Alternating Excitation-Inhibition Dendritic Computing for Classification

Jiayi Li, Zhenyu Lei, Zhiming Zhang, Haotian Li, Yuki Todo, and Shangce Gao, *Senior Member, IEEE*

***Abstract*—The addition of dendritic inhibition has been shown to significantly enhance the computational and representational capabilities of neurons. However, this inhibitory mechanism is mostly ignored in existing artificial neural networks (ANNs). In this paper, we propose the alternating excitatory and inhibitory mechanisms and use them to construct an ANN-based dendritic neuron, the alternating excitation-inhibition dendritic neuron model (ADNM). Subsequently, a comprehensive multilayer neural system named the alternating excitation-inhibition dendritic neuron system (ADNS) is constructed by networking multiple ADNMs. To evaluate the performance of ADNS, a series of extensive experiments are implemented to compare it with other state-of-the-art networks on a diverse set consisting of 47 feature-based classification datasets and 2 image-based classification datasets. The experimental results demonstrate that ADNS outperforms its competitors in classification tasks. In addition, the impact of different hyper-parameters on the performance of the neural model is analyzed and discussed. In summary, the study provides a novel dendritic neuron model with better performance and interpretability for practical classification tasks.**

***Impact Statement*—The novel dendritic neuron model has become a prominent research direction in the field of artificial intelligence, showing significant advantages over the commonly used McCulloch-Pitts neuron model in deep learning. It exhibits characteristics more akin to biological neurons and possesses enhanced non-linear computational capabilities. In this study, we propose the alternating excitation-inhibition dendritic neuron model (ADNM) as an improvement upon the dendritic neuron model (DNM). ADNM offers two key advantages over DNM:**

1) **ADNM introduces the alternating excitation-inhibition (AEI) mechanism, which incorporates the excitatory and inhibitory characteristics of dendrites in biological neurons. This mechanism allows ADNM to assign excitatory or inhibitory signals to input features based on their properties, thereby enhancing the model's feature processing and computational abilities.**
2) **DNM cannot construct multilayer networks due to the vanishing gradient issue caused by cumulative multiplication in its dendritic layer, ADNM addresses this limitation by employing the AEI mechanism and gains the ability to construct multilayer networks.**

Furthermore, we demonstrate the network composition capability of ADNM by combining multiple ADNMs to form an Alternating Excitation-Inhibition Dendritic Neuron System (ADNS). Extensive experiments on various classification datasets validate the robust classification performance of ADNS.

***Index Terms*—dendritic neuron model, classification, neural system, novel neuron, neural network, deep learning**

I. Introduction

Inhibition has been widely acknowledged in the field of neuroscience as a crucial element of normal neuronal function [1]. Various studies have demonstrated that inhibition plays a pivotal role in regulating the information flow between neurons, maintaining the robustness of neural circuits, and participating in cognitive functions such as memory, learning, attention, and emotion [2], [3]. Dendritic inhibition refers to the mechanism by which signals are suppressed from dendrites to the neuronal soma, thereby regulating their response to excitatory inputs [4]. The integration of dendritic inhibition has been shown to significantly enhance the computational and representational capabilities of neurons [5], [6]. Computational simulations have illustrated that dendritic inhibition can augment the plasticity of excitatory synapses and provide new insights into how neurons modulate their responses through dendritic inhibition [7]. In spiking neural networks (SNNs), which closely mimic biological neurons, there is a growing trend of integrating excitatory and inhibitory synaptic mechanisms [8], [9].

The role of dendritic inhibition in the realm of SNN has been well-established. It has been demonstrated that integrating excitatory and inhibitory mechanisms within SNNs can significantly enhance the performance of classification tasks [10]. Some studies have demonstrated that feedback inhibition mechanisms can be used to improve the accuracy and speed of visual information processing [11]. Moreover, the introduction of fast inhibition mechanisms in SNNs has been shown to effectively suppress the interactions between neurons, thereby improving the performance of classification and recognition tasks, as demonstrated in some recent studies [12], [13]. Some studies have employed rudimentary hardware for simulating excitatory and inhibitory connections [14].

In the field of deep learning based on artificial neural networks (ANNs), numerous mechanisms exist that are capable of significantly enhancing model performance, such as increasing the depth of neural networks, incorporating residual connections [15] utilizing dropout [16], and introducing attention mechanisms [17]. However, they all construct networks based on McCulloch-Pitts neurons that are overly simplistic and make it difficult to simulate the excitatory and inhibitory mechanisms within dendrites. Hence, they just provide distinct connection patterns for networks and don't discuss the effect

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP23K24899, and Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145. *(Corresponding authors: Zhenyu Lei, Yuki Todo, and Shangce Gao)*

J. Li, Z. Lei, Z. Zhang, H. Li, and S. Gao are with the Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan. (E-mail: lijiayi212@gmail.com; leizystu@outlook.com; zhangzm0128@163.com; 1591688699lht@gmail.com; gaosc@eng.u-toyama.ac.jp)

Y. Todo is with the Faculty of Electrical, Information and Communication Engineering, Kanazawa University, Ishikawa 9201192, Japan (E-mail: yktodo@se.kanazawa-u.ac.jp)

of neuronal inhibition mechanisms. Although there are some studies on dendritic neuron models based on ANNs, most of them enhance the non-linear processing capabilities of dendritic layers by introducing simple mathematical functions [18] such as sigmoid, ReLU, and multiplication. Despite the considerable performance gains facilitated by these mathematical functions, they also introduce challenges such as poor interpretability, training difficulties, and network composition issues [19], [20].

Motivated by the aforementioned findings, we integrate for the first time the mechanisms of alternating excitatory and inhibitory signals in dendrites into neurons based on ANNs, which are capable of more truly reflecting the transformation process of signals resulting in enhancing the performance and robustness. In essence, we redesign dendritic neuron models based on ANNs and propose ADNM, utilizing the tanh function to simulate the alternating excitatory and inhibitory signals within dendrites. Our enhancements effectively address the challenges of training difficulties and network composition issues encountered in existing dendritic neuron models. Multiple ADNMs can be employed to construct multilayer neural systems termed ADNS. Furthermore, the proposed model can be integrated into deep neural networks, thereby enhancing their performance by introducing inhibitory mechanisms to neurons. Extensive comparative experiments on synthetic datasets, real-world datasets, and image classification datasets validate the effectiveness of the proposed dendritic inhibition model [21]. Results demonstrate superior performance of ADNS in classification tasks, highlighting the enhanced robustness and flexibility conferred by dendritic inhibition models. Additionally, we analyze the impact of different parameters on model performance, leading to discussions on the interpretability of neural systems and providing new insights into understanding their operational mechanisms.

In summary, this paper proposes ADNM and uses several ADNMs to form a more complex neural system. Experiments show that the proposed neural system can provide better performance and interpretability for classification tasks in practical applications. The research contributes to the ongoing development and advancement of deep learning technology, providing a new perspective and potential avenues for future research in this field. The paper proposes an improved dendritic neuron model and demonstrates its superior performance. The contributions of the study are mainly reflected in the following three aspects:

1) The novel tanh-activation parameters are proposed to implement the mechanism of alternating excitation and inhibition of dendrites for the traditional dendritic neuron model. This improvement enhances the performance and robustness of the neuron model, enabling it to better handle complex input data. Moreover, it has better interpretability, facilitating the understanding of the internal workings of the neural system.
2) Multiple ADNMs are employed in this paper to build multilayer neural systems, termed ADNS. Furthermore, ADNS can also be integrated into deep learning networks. Our proposed model is evaluated on synthetic datasets, real-world datasets, and image classification datasets. The results indicate that the proposed ADNS outperforms state-of-the-art models in classification tasks.
3) The article further investigates the influence of different parameters on the performance of the neuron model and discusses the interpretability of the neural system. These research results help to deepen the understanding of the internal structure and operation mechanism of the neuron model, which provides better performance and interpretability for classification tasks in practical applications.

