Ref. [30]. Subsequent research has effectively extended the Spiking LCA on neuromorphic hardware to address practical problems [25–29,31]. However, in these studies, the hardware was limited to solving a single type of optimization problem at a time, as addressing different optimization problems required alterations to the chip architecture [32–34]. Additionally, the neuron model employed in the Spiking LCA was based on a capacitor circuit, which contrasts with the resistor–capacitor (RC) circuit models that more accurately represent real neurons. Our work addresses these issues by developing a generalized Spiking LCA. This algorithm is designed to support efficient multitasking within the same chip architecture, while ensuring compatibility with diverse neuron models and diverse connectivity patterns among neurons.

In the field of signal processing, various optimization algorithms have been developed to solve sparse representation problems. These include gradient descent-based methods such as the ISTA, its accelerated version FISTA [7], and the learned ISTA (LISTA) [35], which incorporates deep learning techniques to enhance performance. Additionally, greedy algorithms like matching pursuit are widely used, iteratively selecting dictionary atoms that best match the residual signal [36]. Basis pursuit formulates the sparse representation problem as a linear programming task, seeking the sparsest solution that satisfies signal reconstruction constraints [37]. Moreover, several evolutionary algorithms (EAs), such as Genetic Algorithm, Differential Evolution, and Particle Swarm Optimization, have been applied to sparse representation optimization problems [38–40]. Among these methods, FISTA has gained prominence due to its effectiveness and computational efficiency in large-scale sparse optimization problems, as detailed in A.2. Accordingly, we compare our generalized Spiking LCA with FISTA to highlight the superior performance of our method.

2. Sparse approximation and recovery problems

Sparse approximation and recovery problems are fundamental problems in signal processing and machine learning, aiming to exploit signals' sparsity for various applications. These problems have attracted significant attention in recent years due to their wide range of applications, including image processing, seismic wave detection, and feature selection [41–44].

Sparse approximation primarily represents a given signal using a few non-zero coefficients from an overcomplete dictionary. It seeks the optimal sparse linear combination of atoms (basis functions) in the dictionary that approximates the given signal. In sparse approximation problems, we assume that the signal comprises structured components and unstructured additive noise, as expressed in Eq. (1),

$$\mathbf{s} = \boldsymbol{\Phi}\mathbf{a} + \boldsymbol{\epsilon}, \tag{1}$$

where a vector input $\mathbf{s} \in \mathbb{R}^M$ corresponds to an input signal from a particular class of signals. It is a linear combination of an overcomplete dictionary $\boldsymbol{\Phi} = [\phi_1, \phi_2, \dots, \phi_N]$ (a dictionary with more atoms than the input signal dimension) using coefficients $\mathbf{a} \in \mathbb{R}^N$. Furthermore, $\boldsymbol{\epsilon}$ represents additive Gaussian white noise.

While the sparse approximation problem focuses on finding the optimal sparse representation $\mathbf{a}$ for a given signal $\mathbf{s}$ using an overcomplete dictionary, sparse recovery aims to reconstruct a sparse signal from a set of limited, noisy, and underdetermined measurements. This problem frequently arises in compressed sensing, which aims to accurately recover the original sparse signal from a smaller number of linear measurements than those required by the Nyquist–Shannon sampling theorem. The primary concept behind compressed sensing is to exploit the inherent sparsity or compressibility in a specific transformation domain, such as the wavelet or Fourier domain [45–47]. Mathematically, compressed sensing is described as follows. Consider a signal $\mathbf{x} \in \mathbb{R}^N$, which is K-sparse in a transformation domain $\boldsymbol{\Psi}$, i.e., only K coefficients in the transformed domain are non-zero ($K \ll N$). This can be expressed as $\mathbf{x} = \boldsymbol{\Psi}\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^N$ is the K-sparse coefficient vector. The signal $\mathbf{x}$ can be measured using an $M \times N$ measurement matrix $\boldsymbol{\Phi}$, with $M \ll N$, resulting in a compressed measurement vector $\mathbf{s} \in \mathbb{R}^M$, satisfying: $\mathbf{s} = \boldsymbol{\Phi}\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{a}$. Since $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are incoherent, i.e., their columns are not correlated, the product of the two matrices $A = \boldsymbol{\Phi}\boldsymbol{\Psi}$ can be treated as a new sensing matrix. The goal is to recover the sparse coefficient vector $\mathbf{a}$ from the compressed measurement vector $\mathbf{s}$.

The algorithms used to solve sparse recovery and sparse approximation problems are often the same. In this paper, A represents either $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}\boldsymbol{\Psi}$ collectively. We aim to solve the underdetermined system $\mathbf{s} = A\mathbf{a} + \epsilon$, with the prior knowledge that only a few entries in $\mathbf{a}$ are non-zero. This problem is mathematically formulated as follows:

$$\min_{\mathbf{a}} E = \frac{1}{2}\|\mathbf{s} - A\mathbf{a}\|_2^2 + \lambda \widetilde{C}(\mathbf{a}), \tag{2}$$

where the objective function in Eq. (2) comprises two terms. The first term measures the mean squared reconstruction error (MSR) of signals while the second term $\widetilde{C}(\mathbf{a})$ imposes a penalty that promotes sparsity in the coefficient vector. The parameter λ balances the trade-off between data fidelity and sparsity. However, directly optimizing this objective function with the ℓ^0-norm, $\widetilde{C}(\mathbf{a}) = \|\mathbf{a}\|_0$, as the sparsity-inducing function is computationally intractable, i.e., an NP-hard problem [48]. Therefore, alternative surrogate sparsity-inducing functions are commonly used, with the ℓ^1-norm, $\widetilde{C}(\mathbf{a}) = \|\mathbf{a}\|_1$, a convex function that encourages sparsity, being a popular solution. The resulting optimization problem, known as LASSO [22], is expressed as:

$$\min_{\mathbf{a}} E = \frac{1}{2}\|\mathbf{s} - A\mathbf{a}\|_2^2 + \lambda\|\mathbf{a}\|_1. \tag{3}$$

In some problems, we have additional requirement on the variable $\mathbf{a}$ to be non-negative, i.e., $\mathbf{a} \geq 0$. This variation is referred to as the constrained LASSO problem:

$$\min_{\mathbf{a}\geq 0} E = \frac{1}{2}\|\mathbf{s} - A\mathbf{a}\|_2^2 + \lambda\|\mathbf{a}\|_1. \tag{4}$$

Although there have been significant advancements in solving the LASSO problem, such as ISTA, FISTA, and LISTA [7,35], its computational complexity is a major challenge for real-time digital signal processing applications that deal with large-scale signals. Thus, the computational demands of the LASSO problem limit its practical applicability, where rapid and low-power reconstruction algorithms are crucial. Therefore, there is a growing demand for efficient algorithms and hardware architectures that enable real-time processing of large-scale signals or data. Indeed, developing such technologies is vital for advancing signal processing and machine learning, and with real-world applications.

3. The generalized Spiking LCA

3.1. Review of LCA and Spiking LCA

To elucidate our approach, we first provide an overview of the Locally Competitive Algorithm (LCA) and its spiking variant, the Spiking LCA. The LCA, often termed Analog LCA, is known for its robust convergence characteristics and capability to tackle large-scale challenges [49]. The LCA's architecture comprises an interconnected neural network encompassing N neurons. The LCA can solve the LASSO problem represented in Eq. (3) by evolving the dynamics of the neuron's membrane potential described by

$$\begin{aligned}&\dot{\mathbf{u}}(t)=\frac{1}{\tau}[\mathbf{b}-\mathbf{u}(t)-\left(A^TA-I\right)\mathbf{a}(t)],\\&\mathbf{a}(t)=T_{\pm\lambda}(\mathbf{u}(t)),\quad T_{\pm\lambda}(\mathbf{u})=T_\lambda(\mathbf{u})+T_\lambda(-\mathbf{u})\\&T_\lambda(\mathbf{u}(t))=\max(|\mathbf{u}|-\lambda,0)\cdot\operatorname{sgn}(\mathbf{u}).\end{aligned}\tag{5}$$

In the above neuronal dynamics model, each neuron receives a constant input $\mathbf{b}\in\mathbb{R}^M$, which is determined as $\mathbf{b}=A^T\mathbf{s}$. Here, the matrix A and vector $\mathbf{s}$ are as defined in the original LASSO problem Eq. (3). When the firing threshold of the neuron is reached, the neuron dispatches inhibitory signals to its counterparts. $W=I-A^TA$ characterizes this inter-neuronal interaction strength, and τ denotes the time constant of neuronal response. The neurons communicate through activations $\mathbf{a}(t)$, akin to spike rates. The function $T_\lambda(\cdot)$ enforces output sparsity via a soft-thresholding mechanism. Recent empirical evidence suggests that LCA demonstrates local asymptotic stability, ensuring resilience against external disturbances and convergence to equilibrium states over time [50]. Consequently, for specific inputs, the system converges to a unique and stable solution consistent with the global minimum of a LASSO optimization problem [50].

Despite the efficacy of LCA in solving large-scale challenges, it has certain limitations. For example, its dependence on continuous-time dynamics may enhance computational and energy costs, especially on traditional computing platforms. Therefore, the Spiking LCA is conceived to circumvent these challenges and harness the potential of neuromorphic hardware [25,51]. The Spiking LCA harnesses the energy-efficient properties inherent to spiking neural encoding by integrating spike-driven neuronal dynamics into the LCA paradigm. This integration significantly enhances power efficiency by capitalizing on the strengths of spiking neural networks (SNNs) [19,25,27,51]. Note that the Spiking LCA addresses the constrained LASSO specifically because of the inherently non-negative firing rates of neurons.

To implement the Spiking LCA in an SNN, each of the N neurons receives a somatic input current $\mu_i(t)$ over time t to change its membrane potential $v_i(t)$. The membrane potential accumulates according to the equation

$$v_i(t)=\int_0^t(\mu_i-\lambda)dt,\tag{6}$$

while it remains below the firing threshold v_{th}, and $\lambda\geq 0$ represents a predefined bias current. This bias current is set as the constant λ specified in Eq. (4). When $v_i(t)$ reaches v_{th} at time $t_{i,sp}$, neuron i is said to fire a spike, and $v_i(t)$ is set to the value of the reset voltage v_{reset}. At the same time, inhibitory currents are injected into all other neurons connected with neuron i. In the numerical simulation, the non-dimensional values $v_{reset}=0$, $v_{th}=1$ are used.