The organization of this paper is summarized as follows: Section II introduces the background of this study. A traditional dendritic neuron model is described in Section III. The novel neuron and neural system are given in Section IV. The detailed experimental results are shown in Section V. Finally, Section VI concludes this paper.

II. Related Work

Dendritic computation is a relatively new research field that aims to study the computational capabilities of dendrites, which are the tree-like structures that receive inputs from other neurons in biological neural networks. Researchers have shown that dendrites can perform complex computations, such as coincidence detection, spike-timing-dependent plasticity, and nonlinear integration of synaptic inputs [22]–[24]. Therefore, incorporating dendritic computation into artificial neural networks can enhance their computational capabilities, especially in handling complex and high-dimensional data [25]. In recent years, there have been numerous studies on dendritic neuron models. They are widely used in the field of artificial neural networks due to their ability to simulate the complex behavior of biological neurons [12]. These tree-based neurons have acquired focus due to their biological plausibility and potential for improving performance of deep learning models. In these models, the input signals are processed by tree-structured dendrites that converge onto a single neuron body.

A typical dendritic neuron module consists of multiple dendritic trees, where each tree is composed of dendritic segments and synapses [6], [26], [27]. The dendritic segments are connected by synapses, which can either be excitatory or inhibitory. The input signal is transmitted through the synapses and undergoes nonlinear processing at each dendritic segment. The output from each segment is then integrated at the neuron soma to produce the final output. Spratling and Johnson proposed a DNM with a sigmoid function to simulate dendritic inhibition [5]. However, this model still has some limitations in terms of its ability to handle complex input signals and its computational efficiency. To overcome these limitations, several improved dendritic neuron models have been proposed in recent years. For example, Wu *et al.* propose a new DNM [6] that simulates the dendritic structure with a two-layer fully connected structure, which performs a linear weighted summation of the sparsely connected inputs in the first layer and a nonlinear transformation of the results of the previous layer in the second layer to simulate the nonlinear output of the dendrites. Some researchers based on lattice algebra to model

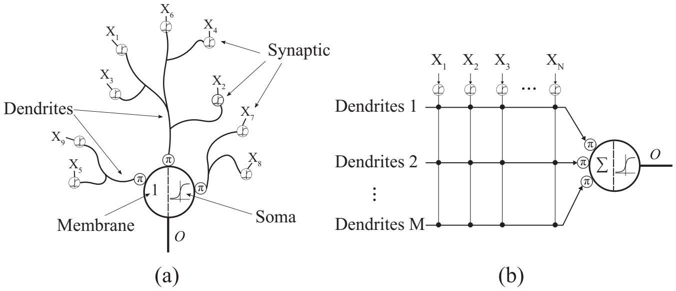


Fig. 1: Dendritic neuron model: (a) representation of the biological architecture, (b) depiction of the ideal complete connection artificial architecture.

the process of dendritic computation and used evolutionary algorithms to train the developed dendritic neuron models [27]. Since the exact principle of dendritic computation has not been studied clearly yet. It has also been proposed to use binary trees to model nonlinear computation of dendrites [26]. These k-trees, which are composed of *k* binary trees, show promising performance in certain machine learning tasks. Nevertheless, it is important to note that the internal binary tree structure of k-trees differs significantly from the dendritic structure observed in real neurons. Peng *et al.* enhanced the computational power of the neural P system, an SNN-based computational model, by incorporating dendritic computation [23]. Additionally, some biologists have proposed dendritic neuron models applicable to both regression and classification tasks, drawing inspiration from pyramidal neurons and developing novel learning algorithms [28].

Todo *et al.* proposed a biologically-inspired DNM with a more complete structure of a real neuron, including synapses, dendrites, cell membranes, and soma [29]. DNM can be utilized for learning in various ways, including unsupervised [29] and supervised [30], [31] learning. In supervised learning, DNM can be trained using different algorithms, such as evolutionary algorithms [19] and backpropagation algorithms, depending on the specific problem. By integrating DNM with convolutional neural networks (CNNs) [20], DNM has been shown to outperform traditional CNNs in practical image recognition problems [15]. In the medical image segmentation domain, DNM can serve as convolutional counterparts, thereby enhancing segmentation performance [32]. Furthermore, DNMs have demonstrated success when combined with vision Transformer in the field of image classification [33]. Moreover, DNM can be extended from the real-valued domain to the complex-valued domain [34], which has broad applications. Despite the variety of DNMs proposed to address real-world problems, there remains a lack of research concerning the incorporation of inhibitory mechanisms and the exploration of multilayer DNM structures.

III. Preliminaries

In this paper, inspired by previous studies [29], an improved version of the dendritic neuron model is proposed. In this section, the fundamental principles of DNM are concisely described, which comprises four main components: synaptic layer, dendritic layer, cell membrane layer, and soma layer. The synaptic layer is responsible for transmitting nerve impulses across synapses. The dendritic layer characterizes the spatial structure of dendritic branches and their mechanisms of integrating nerve impulses. The cell membrane layer and soma model describe the dynamics of the neuronal membrane potential and the generation of action potentials. These interconnected components together form a comprehensive and biologically plausible neural model. Its biological architecture and analog artificial architecture are illustrated in Fig. 1.

Biologically, synapses are used to receive signals from the axons of presynaptic cells, and the synaptic layer in DNM is also used to receive input data. The input data can be represented as x_i $(i = 1, 2, ..., N)$, where N is the dimension of the input data. DNM uses a sigmoid function to represent the synaptic layer. The calculation of the synaptic layer can be expressed as follows:

$$S_{ij} = \frac{1}{1 + e^{-K(w_{ij}x_i - q_{ij})}} \tag{1}$$

where S_{ij} denotes the result of the signal sent by the axon of the i-th presynaptic cell after being received by the synapse on the j-th $(j = 1, 2, ..., M)$ dendrite on the postsynaptic cell, M is the number of dendrites, e is the base of the natural logarithms, K is the slope parameter of the input transform, w_{ij} and q_{ij} are the trainable parameters of DNM. For the trained DNM, its synaptic layer can be represented as four different states according to weights when the input is mapped to 0 to 1. (1) Constant 0 connection, when $0 < w_{ij} < q_{ij}$ or $w_{ij} < 0 < q_{ij}$, which means that the output is always approximately 0 regardless of the input. (2) Constant 1 connection, when $q_{ij} < 0 < w_{ij}$ or $q_{ij} < w_{ij} < 0$, which means that the output is always approximately 1 regardless of the input. (3) Excited connection, when $0 < q_{ij} < w_{ij}$, the output is always proportional to the input. (4) Inhibitory connection, when $w_{ij} < q_{ij} < 0$, the output is always inversely proportional to the input. These four connection instances are vital to inferring the morphology of a neuron by specifying the positions and synapse states of dendrites

The dendritic layer is used to summarize the signals of all synapses on one dendrite, and each dendrite is computed using multiplication, a simple nonlinear arithmetic. Since the synaptic signals are restricted to between 0 and 1 in DNM, the dendritic layer approximates a logical AND operation. It is formulated as follows:

$$D_j = \prod_{i=1}^{N} S_{ij}. \tag{2}$$

The membrane layer uses accumulation to summarize the signals coming from all dendrites. Approximates a logical or operation. The signals summarized by the membrane layer will be passed to the soma. It is expressed as:

$$V = \sum_{j=1}^{M} D_j. \tag{3}$$

The soma is the information processing unit of the DNM, and the signals transmitted from the membrane layer are

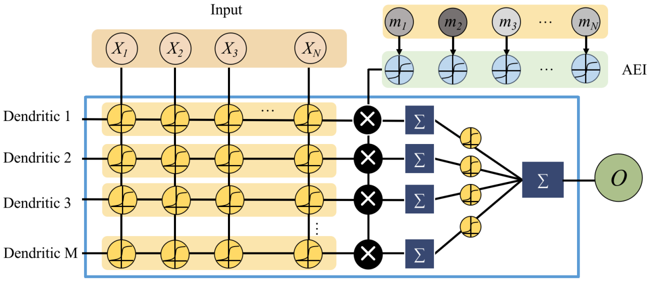


Fig. 2: Structural illustration of single ADNM.

processed by a sigmoid function. Depending on the output of the model it is decided whether the neuron is excited or not. It is expressed as:

$$O = \frac{1}{1 + e^{-K(V-Q)}} \tag{4}$$

where K and Q are set constants. According to [30], Q takes values ranging from 0 to 1 and indicates the threshold value in the soma.