The somatic input current $\mu_i(t)$ consists of a constant input current $b_i=A_i^T\mathbf{s}$ and synaptic input currents from other neurons, which is governed by

$$\mu_i(t)=b_i-\sum_{j\neq i}w_{ij}\left(\alpha*\sigma_j\right)(t),\tag{7}$$

where $w_{ij}=A_i^TA_j$ is the synaptic weight from neuron j to neuron i, $\sigma_j(t)=\sum_k\delta(t-t_{j,k})$ is the sum of Dirac delta functions, and $t_{j,k}$ corresponds to the kth spike time of the jth neuron. The function $\alpha(t)=e^{-t}$ for $t\geq 0$ and zero otherwise, implying that the synaptic current is modulated by a weighted exponential decay function when an input is received, consistent with experimental observation. The operator $*$ denotes convolution. Differentiating both sides with respect to time t, we obtain the differential equation

$$\dot{\mu}_i(t)=b_i-\mu_i(t)-\sum_{j\neq i}w_{ij}\sigma_j(t).\tag{8}$$

The Eqs. (6)–(8), along with the definition of the spike trains $\sigma_i(t)$, describe the Spiking LCA.

To demonstrate the algorithm's convergence, we introduce two variables, the spike rate $a_i(t)$ and the average somatic input current $u_i(t)$, which are defined below:

$$\begin{aligned}a_i(t)&=\frac{1}{t-t_0}\int_{t_0}^t\sigma_i(s)ds,\\u_i(t)&=\frac{1}{t-t_0}\int_{t_0}^t\mu_i(s)ds.\end{aligned}\tag{9}$$

Here, t_0 is the initial time point of the simulation or experiment, serving as the starting point for averaging. Using definitions of u_i, μ_i and a_i from Eqs. (8)–(9), we can derive:

$$\dot{u}_i(t)=b_i(t)-u_i(t)-\sum_{j\neq i}w_{ij}a_j(t)-\frac{[u_i(t)-u_i\left(t_0\right)]}{t-t_0},\tag{10}$$

which is the spiking analog of the original LCA dynamics Eq. (5). In the Spiking LCA, the potential accumulation is regulated by Eq. (6). Consequently, the relationship between $u_i(t)$ and $a_i(t)$ satisfies $T_\lambda\left(u_i(t)\right)-a_i(t)\to 0$ as $t\to\infty$, where $T_\lambda(\cdot)$ is described by

$$T_\lambda(u(t))=\begin{cases}u(t)-\lambda & \text{if } u(t)>\lambda\\0 & \text{else.}\end{cases}\tag{11}$$

Strict inhibitory connections ensure the average soma current remains within bounds. As $t\to\infty$, $\left(u_i(t)-u_i\left(t_0\right)\right)/\left(t-t_0\right)\to 0$. This indicates that the system tends towards the same limit as observed in LCA, which is equivalently the solution to the constrained LASSO problem Eq. (4) [19].

3.2. The generalized Spiking LCA

The spiking LCA excludes essential features such as the leaky property and refractory period of a biological neuron. Furthermore, it considers a linear input–output curve, in contrast with the non-linear dynamics observed in real biological neurons. Regarding the networks' architecture, existing LCA algorithms focus on inhibitory connections, limiting their application to a particular set of optimization problems. Therefore, to generalize the Spiking LCA to integrate a wide range of more biologically plausible neuron models in a unified framework [19,25,27], we now develop a generalized Spiking LCA.

Fig. 1 illustrates the sparse coding idea and the core architecture of the generalized Spiking LCA model. This model involves a network comprising N interconnected neurons linked to all others through a current-based point neuron mechanism. The neuronal dynamics is governed by:

$$\begin{aligned}&c\frac{dv_i}{dt}=I_i^{ion}+I_i^{input}(t),\quad i=1,\ldots,N\\&\text{if } v_i(t)>v_{th}\quad v_i(t)=v_{reset}\quad t\in(t_{sp},t_{sp}+t_{ref}),\end{aligned}\tag{12}$$

where c represents the neuron's membrane capacitance, $I_i^{ion}(t)$ is the ionic current in the neuron, and $I_i^{input}(t)$ refers to the injected current that depends on the recurrent inputs and the external constant inputs, which will be determined below. A neuron's membrane potential governs the generation of a spike train, which accumulates according to Eq. (12). The corresponding neuron generates a spike when the membrane potential reaches the firing threshold v_{th} at a specific time $t=t_{sp}$. This spike either inhibits or excites other neurons and resets its potential to the resting potential v_{reset} during the refractory period.

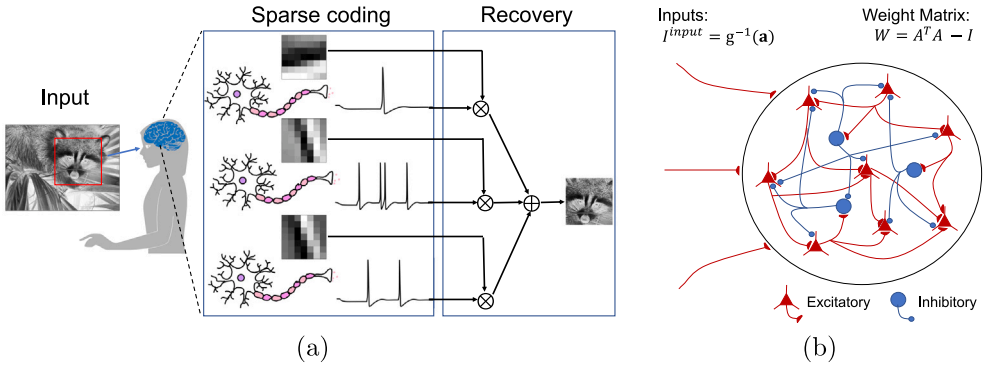


Fig. 1. The generalized Spiking LCA inspired by the visual cortex system. (a) Sparse coding is a technique used to simulate the sparse neural activity observed in the primary visual cortex. The input signal is reconstructed by computing a weighted sum of the receptive fields of model neurons, representing the specific regions of the input space that each neuron responds to. This approach allows for efficient and selective visual information processing, similar to the brain's. (b) In the generalized Spiking LCA, each neuron receives an external input, $I^{input} = g^{-1}(\mathbf{a})$, as well as recurrent input from neighboring neurons.

Following the spiking LCA, the average soma current $u_i(t)$ is required to follow

$$\dot{u}_i(t) = b_i(t) - u_i(t) - \sum_{j\neq i} w_{ij} a_j(t) - \frac{[u_i(t) - u_i(t_0)]}{t - t_0}. \tag{13}$$

Note that, for different forms of ionic current I_i^{ion}, the relationship between firing rate $a_i(t)$ and the input current $u_i(t)$ may not necessarily satisfy $a_i(t) = T_\lambda(u_i(t))$ as that in the classic Spiking LCA model, where $T_\lambda(u)$ is defined in Eq. (11). As this relation is crucial to prove the convergence of the spiking LCA that solves the constrained LASSO problem, our generalized Spiking LCA requires further design on the input current I_i^{input} to make the relation $a_i(t) = T_\lambda(u_i(t))$ hold.

To design the input current, we next take the leaky integrate-and-fire model as an example, i.e., $I_i^{ion} = -g_L(v_i - v_{reset})$, where g_L is the leaky conductance, v_{reset} is the reset potential after a spike. We can analytically solve the model and derive both the gain function $a = g(u)$ and its inverse function, which are depicted in Eq. (14).

$$\begin{aligned} a = g(u) &= \left[t_{ref} - \frac{c}{g_L}\log\left(1 - \frac{g_L v_{th}}{u}\right)\right]^{-1}, u \geq g_L v_{th} \\ g^{-1}(a) &= \frac{g_L v_{th}}{1 - \exp(g_L(t_{ref} - \frac{1}{a})/c)}. \end{aligned} \tag{14}$$

We compute the average soma current, $u_i(t)$, for each neuron at every time step based on Eq. (13). Using this current, we then calculate the input current $I_i^{\text{input}} = g^{-1}(T_\lambda(u_i))$ and apply it to each neuron in the subsequent time step. Accordingly, we then ensure that the output firing rate a_i now satisfies the condition $a_i = T_\lambda(u_i)$. This holds because for the ith neuron,

$$a_i = g(I_i^{\text{input}}) = g(g^{-1}(T_\lambda(u_i))) = T_\lambda(u_i). \tag{15}$$

The above procedure can be generalized to a large variety of neuron models beyond the leaky integrate-and-fire model, and correspondingly we derive the following theorem for the convergence of our generalized Spiking LCA.

Theorem 1. *If the gain curve of the neuron model in the spiking neural network is continuous (not limited to the LIF model), then by applying an external input current* $I_i^{input} = g^{-1}(T_\lambda(u_i))$, *as time approaches infinity, the firing rate* $\mathbf{a}$ *is equivalent to the output of LCA and converges to the optimal solution of Eq. (4).*

Proof. For any neuron model with a continuous gain function $g(\cdot)$, the average current dynamics of the ith neuron in the spiking neural network established based on this model satisfies

$$\begin{aligned} \dot{u}_i(t) &= b_i(t) - u_i(t) - \sum_{j\neq i} w_{ij} a_j(t) - \frac{[u_i(t) - u_i(t_0)]}{t - t_0}, \\ a_i &= g\left(I_i^{\text{input}}\right). \end{aligned} \tag{16}$$

To prove the convergence of this system, we first introduce two lemmas:

Lemma 1. *If* $g(\cdot) = T_\lambda(\cdot)$, *applying any additional current at each step is unnecessary. As time approaches infinity, the firing rate* $\mathbf{a}$ *is equivalent to the output of LCA and converges to the optimal solution of Eq. (4).*

Proof. see Ref. [18] for details. □

Lemma 2. *There exists an upper bound* B_+ *and a lower bound* B_- *such that* $\mu_i(t), u_i(t) \in [B_-, B_+], \forall i, t \geq 0$.

Proof. In terms of network connections, our model distinguishes itself from previous works, which solely permitted inhibitory connections to maintain bounded soma current magnitudes and the corresponding average potentials. By incorporating realistic neuron models into our approach, the firing rate of the neurons is inherently limited, precluding it from becoming infinitely large. Consequently, we establish a lower bound and an $R > 0$ such that $t_{i,k+1} - t_{i,k} \geq 1/R$ for all $i = 1, 2, \ldots, n$, and $k \geq 0$, whenever two spike times are present. This insight confirms that the soma currents in our model are bounded both above and below. We define $C = \max_{i,j}|w_{i,j}|$ and $B = \max_j |b_j|$, acknowledging that the inner product of features and biases is finite. Employing the fact that $(\alpha * \sigma_j)(t) \leq \sum_{l=0}^{\infty} e^{-\frac{l}{R}} < \infty$, we demonstrate the following:

$$\begin{aligned} \|\mu_i(t)\| &= \left\| b_i - \sum_{j\neq i} w_{ij}(\alpha * \sigma_j)(t) \right\| \\ &\leq \left\| |b_i| + \sum_{j\neq i} |w_{ij}| (\alpha * \sigma_j)(t) \right\| \\ &\leq \left\| \max_j |b_j| + \sum_{j\neq i} |w_{ij}| (\alpha * \sigma_j)(t) \right\| \\ &\leq \left\| B + nC(\alpha * \sigma_j)(t) \right\| \\ &\leq \left\| B + nC \sum_{l=0}^{\infty} e^{-\frac{l}{R}} \right\| < \infty. \end{aligned} \tag{17}$$

Implying the soma currents are bounded from above and below. □

Proof.