By analyzing the overall structure of DNM, we can observe two key limitations. Firstly, due to the use of the sigmoid function as the activation function, DNM lacks the inhibitory effect on input signals. Secondly, the multiplication used in the dendritic layer of DNM leads to the vanishing gradient problem during backpropagation, making it challenging to form multilayer networks.

IV. The proposed ADNM and ADNS

A. Alternating Excitation-Inhibitory Dendritic Neuron Model

The alternating excitation and inhibition dendritic neuron model (ADNM) retains the original hierarchical structure of the DNM while incorporating the alternating excitation and inhibition (AEI) mechanism to better extract information from the dendritic layer. In addition, the dendritic layers of DNMs contain multiplicative operations, which can lead to gradient explosion or gradient vanishing problems in backpropagation for multilayer DNMs [35]. ADNM eliminate multiplicative operations, which makes it possible to construct complex networks using multiple layers of ADNM. The structure of ADNM is illustrated in Fig. 2.

It is worth noting that we remove the hyperparameter K of the DNM synaptic layer, because Kw is equivalent to w when $w \in R$. Then, (1) is modified as follows:

$$S_{ij} = \frac{1}{1 + e^{-(w_{ij}x_i - q_{ij})}}. \tag{5}$$

A novel AEI mechanism is designed to enhance the processing of input information for the dendritic layer. The AEI mechanism is a set of weights normalized to between -1 and 1 by the tanh function, which is used to simulate the excitatory and inhibitory effects of dendrites on input information. The formula is as follows:

$$A_i = \tanh(m_i) \tag{6}$$

where m_i are trainable parameters, A_i represents the excitatory inhibitory effect of the dendrites on the input x_i. Its positive numbers represent excitatory effects, while negative numbers represent inhibitory effects. The magnitude of the value determines the degree of excitatory inhibition. Unlike other commonly used activation functions such as ReLU and sigmoid, tanh has an output range of -1 to 1, which makes it a better choice for simulating inhibition effects [36].

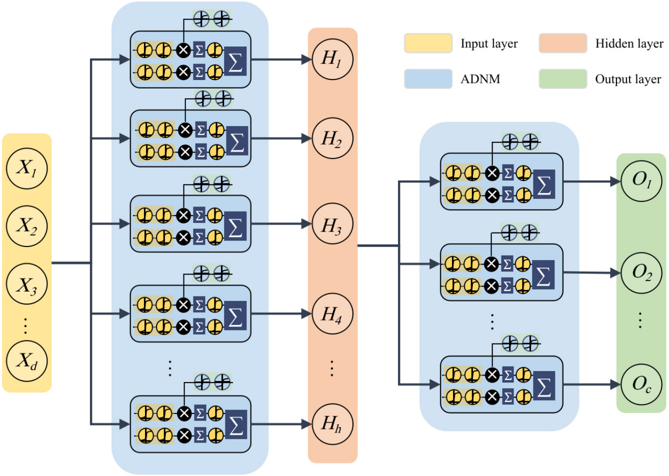


Fig. 3: Structural illustration of the ADNS.

In the dendritic layer, the result S_{ij} obtained from the synaptic layer is multiplied bitwise by A_i, which represents the AEI mechanism acting on the dendrites. Then, the cumulative calculation of all synapses in the dendritic layer simulates the process of combining excitatory and inhibitory signals in biological neurons. The formula is as follows:

$$D_j = \sum_{i=1}^{N} (S_{ij} A_i). \tag{7}$$

In ADNM, the membrane layer and the soma layer are merged. A sigmoid function is used to perform a nonlinear transformation as the information from each dendrite is passed into soma layer. The soma layer accumulates the information obtained from all dendrites as the output of the model. The modification changes the output of each neuron from $[0, 1]$ to $[0, M]$, facilitating a clearer integration of each dendrite's output into the neuron's overall output. This adjustment also simplifies the process of constructing classifiers using multiple neurons in subsequent stages. It is described as follows:

$$O = \sum_{j=1}^{M} \frac{1}{1 + e^{(k_j D_j - \theta_j)}} \tag{8}$$

where k_j and θ_j are designated as learnable parameters. In the context of the DNM, determining hyperparameters for the soma layer presents substantial challenges, as their optimal values vary considerably depending on the specific problem. Therefore, we propose treating the hyperparameters of the soma layer as learnable parameters. Experimental evaluations demonstrate that this modification yields highly favorable outcomes.

B. Back-propagation of ADNM

ADNM can be trained with a cross-entropy loss function. In this section, the back-propagation (BP) process of ADNM based on the cross-entropy loss function is provided. First, the cross-entropy function L can be expressed as:

$$L = -y\log(O) - (1-y)\log(1-O) \tag{9}$$

where O represents the predicted output of ADNM and y is the actual label.

According to the chain rule, the gradient of each parameter is expressed as the product of the loss function for each part of that parameter. For the weights w_{ij}, q_{ij}, m_i, k_j, and θ_j, their gradients is calculated as follows:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial D_j}\frac{\partial D_j}{\partial S_{ij}}\frac{\partial S_{ij}}{\partial w_{ij}} \tag{10}$$

$$\frac{\partial L}{\partial q_{ij}} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial D_j}\frac{\partial D_j}{\partial S_{ij}}\frac{\partial S_{ij}}{\partial q_{ij}} \tag{11}$$

$$\frac{\partial L}{\partial m_i} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial D_j}\frac{\partial D_j}{\partial A_i}\frac{\partial A_i}{\partial m_i} \tag{12}$$

$$\frac{\partial L}{\partial k_j} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial k_j} \tag{13}$$

$$\frac{\partial L}{\partial \theta_j} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial \theta_j} \tag{14}$$

Then, the gradient of each component is required. The gradient calculation with respect to the output O of the loss function is described as follows:

$$\frac{\partial L}{\partial O} = -\frac{y}{O} + \frac{1-y}{1-O}. \tag{15}$$

The gradient of the output O with respect to D_j is required. O consists of a sigmoid function on D_j, so the derivative of O with respect to D_j is expressed as:

$$\frac{\partial O}{\partial D_j} = \sum_{j=1}^{M} \frac{e^{-(k_j D_j - \theta_j)}}{(1 + e^{-(k_j D_j - \theta_j)})^2} \cdot k_j \tag{16}$$

The gradient of O for k_j and θ_j can be calculated as:

$$\frac{\partial O}{\partial k_j} = \sum_{j=1}^{M} \frac{e^{-(k_j D_j - \theta_j)}}{(1 + e^{-(k_j D_j - \theta_j)})^2} \cdot (-D_j) \tag{17}$$

$$\frac{\partial O}{\partial \theta_j} = \sum_{j=1}^{M} \frac{e^{-(k_j D_j - \theta_j)}}{(1 + e^{-(k_j D_j - \theta_j)})^2} \cdot 1 \tag{18}$$

Then the gradient of D_j with respect to S_{ij} is calculated. D_j is a linear function where the coefficients are S_{ij}, so the derivative of D_j with respect to S_{ij} is A_i, which is:

$$\frac{\partial D_j}{\partial S_{ij}} = A_i \tag{19}$$

The gradient of S_{ij} against w_{ij} is expressed as:

$$\frac{\partial S_{ij}}{\partial w_{ij}} = x_i \cdot S_{ij}(1 - S_{ij}) \tag{20}$$

Calculating the gradient of S_{ij} against q_{ij} by:

$$\frac{\partial S_{ij}}{\partial q_{ij}} = -S_{ij}(1 - S_{ij}) \tag{21}$$

The gradient of D_j for A_i can be calculated as:

$$\frac{\partial D_j}{\partial A_i} = S_{ij} \tag{22}$$

Finally, the gradient of A_i against m_i is calculated as follows:

$$\frac{\partial A_i}{\partial m_i} = 1 - \tanh^2(m_i) \tag{23}$$

In summary, the above formulas are used to calculate the gradient of the loss function for each parameter. The gradient can be obtained by substituting them into the first set of formulas, which updates each parameter in the ADNM.

C. Alternating Excitation-Inhibitory Dendritic Neuron System

To solve multi-classification problems using DNM, a single-layer multi-DNM structure with the same number of DNMs as the number of classifications is required [37]. Like the traditional DNM, the ADNM produces a 1-dimensional output and is also a single-neuron model. Therefore, multiple ADNMs are necessary to form a neural system when solving multi-classification problems. Additionally, the ADNM is easier to learn using back-propagation compared to DNM, which enables it to build a multilayer structure and further improve the computational power of the model. This improvement will also be verified in the ablation experiments.

Fig. 3 illustrates a two-layer ADNS, where the input data is fed to the ADNM unit in the first layer. The first layer produces outputs (H_1, H_2, ..., H_h), which are then fed as inputs to the second layer after processing. Here, h refers to the number of ADNM units in the first layer. The second layer contains c ADNM units, and produces an output of length c. In the context of multi-class classification problems, c represents the number of classes. The ADNS can be extended to multiple layers with varying numbers of ADNM units to solve different problems.

The ADNS bears resemblance to a multilayer perceptron (MLP) in terms of structure, with the number of layers and ADNM units being adjustable to suit different problems. The ADNS, however, has dendritic processing capabilities that allow it to integrate information from multiple inputs using a synaptic matrix. This enables the ADNS to produce outputs that are influenced by both excitatory and inhibitory signals, making it capable of capturing complex patterns and relationships in the input data.

In this study, BP algorithm is utilized to train the ADNS. The training process involves minimizing the multi-class cross-entropy loss function L, which measures the difference between the predicted output of the network and the actual labels of the training data. The multi-class cross-entropy loss function is calculated as follows:

$$L = -\frac{1}{n}\sum_{v=1}^{n}\sum_{k=1}^{c} y_{vk}\log(O_{vk}) \tag{24}$$

TABLE I: The details of 47 synthetic and real classification datasets.

		Dataset Name	d	c	n_{tr}	n_{val}	n_{te}		Dataset Name	d	c	n_{tr}	n_{val}	n_{te}
Synthetic	F1	Concentric	2	3	1164	291	162	F2	Horseshoes	2	2	1080	270	150
	F3	Moons	2	2	1080	270	150	F4	Ripley Dataset	2	2	800	200	250
	F5	Three Gaussians	2	3	1152	288	360	F6	Three Spirals	2	3	1080	270	150
	F7	Two Gaussians	2	2	768	192	240	F8	Two Spirals	2	2	1080	270	150
	F9	XOR Problem	2	2	1008	252	140							
Small-scale	F10	Iris Data	4	3	108	27	15	F11	Thyroid Gland Data	5	3	155	39	21
	F12	BUPA Liver Disorders	6	2	244	62	35	F13	Car Evaluation	6	4	1244	311	173
	F14	Seeds	7	3	143	36	20	F15	Pima Indians Diabetes	8	2	552	139	77
	F16	Breast Cancer Wisconsin Original	9	2	491	123	69	F17	Echocardiogram Data	9	2	44	11	6
	F18	Glass Identification	9	6	153	39	22	F19	Tic-tac-toe	9	2	460	115	383
Medium-scale	F20	Indian Liver Patient Dataset	10	2	417	105	57	F21	Vowel Recognition Data	11	11	712	179	99
	F22	Heart Disease Cleveland	13	2	213	54	30	F23	Wine Recognition Data	13	3	128	32	18
	F24	Exactly	13	2	480	120	400	F25	Leaf	14	30	244	62	34
	F26	Australia	14	2	386	97	207	F27	Credit Approval	15	2	469	118	66
	F28	US Congressional Voting Records	16	2	167	42	23	F29	CongressEW	16	2	243	61	130
	F30	Vote	16	2	144	36	120	F31	Climate Model Simulation Crashes	18	2	388	98	54
	F32	Hepatitis Domain	18	2	80	21	11	F33	Statlog (Image Segmentation)	18	7	1663	416	231
	F34	Diabetic Retinopathy Debrecen	19	2	828	208	115							
Large-scale	F35	Parkinsons Dataset	22	2	140	35	20	F36	SpectEW	22	2	64	16	187
	F37	German	24	2	560	140	300	F38	Steel Plates Faults	27	7	1396	350	195
	F39	Breast Cancer Wisconsin Diagnosis	30	2	409	103	57	F40	Ionosphere	33	2	252	63	36
	F41	Dermatology	34	6	257	65	36	F42	KrVsKpEW	36	2	1534	384	1278
	F43	SPECTF Heart Dataset	44	2	192	49	26	F44	SonarEW	60	2	112	29	67
	F45	Mice Protein Expression	80	8	397	100	55	F46	Hillvalley	100	2	678	170	364
	F47	Musk	166	2	266	67	143							

where n denotes the number of samples, c denotes the number of classes, y_{vk} represents the label (0 or 1) that indicates whether the actual class of sample v is k, and O_{vk} denotes the predicted output of the network, such as the probability that the v-th sample is predicted as class k.

V. Experiment

A. Data Description

The datasets used in this study are divided into three main categories: synthetic datasets, real datasets, and image classification datasets.

Synthetic Datasets: The synthetic datasets are artificially generated to simulate specific classification scenarios. These datasets are crafted to emulate diverse classification problem scenarios to evaluate the performance of the models.

Real Datasets: Real datasets are sourced from the UCI Machine Learning Repository, spanning multiple domains such as medicine, environmental science, and robotics. These datasets have been extensively employed in evaluating machine learning algorithms for classification tasks, owing to their diversity and real-world relevance. In order to evaluate the algorithms proposed in this paper more comprehensively, we further subdivided them according to the dimensional scales of the features of real datasets, including **small-scale**, **medium-scale** and **large-scale** datasets.

Image Classification Datasets: Image Classification Datasets contains two large-scale datasets on image classification tasks CIFAR-10 and CIFAR-100. These datasets are renowned benchmarks in the field of computer vision, providing a rich array of images across multiple classes for evaluating the efficacy of deep learning models.