Hence, we adopt the proof of Ref. [19] and state $u(t) = \left[u_1(t), u_2(t), \ldots, u_N(t)\right]^T$ has at least one limit point $u^* \in \mathbb{R}^N$ such that $u\left(t_k\right) \to u^*$ as the sequence of $t_k \to \infty$ when $k \to \infty$ from the Bolzano–Weierstrass theorem. This implies:

$$\lim_{t\to\infty} \dot{u}_i(t) = \lim_{t\to\infty} \frac{\mu_i - u_i}{t - t_0} = 0. \tag{18}$$

For other neuron models, we evaluate the average soma current $u_i(t)$ for each neuron at each iteration. We then calculate the input current $I_i^{\text{input}} = g^{-1}\left(T_\lambda\left(u_i\right)\right)$ using this current. Subsequently, we determine the output firing rate a_i as $a_i = T_\lambda\left(u_i\right)$ by applying the activation function $g(\cdot)$ to input current I_i^{input}. This sequence of steps ensures that our spiking neural network converges to the solution of the constrained LASSO problem. The dynamics of $u_i(t)$ can be expressed as follows:

$$\begin{aligned} &\dot{u}_i(t) = b_i - u_i(t) - \sum_{j\neq i} w_{ij} a_j(t), \\ &a_i = g\left(I_i^{\text{input}}\right) = T_\lambda\left(u_i\right) \quad t \to \infty. \end{aligned} \tag{19}$$

Hence $T_\lambda\left(u\left(t_k\right)\right) \to T_\lambda\left(u^*\right) = a^*$, we can conclude the system converges to the same limit found in LCA. With the above results, we complete the proof. □

In the A.1, four commonly used neuronal models are introduced. Note that analytical expressions for the gain function of these models are generally infeasible. However, we can approximate the gain function numerically, and subsequently incorporate them into our algorithm and perform numerical experiments below.

In our previous analysis, we set $\widetilde{C}(\mathbf{a}) = \|\mathbf{a}\|_1$ in Eq. (2) to solve the LASSO problem. As we progress, we intend to show the versatility of our algorithmic framework in addressing diverse optimization problems. This adaptability can be achieved by substituting $\widetilde{C}(\mathbf{a})$ with alternative penalty functions. Specifically, we investigate our algorithm's capability to tackle the Elastic-Net optimization and the unconstrained LASSO problem, which are two important problems often encountered in real-world applications. In fact, for any penalty function that satisfies the following rules,

1. $\widetilde{C}(\cdot)$ is non-negative and subanalytic on $[0, +\infty)$.
2. The first-order derivative of $\widetilde{C}(\cdot)$ is continuous and non-negative on $[0, +\infty)$, i.e., $\widetilde{C}'(\cdot) > 0$.
3. Define $T_\lambda^{-1}(a) = \lambda \frac{d\widetilde{C}(a)}{da} + a$, then the first-order derivative of $T_\lambda(\cdot)$ is continuous and positive on $(0, +\infty)$. i.e., $T_\lambda'(\cdot) > 0$.

For rules 1 and 2, convergence of the corresponding LCA system is guaranteed [51–54]. Regarding rule 3, it is established that within the interval $(0, +\infty)$, functions such as $T_\lambda(\cdot)$ and $T_\lambda^{-1}(\cdot)$ exist. Based on these assurances, we can establish the following theorem:

Theorem 2. *Consider $\widetilde{C}(\cdot)$ satisfying previously mentioned rules 1–3. Then, the firing rate* **a** *of the spiking neural network is globally asymptotically convergent, and* **a** *will converge to the solution of the corresponding optimization problem described by Eq. (2).*

Proof. Let $\widetilde{C}(\cdot)$ adhere to all the stipulations previously outlined. Under such a condition, the average current dynamics of the ith neuron in the spiking neural network satisfies

$$\begin{aligned} &\dot{u}_i(t) = b_i(t) - u_i(t) - \sum_{j\neq i} w_{ij} a_j(t) - \frac{\left[u_i(t) - u_i\left(t_0\right)\right]}{t - t_0}, \\ &a_i = T_\lambda(u_i), \quad \lambda \frac{d\widetilde{C}\left(a_i\right)}{da_i} = u_i - a_i = T_\lambda^{-1}\left(a_i\right) - a_i, \end{aligned} \tag{20}$$

where the interconnection between $\widetilde{C}(a_i)$ and $T_\lambda(u_i)$ is given by the equation $\lambda \frac{d\widetilde{C}(a_i)}{da_i} = T_\lambda^{-1}(a_i) - a_i$. By utilizing this relationship, we can solve for a_i in terms of u_i to determine the corresponding activation function $T_\lambda(\cdot)$.

To tackle the optimization problem by Eq. (2), we perform the following steps for each neuron in our network. First, we compute the average current $u_i(t)$ at every iteration step. Then, we inject the current $I_i^{\text{input}} = g^{-1}\left(T_\lambda\left(u_i\right)\right)$, where different activation functions $T_\lambda(\cdot)$ are used for the optimization problem with different penalty functions. This leads to the neuron's output activation $a_i = g\left(I_i^{\text{input}}\right) = g\left(g^{-1}\left(T_\lambda\left(u_i\right)\right)\right) = T_\lambda\left(u_i\right)$. This procedure ensures the convergence of our spiking neural network towards the solution of the general optimization problem Eq. (2). □

4. Numerical experiments

This section presents a series of numerical experiments to demonstrate the effectiveness of the generalized Spiking LCA and its convergence behavior. Our tests encompass both synthetic and real datasets. For the synthetic data, we aim to confirm the solution equivalence between the generalized Spiking LCA and various optimization problems. Subsequently, we juxtapose the performance of our algorithm with different penalty functions, highlighting the superiority of our algorithm regarding power consumption and processing time. The dictionary entries Φ are sampled randomly from a standard Gaussian distribution, represented by $\mathcal{N}(0, 1)$. Additionally, we explore the algorithm's efficacy in several practical applications, such as sparse signal and CT image reconstruction. All numerical experiments are conducted the Brainpy neural engineering simulation platform [55] on a server powered by an A100 GPU platform. As a comparative algorithm, FISTA is implemented in Python utilizing the NumPy library for efficient computation.

As previously mentioned, our goal is to solve the sparse approximation and recovery problem, for which we have constructed a corresponding sparsely firing spiking neural network. Based on this, we have analyzed the computational complexity of our algorithm for solving a problem with a dimension of N. For a network with N neurons, the two primary computational components are neuron state updates and spike transmission. Neuron state updates involve recalculating membrane potentials and average somatic input current $u_i(t)$ for each neuron, $i = 1, 2, \ldots, N$, resulting in a computational cost of $O(N)$ per time step. The computational cost of spike transmission depends on the number of neurons firing and their connections at each time point. For instance, when only one neuron fires, processing this spike requires updating the synaptic inputs of all connected neurons, leading to a computational complexity of $O(N)$ in a fully connected network. However, because neurons in our problem fire sparsely—so the average number of neurons firing at each time step is much less than one—the overall computational burden is predominantly due to neuron state updates. In comparison, FISTA has a computational complexity of $O\left(N^2\right)$ per iteration because it involves matrix–vector multiplications and requires updating all variables in each iteration. This leads to higher computational costs, especially for large N.

The energy efficiency of our algorithm is based on its computational complexity, as discussed previously. Our proposed generalized Spiking LCA achieves a lower computational complexity compared to FISTA, thereby reducing the number of operations required for convergence. When implemented on traditional von Neumann hardware architectures, this reduction in computational complexity translates to lower energy consumption.

More importantly, since our algorithm is based on spiking neural networks (SNNs), deploying it on neuromorphic platforms such as Intel's Loihi or IBM's TrueNorth allows it to benefit from hardware optimized for spike-based computations. These chips exploit the sparse and asynchronous nature of spiking neural computations to achieve low-power performance. Recent work has implemented the original Spiking LCA algorithm on the neuromorphic Loihi chip, achieving approximately 1% of the power consumption required by FISTA [25,51]. In addition, the energy consumption of our generalized Spiking LCA is comparable to the original Spiking LCA.

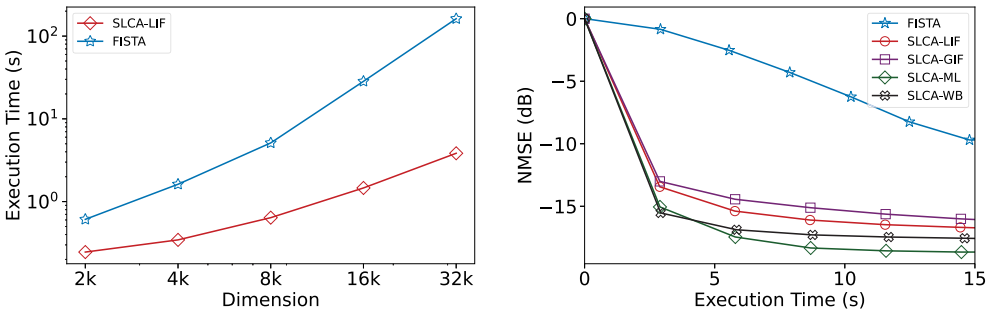


Fig. 2. Numerical results for the toy model. (a) The computational time for solving LASSO problems of varying sizes, comparing spiking neural networks with FISTA. (b) Performance of spiking neural networks based on various models, i.e., Leaky Integrate-and-Fire, Generalized Integrate-and-Fire model, Morris–Lecar, and Wang–Buzsaki model, in solving optimization problems, $A \in \mathbb{R}^{5000\times 10000}$. The metric $NMSE = 10\log_{10}\left(\|\hat{a}-a\|_2^2/\|\hat{a}\|_2^2\right)$ indicates normalized mean square error, where $\hat{a}$ denotes the original signal.

4.1. Signal recovery

In the last decade, sparse signal recovery, particularly in solving the LASSO problem, has attracted significant attention from researchers. Hence, to assess our algorithm's performance, we start with a toy model involving the recovery of a real-valued signal, denoted as $\mathbf{a} \in \mathbb{R}^N$, where N is the signal length. To generate the synthetic sparse signal $\mathbf{a}$, we randomly select K positions out of N to be nonzero. These nonzero elements are assigned values drawn independently from a standard normal distribution, while the remaining $N-K$ elements are set to zero, ensuring $\|\mathbf{a}\|_0 = K$. We then generate the observation matrix $A \in \mathbb{R}^{m\times n}$ (with $m < n$), where each entry A_{ij} is independently drawn from a Gaussian distribution with zero mean and unit variance. The observation vector $\mathbf{b}$ is computed as the product of A and $\mathbf{a}$, i.e., $\mathbf{b} = A\mathbf{a}$. Our experiments assess the execution time of FISTA and generalized Spiking LCA, and their recovery performance. As the signal recovery problem scale enlarges, FISTA's runtime significantly increases. In contrast, generalized Spiking LCA demonstrates a steadier ascent in computation time, as illustrated in Fig. 2(a). This difference is due to our algorithm's execution on the A100 GPU, which supports extensive parallel computations. As a result, the growth in problem size does not lead to a sharp rise in the computation time of our method.

In principle, the execution time of the generalized Spiking LCA can be further substantially reduced if we implement the algorithm on the neuromorphic chip like Loihi by achieving parallel computation across all neuron nodes. In this context, Loihi dedicates additional resource cores to manage larger neuron sizes, enabling extensive parallel processing across all cores. Hence, the runtime of generalized Spiking LCA is more influenced by factors such as the number of neurons within a resource core and spike traffic rather than the specific scale of the optimization problem. This feature highlights the substantial potential of generalized Spiking LCA for practical applications and its promising advantages in real-world scenarios.

We next evaluate the performance of the generalized Spiking LCA. Apart from the LIF model, our algorithm can construct networks based on other biophysical neuron models, such as the GIF model [56], the Morris–Lecar (M–L) model [57], and the Wang–Buzsaki model [58]. The detailed model description can be found in the A.1. For the ease of comparative testing, we set $n = 10000$, $m = 5000$, and $K = 500$, and use normalized mean square error (NMSE) as a measure of estimation error. The simulation results are illustrated in Fig. 2(b), which infers that our algorithm has a quicker initial convergence than the FISTA method across diverse SNN architectures with different neuronal models.

In the previous sections, we have introduced our algorithm's capability to solve the constrained LASSO problem. However, in many cases, leveraging the ℓ_p norm with $0 < p < 1$ as a regularization term can yield solutions that are both sparser and more accurate [59,60]. Direct minimization of these non-convex norms is challenging; consequently, various approximation methods have been widely adopted [59–61]. Common approaches include iterative reweighted algorithms and the construction of non-convex functions that approximate the ℓ_p norm. Given that the second approach is easier to implement, we adopt it to optimize our algorithm.

We then investigate the performance of generalized Spiking LCA with non-convex penalty functions, for instance, exponential, logarithmic, and arctangent functions. The exponential penalty is defined as $\widetilde{C}(x) = 1 - e^{-\gamma x}$ with $\gamma > 0$. The logarithmic penalty is given by $\widetilde{C}(x) = log(x+\theta)$ where $\theta \geq 1$, and the arctangent penalty is expressed as $\widetilde{C}(x) = arctan(x/\eta)$ with $\eta > 0$. These non-convex functions need to satisfy rules 1–3 to ensure stability and convergence during experiments. It is obvious that these functions meet rules 1 and 2. For rule 3, there is a wide range of possible penalties that can be applied under an appropriate λ, To meet this rule, the condition $\widetilde{C}''(x) > -\frac{1}{\lambda}$ must be satisfied for all $x \in (0, +\infty)$.

In our experiments, we initially fine-tune the parameter λ through numerical tests to investigate its impact on the algorithm's performance under various penalty settings. With parameters set to $n = 1200, m = 750$, and $K = 200$, the results reveal that performance is sensitive to λ, as shown in Fig. 3. Specifically, when λ is set too high, a large number of neurons become inhibited and cannot fire within a short time frame, adversely affecting the algorithm's performance. Conversely, a small λ diminishes the penalty effect, effectively nullifying the sparsity constraint. To balance these effects, we fine-tune λ within the range of 0.01 to 0.1 in subsequent experiments to find the optimal solution.

We further investigate the performance of our algorithm under different conditions, we set parameters $\gamma = 1$, $\theta = 1$, and $\eta = 1$. It is important to note that for vector $\mathbf{a}$, the function $\widetilde{C}(\cdot)$ operates on each component of the vector individually. For example, with $\widetilde{C}(x) = 1 - e^{-x}$, the function applied to vector $\mathbf{a}$ would be $\widetilde{C}(\mathbf{a}) = 1 - e^{-\mathbf{a}}$, where the ith component is $\widetilde{C}\left(a_i\right) = 1 - e^{-a_i}$. Additionally, the strategy applied to the current model adheres to the second approach delineated in Theorem 2, allowing for a robust and effective application of the theoretical framework. In our experiments, we focus on the signal recovery problem, same task described in Section 4.1. First, we test the algorithms using the same dictionary $\Phi \in \mathbb{R}^{7500\times 10000}$ in a noise-free scenario, with λ set to 0.1. The choice of penalty function also influences the algorithm's performance. The exponential penalty is the optimal choice among the penalties tested, as illustrated in Fig. 4(a). Fig. 4(b) illustrates the performance across varying sparsity levels, suggesting that the reconstruction quality improves as the signal becomes more sparse, which is evident from the decreasing NMSE values. To evaluate the algorithm's noise robustness, we compare signal recovery results across varying noise intensities, as shown in Fig. 4(c), where the noise intensity is represented by the signal-to-noise ratio (SNR) between the corrupted and original signals. We also examine the model's resilience to failures by comparing the probability of successful recovery using the

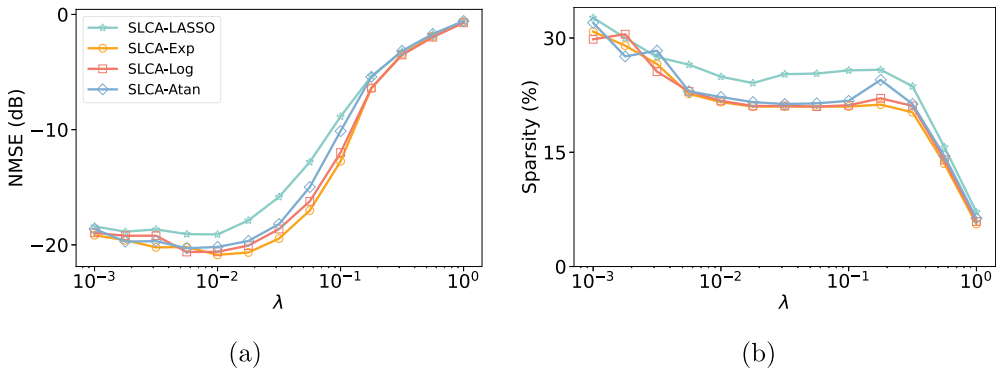


Fig. 3. Performance comparison of the generalized Spiking LCA variants across different values of the regularization parameter λ. (a) Recovery performance under varying λ values, measured by NMSE in dB, showing the accuracy of each algorithm's variant. (b) Sparsity of the recovered signal as a function of λ.

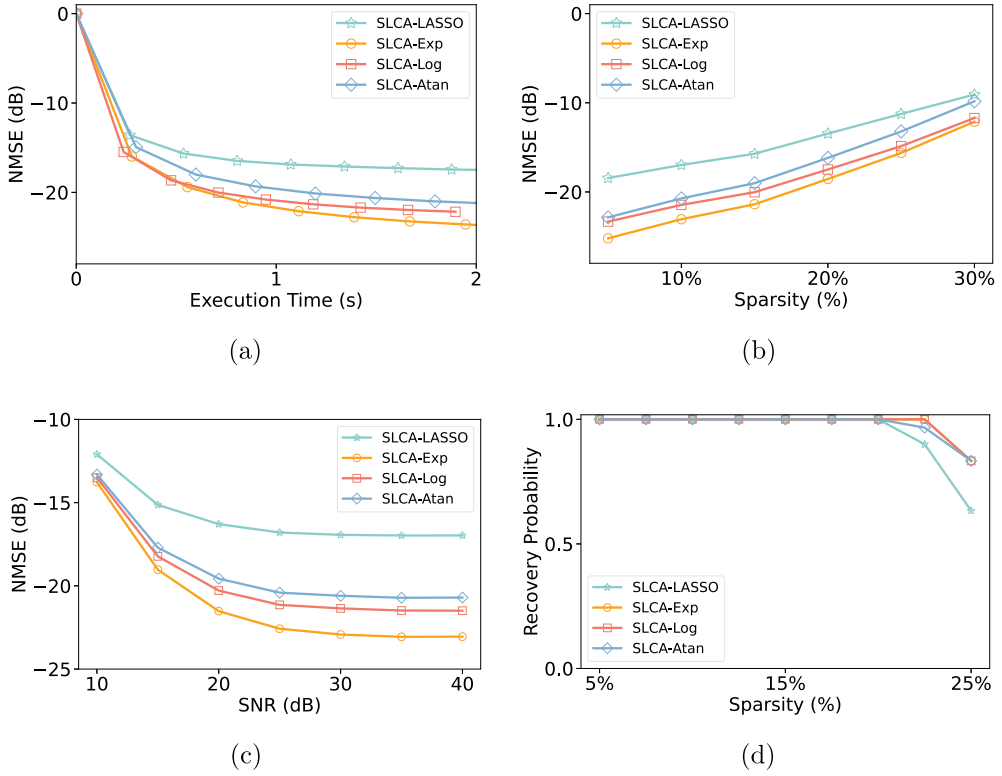


Fig. 4. Comparison of the generalized Spiking LCA algorithm's performance using non-convex penalty functions. (a) Convergence among four different non-convex penalty functions. (b) Performance across diverse sparsity levels, with sparsity represented as a percentage of non-zero elements in the signal. (c) Performance under different levels of noise, with noise intensity represented by the SNR between the corrupted and original signals. (d) Probability of successful recovery across diverse sparsity levels.

same dictionary at different sparsity levels. Here, successful recovery is defined as achieving an NMSE below −15 dB in a noise-free scenario, with each algorithm tested on a sample of 100 trials, following the reference. As shown in Fig. 4(d), when the sparsity is less than 20%, our algorithm consistently demonstrates robust performance across multiple experiments.

4.1.1. Elastic-Net

The Elastic-Net combines the ℓ^1 and ℓ^2 penalties of the LASSO and Ridge methods in a unified regularization approach [23], described as:

$$\min_{\mathbf{a}\geq 0} \frac{1}{2}\|\mathbf{s}-\boldsymbol{\Phi}\mathbf{a}\|_2^2 + \lambda(\rho\|\mathbf{a}\|_1 + \frac{1-\rho}{2}\|\mathbf{a}\|_2^2) \tag{21}$$

The generalized Spiking LCA framework is able to incorporate the Elastic-Net formulation by modifying the slope of the activation function $T_\lambda(\cdot)$ as follows:

$$T_\lambda(a) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } a \leq \lambda\rho \\ \frac{a-\lambda\rho}{\lambda(1-\rho)+1} & \text{if } a > \lambda\rho \end{cases} \tag{22}$$

To demonstrate the effectiveness of the generalized Spiking LCA on solving the Elastic-Net problem, we generate a dataset with sample size smaller than the total number of features, as an underdetermined problem. The target variable $\mathbf{s}$ is formed by combining sinusoidal signals with different frequencies. From the 100 frequencies in $\boldsymbol{\Phi}$, only the lowest 10 are utilized to generate $\mathbf{s}$. The remaining features remain inert, rendering the feature space both high-dimensional and sparse, thus requiring a certain level of ℓ_1-penalization.