The details of 47 synthetic and real classification datasets are listed in Table I. Each dataset is divided into training, validation, and test sets according to the partitioning method recommended by the original dataset authors. We record the data dimensions (d), the number of output classes (c), the number of instances in the training set (n_{tr}), the number of instances in the validation set (n_{val}), and the number of instances in the test set (n_{te}) for each dataset.

B. Experimental Details

The experiments are conducted on a computer equipped with an Intel i9-12900K CPU and an NVIDIA RTX3090 GPU running Windows 11 operating system. The ADNS and the 9 other comparative models are implemented using Python 3.8 and PyTorch 1.11. In the experiments, all models are trained for 20000 epochs with a learning rate of 0.001, cross-entropy is used as the loss function. We publish the code on the following website: https://toyamaailab.github.io/sourcedata.html.

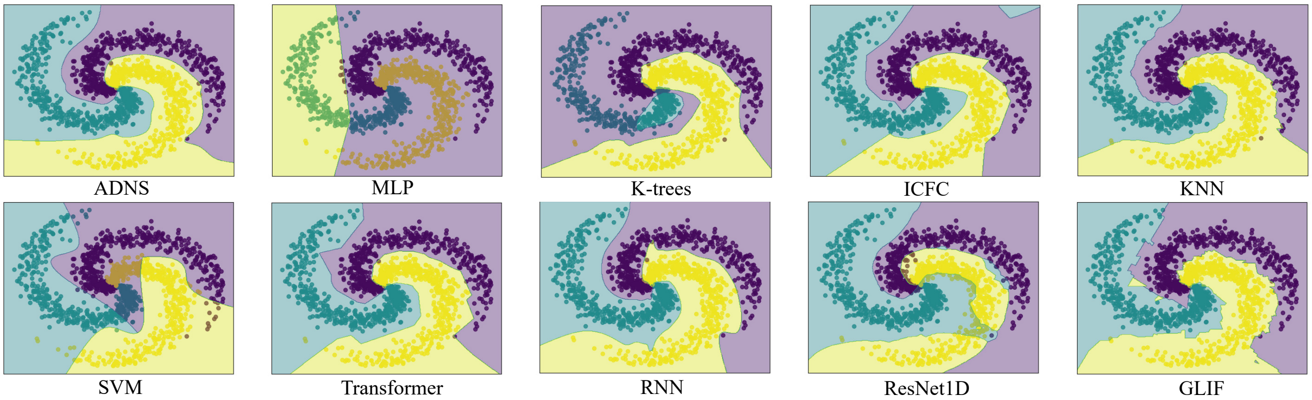
The comparisons between ADNS and the 9 other models are implemented to evaluate the performance of the proposed ADNS in classification tasks. These comparative models includes MLP, k-trees [26], IC fully connected (ICFC) [38], k-nearest neighbors (KNN) [39], support vector machine (SVM) [40], Transformer [17], RNN [35], ResNet1D [41] and GLIF [42]. These compared models include dendritic neurons, spiking neurons, and various neural architectures in deep learning.

During the model training process, early stopping strategy is designed, which means that the training of the model is stopped when the loss function on the validation set do not decrease for 200 consecutive epochs. All the parameters of the model are randomly initialized using a uniform distribution. Additionally, the all experiments are repeated 10 times to avoid the impact of randomness on the experimental results.

The comprehensive classification results are presented in Table II, with the best-performing results highlighted in bold and the next best results underlined. ADNS consistently achieves the top performance across both synthetic datasets and real datasets of varying scales, demonstrating its adaptability to input dimensions of different magnitudes. Specifically, in CIFAR-10 and CIFAR-100 datasets, all models utilize VGG13 as the feature extractor for a fair comparison, yet ADNS outperforms other models, showcasing its superior performance even in comparison to established architectures.

TABLE II: The classification results of all models, synthetic dataset with three scales of real dataset are the average of the classification results of all datasets therein. Detailed results are listed in Supplementary File.

	ADNS	MLP	k-trees	ICFC	KNN	SVM	Transformer	RNN	ResNet1D	GLIF
Synthetic	**94.00%**	63.52%	89.30%	92.98%	93.11%	73.71%	93.58%	93.29%	89.39%	76.67%
Small-scale	**90.37%**	70.17%	74.17%	87.41%	85.24%	88.02%	88.63%	87.65%	85.83%	71.31%
Medium-scale	**91.41%**	70.83%	72.30%	89.13%	82.54%	85.94%	89.50%	87.05%	85.80%	78.05%
Large-scale	**84.74%**	63.91%	71.78%	78.43%	71.14%	71.75%	81.16%	80.44%	77.93%	78.42%
CIFAR10	**93.53%**	93.46%	57.84%	92.92%	28.80%	37.40%	91.58%	68.22%	92.51%	91.68%
CIFAR100	**71.48%**	70.72%	5.56%	60.46%	19.60%	19.50%	69.24%	28.57%	70.46%	69.84%



ADNS MLP K-trees ICFC KNN

SVM Transformer RNN ResNet1D GLIF

Fig. 4: Decision boundary plots of ADNS and 9 comparison models on Three Spirals datasets.

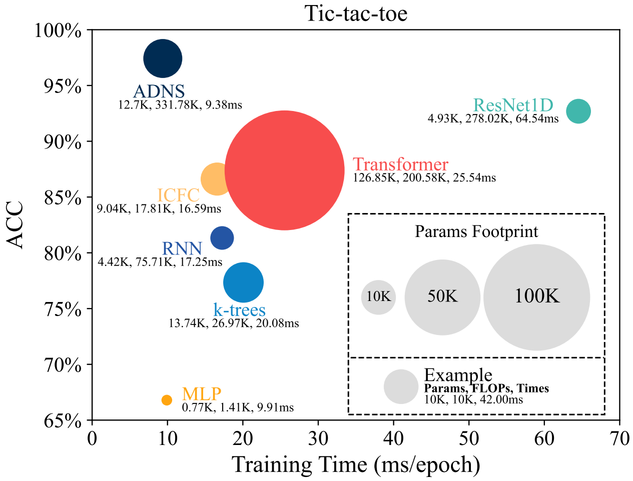


Fig. 5: Efficiency Comparison in Tic-tac-toe Dataset.

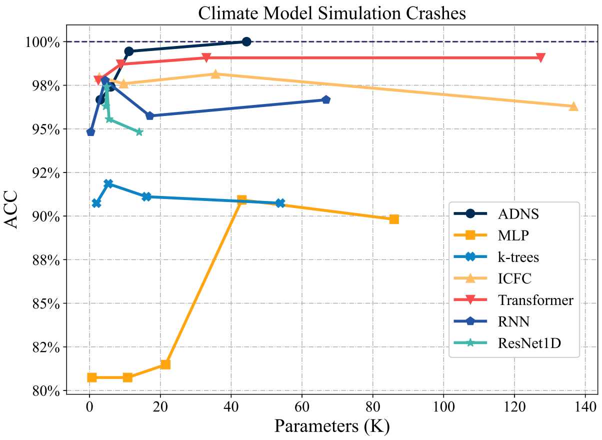


Fig. 6: Comparison of Increasing Model Parameter Size on the Climate Model Simulation Crashes Dataset.

The decision boundary plot is a visualization technique used to display the classification decision boundary of a classifier for different categories of data in feature space. To visually compare the performance of the ADNS with other benchmark models in terms of classification decision, the Three Spirals dataset is chose as an example. An illustration of the decision boundary for each model is plotted in Fig. 4. From it, it can be see that for the Three Spirals dataset, the decision boundary of the ADNS more accurately separates the dataset into three spiral-shaped categories compared to the other seven benchmark models. Additionally, the results imply that SVM and MLP perform poorly in classifying the Three Spirals dataset, which further confirms the superiority of deep learning algorithms in handling complex nonlinear data. Compared to other neuronal models like k-trees, ICFC, Transformer, RNN, ResNet1D and GLIF, the ADNS exhibits smoother decision boundaries and achieves higher accuracy in classifying specific values.