We then split the data into training and testing sets. In Fig. 5(a), the blue line represents the signal we aim to reconstruct, and the light red and orange dots are the sampled points. Due to noise, these sampled

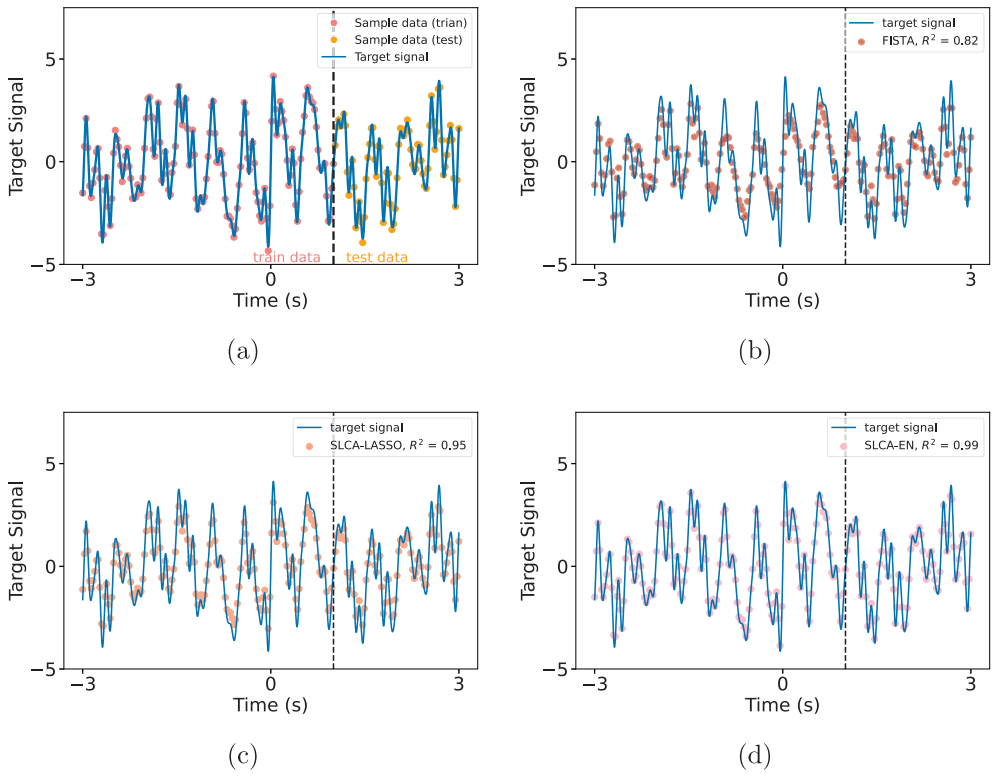


Fig. 5. LASSO and Elastic-Net methods for sparse signals recovery. (a) We partition the sampled points into training and testing sets to restore superimposed sinusoidal signals. (b) Reconstruction results of the FISTA algorithm, where the red pot represents the reconstructed signal and the blue line represents the target signal. The R^2 value is used to measure the discrepancy between the recovered signals in the test set and the actual data points. The execution time for the algorithm is standardized at 2 s, during which the FISTA optimization algorithm achieves an R^2 value of 0.82. (c) The R^2 value for LASSO is 0.95. (d) Elastic-Net achieves an R^2 value of 0.99.

points deviate from the true signal values. The light red dots serve as the training dataset, used to determine the coefficient value $\mathbf{a}$, while the orange dots act as the testing dataset, to which we subsequently apply this determined value. The performance of algorithm is evaluated based on their goodness of fit score. Fig. 5(b–d) display the results of the FISTA and the generalized Spiking LCA when applied to LASSO and Elastic-Net models. Both algorithms are run for 2 s. The generalized Spiking LCA consistently demonstrated more accurate predictions compared to FISTA. Moreover, the results underscore that Elastic-Net outperforms in terms of R^2 score. While LASSO is renowned for its capability in sparse data recovery, it underperforms when the features are highly correlated. Indeed, when several correlated features influence the target, LASSO selects only one representative feature from the group and discards the others. This can lead to potential loss of information. In contrast, Elastic-Net promotes sparsity in coefficient selection and slightly shrinks towards zero. Therefore, Elastic-Net adjusts their weights without eliminating them. This produces a less sparse model than a pure LASSO model.

4.1.2. LASSO

Next, we extend the generalized Spiking LCA to solve the unconstrained LASSO problem. We introduce non-negative variables $\mathbf{a}^+ \geq 0$ and $\mathbf{a}^- \geq 0$, such that $\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-$ and $|\mathbf{a}| = \mathbf{a}^+ + \mathbf{a}^-$. By defining $\widetilde{\boldsymbol{\Phi}} = [\boldsymbol{\Phi}, -\boldsymbol{\Phi}]$ and $\mathbf{z} = \left[\mathbf{a}^+, \mathbf{a}^-\right]$, we reformulate the objective function of the unconstrained LASSO as:

$$\begin{aligned} &\min_{\mathbf{a}\in\mathbb{R}^n} \sum_{i=1}^{n} \mathbf{a}^+ + \mathbf{a}^- \\ &\text{s.t. } \boldsymbol{\Phi}\mathbf{a}^+ - \boldsymbol{\Phi}\mathbf{a}^- = \mathbf{s}. \\ &\mathbf{a}^+, \mathbf{a}^- \geq 0 \end{aligned} \equiv \begin{aligned} &\min_{\mathbf{z}\geq 0} \|\mathbf{z}\|_1, \\ &\text{s.t. } \widetilde{\boldsymbol{\Phi}}\mathbf{z} = \mathbf{s}. \end{aligned} \tag{23}$$

The problem can be equivalently described using a Lagrangian multiplier approach as follows:

$$\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{s} - \widetilde{\boldsymbol{\Phi}}\mathbf{z}\|_2^2 + \lambda \sum_{k=1}^{2N} z_k, \quad z_k \geq 0. \tag{24}$$

Then we introduce a logarithmic barrier term to ensure that $z_k \geq 0$, transforming the constrained optimization problem into an unconstrained problem [62]. This can be written as:

$$\begin{aligned} &\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{x} - \widetilde{\boldsymbol{\Phi}}\mathbf{z}\|_2^2 + \lambda \sum_{k=1}^{2N} z_k - \left(\frac{1}{\gamma}\right) \sum_{k=1}^{2N} \log\left(z_k\right) \\ &\text{s.t.} \quad C\left(z_k\right) = z_k - \frac{\log\left(z_k\right)}{\gamma\lambda} \end{aligned} \tag{25}$$

where $C(z_k)$ represents a differentiable penalty function. The interconnection between $C(z_k)$ and $T_\lambda^{-1}(z_k)$ is given $\lambda\frac{dC(z_k)}{dz_k} = T_\lambda^{-1}(z_k) - z_k$, which allows one to solve for z_k in terms of u_k to determine the corresponding activation function $T_\lambda(\cdot)$. As an illustrative example, the generalized Spiking LCA has been effectively applied to reconstruct the CT image in Section 4.2.2, which belongs to the unconstrained LASSO problem.

4.2. Applications

4.2.1. Seismic data reconstruction

Next, we demonstrate the utility of our method in a more practical context: the reconstruction of seismic wave signals, a crucial aspect of the Earth's subsurface exploration and monitoring. Our experiment focuses on reconstructing seismic wave signals using the Ricker wavelet, renowned for its ability to provide a sparse representation of these signals. It showcases a relatively straightforward appearance characterized

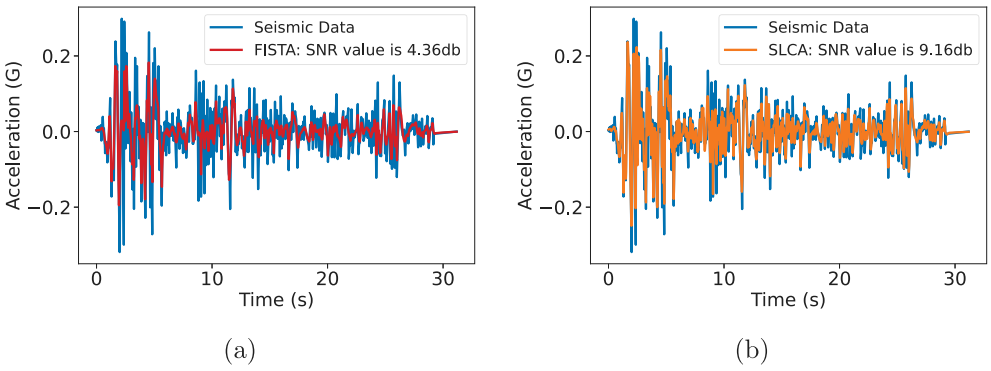


Fig. 6. Comparison of seismic signal reconstruction performance using the FISTA and generalized Spiking LCA algorithms. (a) Reconstruction results of the FISTA algorithm, where the red line represents the reconstructed signal and the blue line represents the original data. (b) Reconstruction results of the generalized Spiking LCA algorithm, where the yellow line represents the reconstructed signal and the blue line represents the original data.

by a dominant positive peak flanked by two negative side lobes. From an analytical perspective, its representation in the time domain is as follows:

$$A(t) = \left[1 - 2\pi^2 f^2 (t - t_0)^2\right] e^{-\pi^2 f^2 (t - t_0)^2}. \tag{26}$$

Here, $A(t)$ delineates the wavelet's amplitude at time t, f denotes the dominant frequency, and t_0 represents the center time of the wavelet. The goal is to recover the wavelet coefficient sequence $\mathbf{a}$ via the following model:

$$\min_{\mathbf{a} \geq 0} \frac{1}{2} \|\mathbf{s} - A\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1, \tag{27}$$

where $\mathbf{s}$ is the measured seismic data, A is the Ricker wavelet matrix, and λ controls the trade-off between the data fitting and the sparsity terms. By adjusting the frequency and center time of Ricker wavelets, you can create a diverse set of wavelets, each with its own distinctive oscillation speed and temporal positioning. The matrix A is then formed by sampling these varied wavelets at specific time points, with each column capturing the sampled values of a wavelet at a given frequency and center time, thereby encompassing a broad spectrum of seismic characteristics. Utilizing A, we model $\mathbf{s}$ by identifying an optimal combination of wavelet coefficients in the sequence $\mathbf{a}$. This approach enhances the interpretation of seismic data, improves the detection of seismic events, and facilitates the inversion process for estimating subsurface characteristics.

We evaluate the performance of both FISTA and the generalized Spiking LCA in recovering this signal, specifically using the El Centro Earthquake dataset. The effectiveness of each algorithm is assessed using the SNR, with λ fine-tuned for optimal performance, ultimately set to 0.01. As shown in Fig. 6, upon execution for a consistent time-frame of 2 s, the generalized Spiking LCA algorithm is demonstrated to have superior signal reconstruction quality compared to the FISTA algorithm, achieving an SNR value of 9.16 dB, in contrast to FISTA's 4.36 dB.

4.2.2. Computed tomography construction

This section highlights our system's potential benefits in a medical imaging application, where real-time compressive sensing (CS) recovery techniques can provide substantial improvements. Computed tomography (CT) is a widely used imaging technique in medical diagnosis and treatment. It involves using X-rays to create detailed images of the body's internal structures. Compressive CT imaging is of significant interest because it can reduce scan times, improving patient throughput and safety by reducing radiation exposure.

In this study, we explore the effectiveness of two different algorithms, generalized Spiking LCA and FISTA, for reconstructing a head CT image. The image can be represented as a vector $\mathbf{x}$, which is sparse in the Daubechies wavelet basis, i.e., it can be represented using a small number of wavelet coefficients $\mathbf{x} = \Psi\mathbf{a}$, where Ψ is the discrete orthogonal wavelet transform matrix [63]. The signal $\mathbf{x}$ can be measured using an $M \times N$ discrete Gaussian random measurement matrix Φ, with $M \ll N$ (e.g., the size ratio $M/N = 0.2$), resulting in a compressed measurement vector $s \in \mathbb{R}^M$. The experiment's objective is to recover the head CT image $\mathbf{x}$ using the following model:

$$\min_{\mathbf{a}} \frac{1}{2} \|s - \Phi\Psi\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 \tag{28}$$

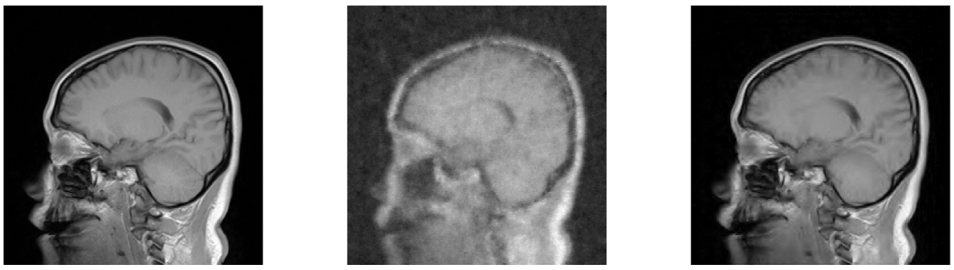
We evaluate the performance of the generalized Spiking LCA and FISTA algorithms for image reconstruction in CT using peak signal-to-noise ratio (PSNR) as a measure of estimation error. The parameter λ is fine-tuned and set to 0.01 for optimal results. Fig. 7 shows that the generalized Spiking LCA algorithm can produce higher quality reconstructed images, while the quality of images obtained by FISTA is relatively poor.

5. Conclusion

This paper develops a novel optimization algorithm for constructing spiking neural networks, generalized from the classical Spiking LCA. Our algorithm's unique feature is its flexibility, allowing it to handle various optimization problems within a unified network architecture. This is achieved by simply adjusting the external input current to neurons, which enhances the algorithm's versatility and minimizes the need for architectural adjustments, particularly when implemented on neuromorphic chips. The proposed spiking neural network is based on a large number of biologically plausible neuron models, encompassing a range from the LIF model to the Hodgkin–Huxley type model. We have demonstrated our model's practical utility and efficiency by applying it to various real-world problems related to compressed sensing and signal processing. By outperforming popular optimization algorithms, such as FISTA, our model confirms its capability to address large-scale optimization problems more effectively.

One limitation of our algorithm is that the dictionary bases are not learnable. However, in certain problems, learning the dictionary can better represent the underlying structure of the data and improve the performance of the algorithm. Recent studies have focused on designing neural network architectures to solve dictionary learning problems [64–67]. Incorporating these methods in our existing architecture can be a promising future research direction.

Another limitation of our algorithm is that it has not been applied to dynamic signals. However, our algorithm can process dynamic signals frame by frame independently with moderate efficiency as we have demonstrated the fast convergence of our algorithm for static signals. The efficiency can be further improved by taking advantage of the temporal variations of dynamic signals in the future.



(a) (b) (c)

Fig. 7. Reconstruction of 256 × 256 pixel CT images from simulated CS acquisition. (a) The original image. (b) The reconstructed image by FISTA, with PSNR value 24.4 dB at 12 s. (c) The reconstructed image by the generalized Spiking LCA, with PSNR value 34.6 dB at 12 s. These comparative images demonstrate the superior performance of the generalized Spiking LCA algorithm over FISTA when both algorithms are run for the same time.

CRediT authorship contribution statement

Xuexing Du: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Zhong-qi K. Tian:** Conceptualization. **Songting Li:** Writing – review & editing, Visualization, Supervision, Software, Methodology, Funding acquisition, Conceptualization. **Douglas Zhou:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Science and Technology Innovation 2030 - Brain Science and Brain-Inspired Intelligence Project with Grant No. 2021ZD0200204 (S.L., D.Z.); National Natural Science Foundation of China Grant 12271361, 12250710674 (S.L.); National Natural Science Foundation of China with Grant No. 12225109, 12071287 (D.Z.), Lingang Laboratory Grant No. LG-QS-202202-01, and the Student Innovation Center at Shanghai Jiao Tong University (X.-x.D., Z.-q.K.T., S.L., and D.Z.). We also thank Xiaoqun Zhang and Yanan Zhu for their valuable suggestions and discussions on optimization algorithms.

Appendix

A.1. Neuron models

GIF model. The dynamics of the ith neuron of a generalized leaky integrate-and-fire (GIF) network is governed [56,68]

$$\begin{aligned} c\frac{\mathrm{d}v_i}{\mathrm{d}t} &= -g_L\left(v_i - v_L\right) + \sum_j I_j + I_i^{\text{input}}, \\ \frac{\mathrm{d}\theta_i}{\mathrm{d}t} &= a\left(v_i - v_L\right) - b\left(\theta_i - \theta_\infty\right), \\ \frac{\mathrm{d}I_j}{\mathrm{d}t} &= -k_j I_j, \quad j = 1, 2, \ldots, n \\ \text{if } v_i > \theta_i, &\quad I_j \leftarrow r_j I_j + A_j, v_i \leftarrow v_{\text{reset}}, \theta_i \leftarrow \max\left(\theta_{\text{reset}}, \theta_i\right), \end{aligned} \tag{A.1}$$

where v_i represents the membrane potential of the neuron, and c refer to the membrane capacitance. The terms g_L and v_L are used to denote the leak conductance and the reversal potential, respectively. I_j represents an arbitrary number of internal currents. These currents are primarily influenced by the dynamics of ion channels, providing the model with the flexibility to simulate various neuronal firing patterns. Additionally, I_i^{input} denotes the externally injected current. The model encompasses a total of $n+2$ variables: the membrane potential v_i, the membrane potential threshold θ, and n internal currents. The decay of each internal current I_j is described by a third differential equation, with a decay rate of k_j.

In its formulation, the GIF model extends the Leaky Integrate-and-Fire (LIF) model by integrating the effects of internal ionic currents into the first differential equation. The second differential equation elaborates on the dynamics of the firing threshold θ, which is not constant. The first term of this equation accounts for the influence of the membrane potential on θ, while the second term delineates the decay of θ towards its equilibrium value θ_∞ at a decay rate b. Upon the firing of a neuron, the membrane potential v_i and the threshold θ are reset, and the internal currents are adjusted according to specific rules.

In the numerical simulation, we set parameters as $c = 1\ \mu\text{F}\cdot\ \text{cm}^{-2}$, $g_L = 0.05\ \text{mS}\cdot\ \text{cm}^{-2}$, $v_L = -70$ mV, $v_{reset} = -70$ mV, $\theta_\infty = -50$ mV, $\theta_{reset} = -60$ mV, $n = 0$, $r = 20$, $a = 0$, $b = 0.01$, $k_1 = 0.2$, $k_2 = 0.02$.

Morris–Lecar Model. The dynamics of the ith neuron of a Morris–Lecar network is governed by [57]

$$\begin{aligned} c\frac{dv_i}{dt} &= -g_{\text{Ca}} m_i^\infty\left(v_i - v_{\text{Ca}}\right) - g_K w_i\left(v_i - v_{\text{K}}\right) \\ &\quad - g_{\text{L}}\left(v_i - v_{\text{L}}\right) + I_i^{\text{input}} \\ \frac{dw_i}{dt} &= \frac{1}{\tau_i^w}\left(w_i^\infty - w_i\right), \end{aligned} \tag{A.2}$$

where c and v_i denote the neuron's membrane capacitance and membrane potential, respectively. I_i^{input} is the injected current. The terms $v_{\text{Ca}}, v_{\text{K}}$, and v_{L} represent the reversal potentials for the calcium, potassium, and leak currents, respectively. Correspondingly, $g_{\text{Ca}}, g_{\text{K}}$, and g_{L} are the maximum conductances for these currents. The variable w_i refers to the neuron's recovery variable, which is a normalized potassium conductance. The parameters m_i^∞ and w_i^∞ are the voltage-dependent equilibrium values for the normalized conductances of calcium and potassium, respectively, and are defined by

$$\begin{aligned} m_i^\infty &= 0.5\left(1 + \tanh\left[\left(v_i - V_1\right)/V_2\right]\right) \\ w_i^\infty &= 0.5\left(1 + \tanh\left[\left(v_i - V_3\right)/V_4\right]\right), \end{aligned} \tag{A.3}$$

where V_1, V_2, V_3, and V_4 are the constant parameters. τ_i^w is a voltage-dependent time constant of w_i, defined by

$$\tau_i^w = \phi\left(\cosh\frac{v_i - V_3}{2V_4}\right)^{-1} \tag{A.4}$$

where ϕ is a temperature-dependent parameter, fixed as a constant in simulation. When the voltage reaches threshold v_{th}, the neuron emits a spike to all its postsynaptic neurons. The parameters are set in numerical simulations as $g_{\text{Ca}} = 4.4\ \text{mS}\cdot\ \text{cm}^{-2}$, $v_{\text{Ca}} = 130$ mV, $g_{\text{K}} = 8\ \text{mS}\cdot\text{cm}^{-2}$, $v_{\text{K}} = -84$ mV, $g_{\text{L}} = 2\ \text{mS}\cdot\text{cm}^{-2}$, $v_{\text{L}} = -60$ mV, $c = 20\ \mu\text{F}\cdot\text{cm}^{-2}$, $V_1 = -1.2$ mV, $V_2 = 18$ mV, $V_3 = 2$ mV, $V_4 = 30$ mV,

$\phi = 0.04$, and $v_{\text{th}} = 0$ mV.