We compared the predictive performance, the number of parameters (Params), floating point operations per second (FLOPs), and training time of all deep learning models on the Tic-tac-toe dataset, as shown in Fig. 5, where the area of circles represents Params, other metrics are shown below it. ADNS demonstrates higher classification accuracy compared to other models with similar or larger parameter sizes. Additionally, although ADNS exhibits higher FLOPs

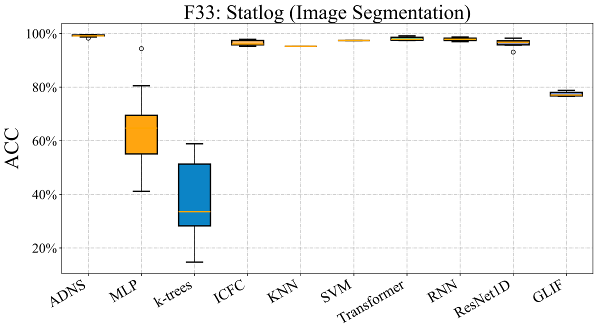


Fig. 7: The box-and-whisker plot of all models on Statlog dataset.

compared to other models, it boasts the fastest training speed. This may be attributed to the substantial number of dendritic layers within ADNS, resulting in increased parameter and computational load. However, these dendritic layers can be computed in parallel, thereby enabling rapid training of the model. Therefore, ADNS has the least training time among these models.

In Fig. 6, we compare the performance of different deep learning models on the Climate Model Simulation Crashes dataset as increasing the model parameter size. It is noteworthy that ADNS exhibits higher accuracy with relatively fewer parameters compared to other models. Conversely, other deep learning models reach a state where increasing the parameter size no longer improves accuracy. This highlights the effectiveness of ADNS in learning representations efficiently, allowing it to make better use of limited parameter space and achieve significant performance advantages in complex tasks.

We draw a box-and-whisker plot to verify the robustness of ADNS on Statlog (Image Segmentation) dataset, as shown in Fig. 7. The plot exhibits the results distribution of different independent experiments. From this figure, it is observed that ADNS has stable results with high-quality, which demonstrates the superior generalization and robustness of ADNS. The plots of other datasets are shown in Supplementary File.

C. Analysis of Hyperparameters and AEI Structure

1) Analysis of hidden layer size: In the parameter discussion section, we initiate an analysis of the hidden layer dimensions for all models. We evaluate the cases of hidden layer sizes set to 16, 32, and 64 across 47 feature-based classification datasets individually. The average across these 47 datasets is then computed and tabulated in Table III. The optimal results for each model are highlighted in bold. Moreover, within the Transformer architecture, the number of attention heads is set to 8 to ensure compatibility with the experimental configuration. Based on the experimental results in Table III, we adopted the best hidden size of MLP, ICFC, k-trees, Transformer, RNN, and ResNet1D to 16, 64, 32, 64, 64, and 16, respectively.

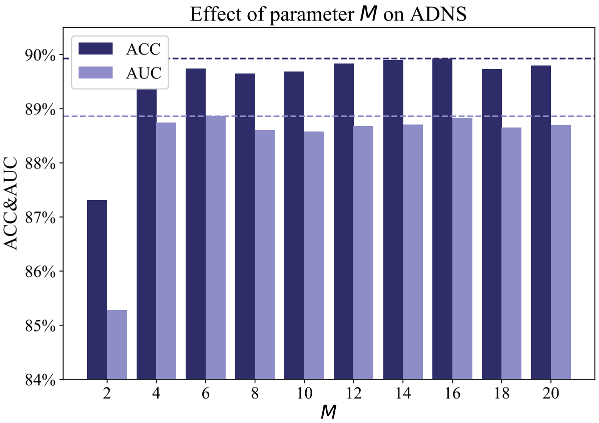


Fig. 8: The average ACC and AUC of ADNS with different M in 47 feature-based classification datasets.

TABLE III: Analysis of hidden layer sizes for all models.

	ACC			AUC		
	16	32	64	16	32	64
ADNS	89.66%	**89.93%**	89.77%	88.54%	**88.83%**	88.73%
MLP	**67.49%**	66.34%	65.95%	**69.10%**	68.13%	67.41%
k-trees	73.89%	**74.98%**	74.69%	74.79%	**75.85%**	75.17%
ICFC	86.80%	87.24%	**87.67%**	85.35%	86.29%	**86.85%**
Transformer	88.64%	88.68%	**88.96%**	88.14%	88.11%	**88.54%**
RNN	85.33%	86.18%	**86.29%**	84.53%	85.41%	**85.87%**
ResNet1D	**85.04%**	85.01%	84.23%	**85.01%**	84.96%	84.56%

2) Analysis of the ADNS hyperparameter M: The parameters of ADNS are analyzed to select the best setup, where M is a pivotal parameter. The impact of M value is discussed, in which the experiment with M values of $[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]$ is executed on 47 feature-based classification datasets. Similarly, each experiment is repeated 10 times to reduce the random error. The average ACC and average AUC of the 47 datasets are plotted in Fig. 8. Its results show that as the value of M increases from 2 to 6, the accuracy gradually improves, further proving that the number of dendrites has an impact on the computational capability of the model. Between M values of 6 to 20, the accuracy stabilizes and does not improve further, indicating that the computational power of ADNS with dendrite numbers between 6 to 20 is sufficient to solve the classification problems in the 47 datasets.

Furthermore, the highest average ACC is achieved at $M = 16$, while the highest average AUC is achieved at $M = 6$. The sum of average accuracy and average AUC is also the maximum at $M = 16$, out of all the M values. Therefore, M is set to 16 as the parameter for the comparative experiments with other models.

3) Analysis of parameters k and θ in ADNS: In Section IV, we highlighted our approach of introducing learnable parameters k and θ within the soma layer of the ADNS,

TABLE IV: Experimental discussion of k and θ for ADNS.

	ACC	AUC
Trainable	**89.93%**	**88.83%**
k=1, θ=0	89.28%	88.08%
k=5, θ=0.1	89.45%	88.61%
k=10, θ=0.1	89.51%	88.51%
k=15, θ=0.3	89.24%	88.40%

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3416236

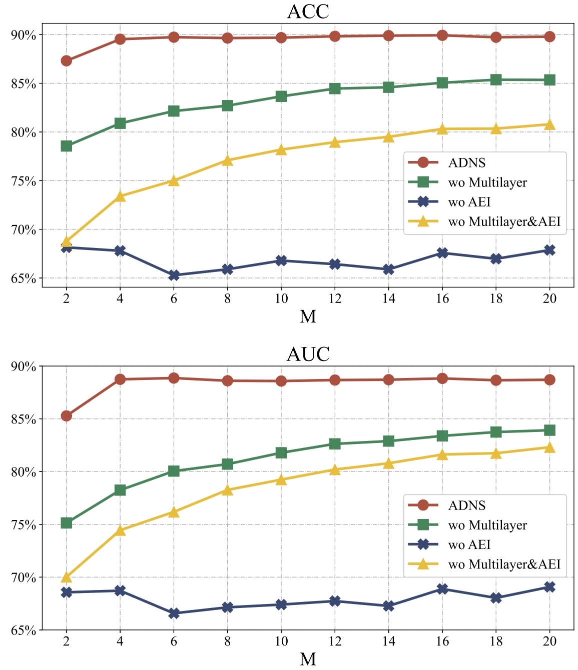


Fig. 9: Average ACC and average AUC results obtained in 47 datasets using different modules.