Wang–Buzsaki model. The dynamics of the ith neuron of a Wang–Buzsaki network is governed by [58]

$$c\frac{dv_i}{dt} = -\left(I_{Na} + I_K + I_L\right) + I_i^{\text{input}}, \tag{A.5}$$

where c is the cell membrane capacitance; v_i is the membrane potential (voltage); I_i^{input} is the injected current. The leak current $I_L = g_{\text{L}}\left(v_i - v_{\text{L}}\right)$, and the transient sodium current

$$I_{\text{Na}} = g_{\text{Na}} m_{i,\infty}^3 h\left(v_i - v_{\text{Na}}\right), \tag{A.6}$$

where the activation variable m_i is assumed fast and substituted by its steady-state function $m_{i,\infty} = \alpha_m/\left(\alpha_m + \beta_m\right)$. Additionally, the inactivation variable h_i follows first-order kinetics:

$$\frac{dh_i}{dt} = \phi\left(\alpha_h(1-h_i) - \beta_h h_i\right). \tag{A.7}$$

The delayed rectifier potassium current

$$I_{\text{K}} = g_{\text{k}} n_i^4\left(v_i - v_{\text{K}}\right), \tag{A.8}$$

where the activation variable n obeys the following equation:

$$\frac{dn_i}{dt} = \phi\left(\alpha_n(1-n_i) - \beta_n n_i\right). \tag{A.9}$$

The m_i, h_i, and n_i are gating variables; $v_{\text{Na}}, v_{\text{K}}$, and v_L are the reversal potentials for the sodium, potassium, and leak currents, respectively. Meanwhile, g_{Na}, g_{K}, and g_L correspond to the maximum conductances for these currents. The constant ϕ serves as a temperature regulation factor. The rate variables α_z and β_z $(z = m, h, n)$ are defined as:

$$\begin{aligned}
\alpha_m(v) &= -\frac{0.1(v+35)}{\exp(-0.1(v+35))-1}, & \beta_m(v) &= 4\exp(-\frac{v+60}{18}),\\
\alpha_h(v) &= 0.07\exp(-\frac{v+58}{20}), & \beta_h(v) &= 1/(\exp(-0.1(v+28))+1),\\
\alpha_n(v) &= \frac{-0.01(v+34)}{\exp(-0.1(v+34))-1}, & \beta_n(v) &= 0.125\exp(-\frac{v+44}{80}).
\end{aligned} \tag{A.10}$$

We take the parameters as in Ref. [58] that $c = 1\ \mu\text{F}\cdot\text{cm}^{-2}$, $v_{\text{Na}} = 55$ mV, $v_{\text{K}} = -90$ mV, $v_L = -65$ mV, $g_{\text{Na}} = 35\ \text{mS}\cdot\text{cm}^{-2}$, $g_{\text{K}} = 9\ \text{mS}\cdot\text{cm}^{-2}$, $g_L = 0.1\ \text{mS}\cdot\text{cm}^{-2}$ and $\phi = 5$. When the voltage v_i reaches the firing threshold, $v_{\text{th}} = 20$ mV, we say the ith neuron generates a spike at this time.

A.2. Optimization algorithms

In the study of sparse representation and optimization, various algorithms have been developed to solve the LASSO problem [7,35]. Among these, the FISTA has gained prominence due to their effectiveness and computational efficiency. This section provides a rationale for the choice of algorithms compared in our numerical results, focusing on the efficiency of FISTA in handling large-scale optimization problems relative to other existing methods.

Currently, optimization algorithms include a series of gradient descent-based methods such as ISTA, FISTA, and LISTA. FISTA is an accelerated version of ISTA and is well-known for its superior convergence properties. Specifically, FISTA achieves a convergence rate of $O(1/k^2)$, compared to the ISTA's $O(1/k)$, where k denotes the iteration number [7]. This accelerated convergence makes FISTA efficient for large-scale optimization problems, which are central to our research focus. Additionally, LISTA introduces a learning-based approach to accelerate convergence by optimizing parameters through training data [35]. While LISTA can outperform traditional iterative methods in terms of speed, it requires extensive training data and computational time for parameter learning. Our study emphasizes unsupervised algorithms that do not rely on prior training, aligning with scenarios where training data may not be readily available. Therefore, to focus on more efficient algorithms that do not depend on training data, we exclude LISTA from our comparative analysis.

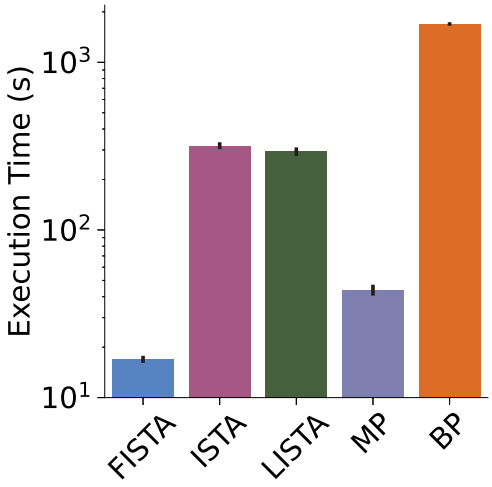


Fig. A.1. Comparison of execution times for various optimization algorithms (FISTA, ISTA, LISTA, MP, BP) in seconds, shown on a logarithmic scale. FISTA demonstrates the shortest execution time, followed by MP, LISTA, ISTA, and BP, which has the longest execution time. Error bars indicate the variability in execution time for each algorithm across multiple runs.

Several algorithms from the field of signal processing and sparse representation, such as Matching Pursuit (MP) and Basis Pursuit (BP), offer alternative approaches to solving optimization problems. However, these algorithms present certain limitations when applied to large-scale problems. MP is a greedy algorithm that iteratively selects dictionary atoms to best match the residual signal. The sequential nature of atom selection can result in significant computational overhead, making it less efficient for high-dimensional data. BP formulates the sparse representation problem as a linear programming task. While precise, linear programming methods can be computationally intensive for large-scale problems, reducing their practicality in such contexts.

Based on these facts, FISTA outperforms these methods in the context of large-scale optimization problems. To demonstrate this, we next apply ISTA, LISTA, MP, BP, and FISTA to the same sparse optimization problem described in Section 4.1, using a consistent dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{7500\times 10000}$ and a regularization parameter $\lambda = 0.02$. The iterations are terminated once when NMSE reached below -12 dB, allowing us to compare the final runtime of each algorithm. As shown in Fig. A.1, FISTA outperforms the other algorithms in terms of runtime.

Furthermore, Evolutionary Algorithms (EAs), including Genetic Algorithm, Differential Evolution, and Particle Swarm Optimization, are heuristic methods widely applied to optimization problems. While they offer versatility, EAs often entail high computational complexity due to the large number of function evaluations required, and they do not guarantee convergence to the global optimum in optimization problems like LASSO. In contrast, FISTA provides a deterministic approach with proven convergence guarantees and utilizes gradient information to accelerate optimization, making it more suitable for large-scale optimization problems where efficiency and reliability are essential. We have applied these EAs along with the FISTA algorithm to the same sparse optimization task described in Section 4.1, using a consistent dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{10\times 40}$ and a regularization parameter $\lambda = 0.1$. As shown in Fig. A.2, under the same termination condition of NMSE below -12 dB, the runtimes of these algorithms are several orders of magnitude longer than that of the FISTA algorithm.

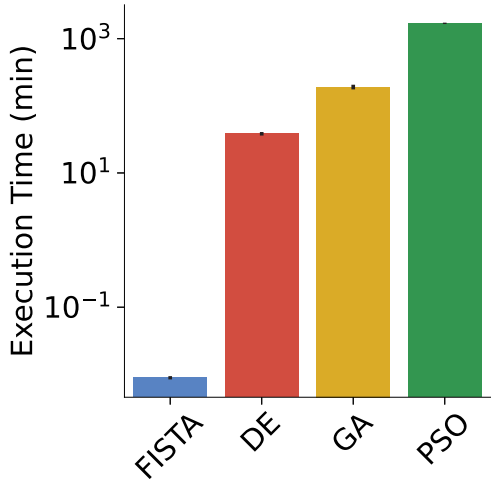


Fig. A.2. Comparison of execution times (in minutes) for different optimization algorithms: FISTA, Differential Evolution (DE), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). The execution time is plotted on a logarithmic scale, indicating that FISTA is significantly faster than the other algorithms. Error bars represent the variability in execution time across multiple runs.

Data availability

All code for synthetic signal generation, as well as the implementations of FISTA, generalized Spiking LCA, and related plotting functions, is available on GitHub at https://github.com/XuexingDu/Spiking-LCA.