TABLE V: Experimental discussion of AEI and S-AEI.

	ACC	AUC
AEI	**89.93%**	**88.83%**
S-AEI	89.89%	88.74%

aiming to enhance adaptability across diverse datasets. In this section, we investigate four scenarios, each representing commonly adopted settings for k and θ in traditional DNM. By juxtaposing these scenarios with our devised adaptive strategy, we present experimental results across 47 datasets in Table IV. It is evident that configuring k and θ as learnable parameters yields improved performance in terms of both ACC and AUC evaluation metrics. This observation underscores the efficacy of our adaptive approach in capturing intricate dataset variations.

4) Analysis of AEI Structure: In this section, we further discussed the choice of activation function for the AEI mechanism. The AEI mechanism maps synaptic outputs to a range between -1 and 1 to simulate the excitatory and inhibitory actions of synapses in biological neural models. Hence, the sigmoid-like function is capable of simulating this process. tanh can directly map trainable parameters into a interval $(-1, 1)$, which is the AEI mechanism. Besides, a modified sigmoid function also can remap the trainable parameters into the interval via translation, called S-AEI mechanism, it is described as follows:

$$A_i = 2(\text{sigmoid}(m_i) - 0.5) \tag{25}$$

We examined both of these approaches and conducted experiments on 47 datasets, as summarized in Table V. From the table, it is evident that the AEI outperforms S-AEI. We attributed this superiority to the steeper slope of the tanh activation function compared to sigmoid. This steeper slope enables better modulation of excitatory and inhibitory effects on synapses. As a result, we opted to incorporate the tanh activation function in our model.

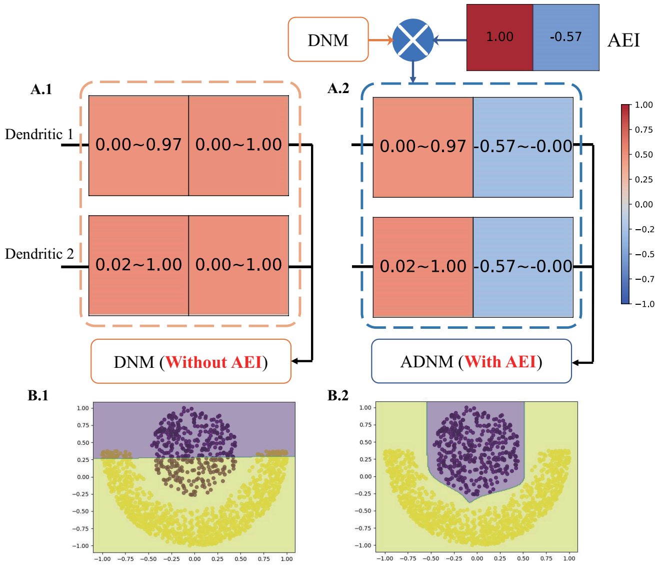


Fig. 10: The impact of AEI on the Moons dataset. A shows the synaptic output range. B shows the classification results of DNM and ADNM.

D. Ablation Study

In order to evaluate the importance of different modules in ADNS, ablation experiments are conducted. Meanwhile, the impact of different numbers of dendrites on each module is also investigated. For visually exhibiting the experimental results on 47 classification datasets, a line chart is drawn to show average results in Fig. 9.

The results depicted in Fig. 9 lead us to three conclusions: 1) Incorporating the AEI mechanism into DNM effectively enhances performance. 2) ADNM with the AEI mechanism demonstrates improved performance by increasing the number of layers. 3) DNM without the AEI mechanism exhibits significantly decreased performance when increasing the number of layers, indicating the inability of traditional DNM to form multi-layered neural systems.

Additionally, in Fig. 10, we conducted a straightforward experiment using the Moons dataset to observe the impact of AEI on classification outcomes. We employed DNM and ADNM with dendritic layer size $M = 2$. After training, we depicted the synaptic layer outputs of both models in Fig. 10 (A.1 and A.2). Subsequently, we generated decision boundary plots for both models to visually represent the classification results in Fig. 10 (B.1 and B.2). The decision boundary plots clearly illustrate that the inclusion of the AEI mechanism significantly enhances the model's classification performance. On the other hand, incorporating the AEI mechanism effectively enhances the versatility of DNM, enabling it to form more complex systems. These findings highlight the crucial role of biologically-inspired mechanisms, such as the AEI mechanism, in improving the performance and increasing the versatility of artificial neural networks.

VI. Conclusion

A typical dendritic neural network consists of multiple dendritic trees, where each tree is composed of dendritic segments and synapses. The dendrite is connected by synapses, which can either be excitatory or inhibitory. In this paper, a novel neural system ADNS that incorporates the AEI mechanism and the multilayer mechanism is proposed to improve the performance and versatility of the traditional DNM. The experimental results show that using the AEI mechanism can significantly enhance the accuracy of the model even without the multilayer mechanism. Moreover, adding the multilayer mechanism to DNM using the AEI mechanism can further improve the model performance. Experiments conducted on 47 feature-based classification datasets and 2 image-based classification datasets demonstrate that the proposed ADNS has significant advantages over other machine learning methods with deep learning models.

In summary, this study underscores the significance of integrating biologically-inspired mechanisms, such as AEI mechanism, into DNM to enhance its performance and expand its applicability. The proposed model not only outperforms the conventional single DNM model but also presents a novel approach for constructing multilayer neural systems. However, it's important to note that the current applicability of the ADNS is limited to feature classification. Its performance has not been adequately validated on other types of datasets like videos, audio, text, etc. In the future, we can further leverage the potential of the ADNS for in-depth investigations and maximize its feature classification capabilities. For instance, the incorporation of ADNS into popular network architectures like ResNet or Transformer can be explored, utilizing these networks as feature extractors for images or text, while employing ADNS as the feature classifier. This approach can tackle more complex real-world problems, thus presenting an exciting avenue for future research.

References

[1] L. Luo, "Architectures of neuronal circuits," *Science*, vol. 373, no. 6559, p. eabg7285, 2021.

[2] J. S. Isaacson and M. Scanziani, "How inhibition shapes cortical activity," *Neuron*, vol. 72, no. 2, pp. 231–243, 2011.

[3] B. Haider, M. Häusser, and M. Carandini, "Inhibition dominates sensory responses in the awake cortex," *Nature*, vol. 493, no. 7430, pp. 97–100, 2013.

[4] P. A. Fortier and C. Bray, "Influence of asymmetric attenuation of single and paired dendritic inputs on summation of synaptic potentials and initiation of action potentials," *Neuroscience*, vol. 236, pp. 195–209, 2013.

[5] M. W. Spratling and M. Johnson, "Exploring the functional significance of dendritic inhibition in cortical pyramidal cells," *Neurocomputing*, vol. 52, pp. 389–395, 2003.

[6] X. Wu, X. Liu, W. Li, and Q. Wu, "Improved expressivity through dendritic neural networks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[7] L. Bar-Ilan, A. Gidon, and I. Segev, "The role of dendritic inhibition in shaping the plasticity of excitatory synapses," *Frontiers in neural circuits*, vol. 6, p. 118, 2013.

[8] S. Yang, J. Tan, and B. Chen, "Robust spike-based continual meta-learning improved by restricted minimum error entropy criterion," *Entropy*, vol. 24, no. 4, p. 455, 2022.

[9] S. Yang, B. Linares-Barranco, and B. Chen, "Heterogeneous ensemble-based spike-driven few-shot online learning," *Frontiers in Neuroscience*, vol. 16, p. 850932, 2022.