References

[1] A. De Maio, Y.C. Eldar, A.M. Haimovich, Compressed Sensing in Radar Signal Processing, Cambridge University Press, 2019.
[2] J. Zhang, B. Chen, R. Xiong, Y. Zhang, Physics-inspired compressive sensing: Beyond deep unrolling, IEEE Signal Process. Mag. 40 (1) (2023) 58–72.
[3] X. Chai, J. Fu, Z. Gan, Y. Lu, Y. Zhang, An image encryption scheme based on multi-objective optimization and block compressed sensing, Nonlinear Dynam. 108 (3) (2022) 2671–2704.
[4] D. Kuzin, O. Isupova, L. Mihaylova, Bayesian neural networks for sparse coding, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 2992–2996.
[5] L. Aitchison, M. Lengyel, With or without you: predictive coding and Bayesian inference in the brain, Curr. Opin. Neurobiol. 46 (2017) 219–227.
[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 689–696.
[7] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[8] J. Xiang, Y. Dong, Y. Yang, FISTA-Net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging, IEEE Trans. Med. Imaging 40 (5) (2021) 1329–1339.
[9] C. Yi, L. Ran, J. Tang, H. Jin, Z. Zhuang, Q. Zhou, J. Lin, An improved sparse representation based on local orthogonal matching pursuit for bearing compound fault diagnosis, IEEE Sens. J. 22 (22) (2022) 21911–21923.
[10] D.J. Field, What is the goal of sensory coding? Neural Comput. 6 (4) (1994) 559–601.
[11] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.
[12] M. Rehn, F.T. Sommer, A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields, J. Comput. Neurosci. 22 (2) (2007) 135–146.
[13] R.N. Sachdev, M.R. Krause, J.A. Mazer, Surround suppression and sparse coding in visual and barrel cortices, Front. Neural Circuits 6 (2012) 43.
[14] M. Zhu, C.J. Rozell, Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system, PLoS Comput. Biol. 9 (8) (2013) e1003191.
[15] V.J. Barranca, D. Zhou, Compressive sensing inference of neuronal network connectivity in balanced neuronal dynamics, Front. Neurosci. 13 (2019) 492216.
[16] V. Boutin, A. Franciosini, F. Chavane, F. Ruffier, L. Perrinet, Sparse deep predictive coding captures contour integration capabilities of the early visual system, PLoS Comput. Biol. 17 (1) (2021) e1008629.
[17] V.J. Barranca, G. Kovačič, D. Zhou, D. Cai, Sparsity and compressed coding in sensory systems, PLoS Comput. Biol. 10 (8) (2014) e1003793.
[18] S. Shapero, M. Zhu, J. Hasler, C. Rozell, Optimal sparse approximation with integrate and fire neurons, Int. J. Neural Syst. 24 (05) (2014) 1440001.
[19] P.T.P. Tang, T.H. Lin, M. Davies, Sparse coding by spiking neural networks: Convergence theory and computational results, 2017, arXiv preprint arXiv:1705.05475.
[20] V.J. Barranca, G. Kovačič, D. Zhou, The role of sparsity in inverse problems for networks with nonlinear dynamics, Commun. Math. Sci. 17 (5) (2019).
[21] V.J. Barranca, Neural network learning of improved compressive sensing sampling and receptive field structure, Neurocomputing 455 (2021) 368–378.
[22] L. Zhang, X. Wei, J. Lu, J. Pan, Lasso regression: from explanation to prediction, Adv. Psychol. Sci. 28 (10) (2020) 1777.
[23] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2) (2005) 301–320.
[24] L. He, Y. Chen, C. Zhong, K. Wu, Granular elastic network regression with stochastic gradient descent, Mathematics 10 (15) (2022) 2628.
[25] M. Davies, N. Srinivasa, T.H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (1) (2018) 82–99.
[26] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G.A.F. Guerra, P. Joshi, P. Plank, S.R. Risbud, Advancing neuromorphic computing with loihi: A survey of results and outlook, Proc. IEEE 109 (5) (2021) 911–934.
[27] K.L. Fair, D.R. Mendat, A.G. Andreou, C.J. Rozell, J. Romberg, D.V. Anderson, Sparse coding using the locally competitive algorithm on the TrueNorth neurosynaptic system, Front. Neurosci. 13 (2019) 754.
[28] Y. Watkins, E. Kim, A. Sornborger, G.T. Kenyon, Using sinusoidally-modulated noise as a surrogate for slow-wave sleep to accomplish stable unsupervised dictionary learning in a spike-based sparse coding model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 360–361.
[29] C. Wu, Y. Xue, H. Bao, L. Yang, J. Li, J. Tian, S. Ren, Y. Li, X. Miao, Forward stagewise regression with multilevel memristor for sparse coding, J. Semicond. 44 (10) (2023) 104101.
[30] C.N. Chou, K.M. Chung, C.J. Lu, On the algorithmic power of spiking neural networks, 2018, arXiv preprint arXiv:1803.10375.
[31] K. Henke, M. Teti, G. Kenyon, B. Migliori, G. Kunde, Apples-to-spikes: The first detailed comparison of LASSO solutions generated by a spiking neuromorphic processor, in: Proceedings of the International Conference on Neuromorphic Systems 2022, 2022, pp. 1–8.
[32] D. Chavez Arana, A. Renner, A. Sornborger, Spiking LCA in a neural circuit with dictionary learning and synaptic normalization, in: Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference, 2023, pp. 47–51.
[33] N. Zins, Neuromorphic Computing Applications in Robotics (Ph.D. thesis), Michigan Technological University, 2023.
[34] D. Chavez Arana, A. Renner, A. Sornborger, A neuromorphic normalization algorithm for stabilizing synaptic weights with application to dictionary learning in LCA, in: Proceedings of the 2022 Annual Neuro-Inspired Computational Elements Conference, 2022, pp. 58–60.
[35] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th international conference on international conference on machine learning, 2010, pp. 399–406.
[36] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[37] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.
[38] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.
[39] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2010) 4–31.
[40] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, Soft Comput. 22 (2) (2018) 387–408.
[41] I.A. Kougioumtzoglou, I. Petromichelakis, A.F. Psaros, Sparse representations and compressive sampling approaches in engineering mechanics: A review of theoretical concepts and diverse applications, Probab. Eng. Mech. 61 (2020) 103082.
[42] L. Wang, L. Zhao, L. Yu, J. Wang, G. Bi, Structured Bayesian learning for recovery of clustered sparse signal, Signal Process. 166 (2020) 107255.
[43] C. Wang, X. Li, K. Xuan, Y. Jiang, R. Jia, J. Ji, J. Liu, Interpolation of soil properties from geostatistical priors and DCT-based compressed sensing, Ecol. Indic. 140 (2022) 109013.
[44] G. Xu, B. Zhang, H. Yu, J. Chen, M. Xing, W. Hong, Sparse synthetic aperture radar imaging from compressed sensing and machine learning: Theories, applications, and trends, IEEE Geosci. Remote. Sens. Mag. 10 (4) (2022) 32–69.
[45] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306.
[46] Y.M. Li, D. Wei, L. Zhang, Double-encrypted watermarking algorithm based on cosine transform and fractional Fourier transform in invariant wavelet domain, Inform. Sci. 551 (2021) 205–227.

[47] T. Ueda, Y. Ohno, K. Yamamoto, A. Iwase, T. Fukuba, S. Hanamatsu, Y. Obama, H. Ikeda, M. Ikedo, M. Yui, et al., Compressed sensing and deep learning reconstruction for women's pelvic MRI denoising: utility for improving image quality and examination time in routine clinical practice, Eur. J. Radiol. 134 (2021) 109430.

[48] B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput. 24 (2) (1995) 227–234.

[49] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, B.A. Olshausen, Sparse coding via thresholding and local competition in neural circuits, Neural Comput. 20 (10) (2008) 2526–2563.

[50] A. Balavoine, J. Romberg, C.J. Rozell, Convergence and rate analysis of neural networks for sparse approximation, IEEE Trans. Neural Netw. Learn. Syst. 23 (9) (2012) 1377–1389.

[51] X. Zhang, L. Yu, G. Zheng, Y.C. Eldar, Spiking sparse recovery with non-convex penalties, IEEE Trans. Signal Process. 70 (2022) 6272–6285.

[52] A. Balavoine, C.J. Rozell, J. Romberg, Convergence of a neural network for sparse approximation using the nonsmooth Łojasiewicz inequality, in: The 2013 International Joint Conference on Neural Networks, IJCNN, IEEE, 2013, pp. 1–8.

[53] L. Chen, Y. Gu, The convergence guarantees of a non-convex approach for sparse recovery, IEEE Trans. Signal Process. 62 (15) (2014) 3754–3767.

[54] S.M. Fosson, A biconvex analysis for lasso ℓ_1 reweighting, IEEE Signal Process. Lett. 25 (12) (2018) 1795–1799.

[55] C. Wang, T. Zhang, X. Chen, S. He, S. Li, S. Wu, BrainPy, a flexible, integrative, efficient, and extensible framework for general-purpose brain dynamics programming, Elife 12 (2023) e86365.

[56] Ş. Mihalaş, E. Niebur, A generalized linear integrate-and-fire neural model produces diverse spiking behaviors, Neural Comput. 21 (3) (2009) 704–718.

[57] K. Tsumoto, H. Kitajima, T. Yoshinaga, K. Aihara, H. Kawakami, Bifurcations in Morris–Lecar neuron model, Neurocomputing 69 (4–6) (2006) 293–316.

[58] X.J. Wang, G. Buzsáki, Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network model, J. Neurosci. 16 (20) (1996) 6402–6413.

[59] L. Calatroni, L.U. Perrinet, D. Prandi, et al., Beyond ℓ_1 sparse coding in V1, PLoS Comput. Biol. 19 (9) (2023) e1011459.

[60] E. Soubies, L. Blanc-Féraud, G. Aubert, A continuous exact ℓ_0 penalty (CEL0) for least squares regularized problem, SIAM J. Imaging Sci. 8 (3) (2015) 1607–1639.

[61] Z. Lu, Iterative reweighted minimization methods for ℓ_p regularized unconstrained nonlinear programming, Math. Program. 147 (1) (2014) 277–307.

[62] A.S. Charles, P. Garrigues, C.J. Rozell, Analog sparse approximation with applications to compressed sensing, 2011, arXiv preprint arXiv:1111.4118.

[63] J. Shen, G. Strang, Asymptotics of daubechies filters, scaling functions, and wavelets, Appl. Comput. Harmon. Anal. 5 (3) (1998) 312–331.

[64] T. Hu, C. Pehlevan, D.B. Chklovskii, A hebbian/anti-hebbian network for online sparse dictionary learning derived from symmetric matrix factorization, in: 2014 48th Asilomar Conference on Signals, Systems and Computers, IEEE, 2014, pp. 613–619.

[65] W. Brendel, R. Bourdoukan, P. Vertechi, C.K. Machens, S. Denéve, Learning to represent signals spike by spike, PLoS Comput. Biol. 16 (3) (2020) e1007692.

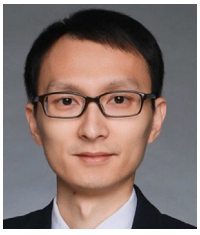
[66] C. Pehlevan, A.M. Sengupta, D.B. Chklovskii, Why do similarity matching objectives lead to Hebbian/anti-Hebbian networks? Neural Comput. 30 (1) (2017) 84–124.

[67] T.H. Lin, P.T.P. Tang, Sparse dictionary learning by dynamical neural networks, in: International Conference on Learning Representations, 2019.

[68] C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz, et al., Generalized leaky integrate-and-fire models classify multiple neuron types, Nat. Commun. 9 (1) (2018) 709.

Xuexing Du earned his B.S. in Mathematics from Shanghai Jiao Tong University in 2018 and is currently pursuing a Ph.D. in Applied Mathematics at the same institution. His research focuses on the dynamical modeling of neurons and neural networks, leveraging statistical machine learning methods for parameter estimation in complex models, as well as the efficient solution of large-scale optimization problems.

Zhong-qi K. Tian earned his B.S. in Mathematics (2014) and Ph.D. in Applied Mathematics (2020) from Shanghai Jiao Tong University. His research interests include causal inference, the development of efficient computational methods for large-scale cortical networks, and the exploration of mechanisms underlying brain information processing.



Songting Li is a Professor at the Institute of Natural Sciences and the School of Mathematical Sciences at Shanghai Jiao Tong University. He earned both his B.S. (2010) and Ph.D. (2014) in Mathematics from Shanghai Jiao Tong University before serving as a Postdoctoral Researcher at the Courant Institute of Mathematical Sciences, New York University, from 2015 to 2018. His research has been featured in prestigious journals such as *PNAS*, *CPAM*, *Cerebral Cortex*, *Cell Reports*, and *NeurIPS*. His academic interests span applied mathematics and computational neuroscience, with a particular focus on dendritic computation in neurons, the structure and dynamics of neural networks, and the development of brain-inspired algorithms.

Douglas Zhou is a Distinguished Professor at the Institute of Natural Sciences and the School of Mathematical Sciences at Shanghai Jiao Tong University. He earned both his B.S. (2002) and Ph.D. (2007) in Mathematics from Peking University before conducting postdoctoral research at the Courant Institute of Mathematical Sciences, New York University, from 2007 to 2009. His research has been featured in prestigious journals such as *CAMP*, *PNAS*, *Physical Review Letters*, and *NeurIPS*. His academic interests focus on computational neuroscience, including the mathematical modeling and simulation of neuronal network dynamics, the development of efficient computational methods for large-scale cortical networks, the exploration of mechanisms underlying brain information processing, and the creation of novel mathematical tools to analyze experimental neurophysiological data.