[10] D. Zhao, Y. Zeng, and Y. Li, "BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons," *Neural Networks*, vol. 154, pp. 68–77, 2022.

[11] R. Urbanczik and W. Senn, "Learning by the dendritic prediction of somatic spiking," *Neuron*, vol. 81, no. 3, pp. 521–528, 2014.

[12] A. Vandesompele, F. Wyffels, and J. Dambre, "Dendritic computation in a point neuron model," in *Artificial Neural Networks and Machine Learning – ICANN 2020*, Springer. Springer International Publishing, 2020, pp. 599–609.

[13] D. V. Vargas and J. Murata, "Spectrum-diverse neuroevolution with unified neural models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1759–1773, 2017.

[14] J. Luo, L. Yu, T. Liu, M. Yang, Z. Fu, Z. Liang, L. Chen, C. Chen, S. Liu, S. Wu, Q. Huang, and R. Huang, "Capacitor-less stochastic Leaky-FeFET neuron of both excitatory and inhibitory connections for SNN with reduced hardware cost," in *2019 IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 6.4.1–6.4.4.

[15] J. Li, Z. Liu, R.-L. Wang, and S. Gao, "Dendritic deep residual learning for COVID-19 prediction," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 18, no. 2, pp. 297–299, 2023.

[16] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu *et al.*, "R-drop: Regularized dropout for neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 890–10 905, 2021.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] Z. Jiang, Y. Wang, C.-T. Li, P. Angelov, and R. Jiang, "Delve into neural activations: Towards understanding dying neurons," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 4, pp. 959–971, 2023.

[19] Z. Xu, Z. Wang, J. Li, T. Jin, X. Meng, and S. Gao, "Dendritic neuron model trained by information feedback-enhanced differential evolution algorithm for classification," *Knowledge-Based Systems*, vol. 233, p. 107536, 2021.

[20] R.-L. Wang, Z. Lei, Z. Zhang, and S. Gao, "Dendritic convolutional neural network," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 17, no. 2, pp. 302–304, 2022.

[21] M. Robnik-Šikonja, "Data generators for learning systems based on RBF networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 926–938, 2016.

[22] P. Poirazi and A. Papoutsi, "Illuminating dendritic function with computational models," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 303–321, 2020.

[23] H. Peng, T. Bao, X. Luo, J. Wang, X. Song, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "Dendrite P systems," *Neural Networks*, vol. 127, pp. 110–120, 2020.

[24] X. Wen, M. Zhou, X. Luo, L. Huang, and Z. Wang, "Novel pruning of dendritic neuron models for improved system implementation and performance," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 1559–1564.

[25] X. Li, J. Tang, Q. Zhang, B. Gao, J. J. Yang, S. Song, W. Wu, W. Zhang, P. Yao, N. Deng *et al.*, "Power-efficient neural network with artificial dendrites," *Nature Nanotechnology*, vol. 15, no. 9, pp. 776–782, 2020.

[26] I. S. Jones and K. P. Kording, "Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree?" *Neural Computation*, vol. 33, no. 6, pp. 1554–1571, 2021.

[27] F. Arce, E. Zamora, H. Sossa, and R. Barrón, "Differential evolution training algorithm for dendrite morphological neural networks," *Applied Soft Computing*, vol. 68, pp. 303–313, 2018.

[28] J. Sacramento, R. Ponte Costa, Y. Bengio, and W. Senn, "Dendritic cortical microcircuits approximate the backpropagation algorithm," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[29] Y. Todo, H. Tamura, K. Yamashita, and Z. Tang, "Unsupervised learnable neuron model with nonlinear interaction on dendrites," *Neural Networks*, vol. 60, pp. 96–103, 2014.

[30] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[31] X. Luo, X. Wen, M. Zhou, A. Abusorrah, and L. Huang, "Decision-tree-initialized dendritic neuron model for fast and accurate data classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4173–4183, 2022.

[32] Z. Liu, Z. Zhang, Z. Lei, M. Omura, R.-L. Wang, and S. Gao, "Dendritic deep learning for medical segmentation," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 3, pp. 803–805, 2024.

[33] Z. Zhang, Z. Lei, M. Omura, H. Hasegawa, and S. Gao, "Dendritic learning-incorporated vision transformer for image recognition," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 539–541, 2024.

[34] S. Gao, M. Zhou, Z. Wang, D. Sugiyama, J. Cheng, J. Wang, and Y. Todo, "Fully complex-valued dendritic neuron model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 2105–2118, 2023.

[35] A. Gupta, H. P. Gupta, B. Biswas, and T. Dutta, "Approaches and applications of early classification of time series: A review," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 1, pp. 47–61, 2020.

[36] T. De Ryck, S. Lanthaler, and S. Mishra, "On the approximation of functions by tanh neural networks," *Neural Networks*, vol. 143, pp. 732–750, 2021.

[37] Q. Peng, S. Gao, Y. Wang, J. Yi, G. Yang, Y. Todo *et al.*, "An extension network of dendritic neurons," *Computational Intelligence and Neuroscience*, vol. 2023, p. 7037124, 2023.

[38] J. An, F. Liu, F. Shen, J. Zhao, R. Li, and K. Gao, "IC neuron: An efficient unit to construct neural networks," *Neural Networks*, vol. 145, pp. 177–188, 2022.

[39] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.

[40] X. Wu, W. Zuo, L. Lin, W. Jia, and D. Zhang, "F-SVM: Combination of feature transformation and svm learning via convex relaxation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5185–5199, 2018.

[41] S. Hong, Y. Xu, A. Khare, S. Priambada, K. Maher, A. Aljiffry, J. Sun, and A. Tumanov, "HOLMES: Health online model ensemble serving for deep learning models in intensive care units," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1614–1624.

[42] X. Yao, F. Li, Z. Mo, and J. Cheng, "GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 160–32 171, 2022.

Jiayi Li received his M.E. degree from University of Toyama, Toyama, Japan in 2023, where he is currently pursuing his Ph.D. degree. His current research interests include machine learning, evolutionary computation, neural networks, and bioinformatics.

Zhenyu Lei received the Ph.D. degree in Science and Engineering from the University of Toyama, Toyama, Japan, in 2023. He is currently an Assistant Professor with the Faculty of Engineering, University of Toyama, Japan. His current research interests include evolutionary computation, machine learning, and neural network for real-world applications and optimization problems.

Zhiming Zhang received his M.E. degree from University of Toyama, Toyama, Japan in 2022, where he is currently pursuing his Ph.D. degree. His current research interests include machine learning, neural networks, and bioinformatics.

Haotian Li received his M.E. degree in Intellectual Information Engineering from University of Toyama, Japan in 2023. He is currently pursuing the Ph.D. degree at University of Toyama, Toyama, Japan. His current research interests are computational intelligence.

Yuki Todo Yuki Todo received the D.E. degree from Kanazawa University, Kanazawa, Japan, in 2005. From 1987 to 1989, she was an Assistant Professor with the Institute of Microelectronics, Shanghai Jiaotong University, Shanghai, China. In 2012, she joined Kanazawa University, where she is currently an Associate Professor with the Faculty of Electrical, Information and Communication Engineering. Her current research interests include brain-like computing, neural networks, and optimization.

Shangce Gao (Senior Member, IEEE) received his Ph.D. degree in Innovative Life Science from University of Toyama, Toyama, Japan in 2011. He is currently a Professor with the Faculty of Engineering, University of Toyama, Japan. His current research interests include nature-inspired technologies, machine learning, and neural networks for real-world applications. He serves as an Associate Editor for many international journals such as IEEE Transactions on Neural Networks and Learning Systems, and IEEE/CAA Journal of Automatica Sinica.