Artificial Intelligence Review (2025) 58:129
https://doi.org/10.1007/s10462-025-11123-y

Dmixnet: a dendritic multi-layered perceptron architecture for image recognition

Weixiang Xu[1] · Yaotong Song[1] · Shubham Gupta[2] · Dongbao Jia[3] · Jun Tang[4] · Zhenyu Lei[1] · Shangce Gao[1]

Accepted: 23 January 2025 / Published online: 13 February 2025

Abstract
In the field of image recognition, the all-MLP architecture (MLP-Mixer) shows superior performance. However, the current MLP-Mixer is solely based on fully connected layers. The nonlinear capability of fully connected layers is relatively weak, and their simple stacked structure has limitations under complex conditions. Therefore, inspired by the diversity of neurons in the human brain, we propose an innovative DMixNet, a dendritic multi-layered perceptron architecture. Rooted in the theory of dendritic neurons from neuroscience, we propose a dendritic neural unit (DNU) that enhances DMixNet with stronger biological interpretability and more robust nonlinear processing capabilities. The flexibility of dendritic structures allows the DNU to adjust its architecture to achieve different functionalities. Based on the DNU, we propose a novel channel fusion network $\mathrm{DNU_E}$ and a dendritic classifier $\mathrm{DNU_C}$. The $\mathrm{DNU_E}$ substitutes the traditional two fully connected layers as the channel mixer, constructing a dendritic mixer layer to enhance the fusion capability of channel information within the entire framework. Meanwhile, the $\mathrm{DNU_C}$ replaces the traditional linear classifier, effectively improving the model's classification performance. Experimental results demonstrate that DMixNet achieves improvements of 2.13%, 4.79%, 4.71%, 23.14% on the CIFAR-10, CIFAR-100, Tiny-ImageNet and COIL-100 benchmark image recognition datasets, respectively, as well as a 14.78% enhancement on the medical image classification dataset PathMNIST, outperforming other state-of-the-art architectures. Code is available at https://github.com/KarilynXu/DMixNet.

Keywords MLP-mixer · Dendritic neural unit · Dendritic channel module · Dendritic classifier · Image recognition

1 Introduction

MLP-Mixer is a novel visual framework proposed by Google, achieving competitive results solely through fully connected layer (Tolstikhin et al. 2021). The emergence of MLP-Mixer challenges the mainstream status of vision transformer (ViT) (Dosovitskiy et al. 2021) and convolutional neural network (CNN) frameworks (He et al. 2016) in computer vision. Its

Extended author information available on the last page of the article presence indicates that convolution or attention mechanisms are unnecessary for superior model performance (Yu et al. 2022). The MLP-Mixer framework is simple, and it is more efficient than the ViTs computation as they do not require matrix multiplications for key-query operations (Benz et al. 2021). Solely relying on the fully connected layer, it outperforms state-of-the-art visual backbone networks (Wei et al. 2023).

The MLP-Mixer framework achieves spatial information fusion through the collaborative effort of token-mixer and channel-mixer mechanisms, where token-mixer integrates information within patches and channel-mixer facilitates communication across channels. The token-mixer and channel-mixer in the original MLP-Mixer architecture are both composed of two fully connected layers. Most variations of the MLP-Mixer focus on improving the information fusion mechanism of the token-mixer. These token MLP-Mixer variants spatially recode the information before it enters the two fully connected layers of the token-mixer, enhance the spatial mixing of the data in a way similar to data preprocessing (such as spatial-shift and axial-shift), thus improving the performance of the whole model (Yu et al. 2022; Hou et al. 2022). This phenomenon indicates that the nonlinear processing capability of simple stacked two-layer fully connected layers is limited. Consequently, by increasing the degree of spatial information mixing before the two fully connected layers of the token-mixer, these token MLP-Mixer variants can overcome inherent limitations of fully connected layers and achieve better performance (Liu et al. 2022). As to channel-mixer, they retain the initial two fully-connected layers to process the data.

Another category of variants focuses on improving the entire MLP-Mixer framework (Touvron et al. 2022; Liu et al. 2021). They still use fully connected layers for both the token-mixer and channel-mixer but enhance the framework's spatial information fusion capability, resulting in better performance. Whether it is the token MLP-Mixer variants or the framework MLP-Mixer variants, both improve the model's ability to fuse spatial information in different ways. This indicates that fully connected layers have limitations in their processing capabilities. Thus, it raises the question: can fully connected layers satisfy the diverse functional requirements of different modules? For the MLP-Mixer architecture, is it optimal to implement all modules exclusively using fully connected layers?

Modern artificial intelligence primarily relies on multilayer perceptrons, whose basic units mimic the simple linear model of biological neurons (Fan et al. 2023). This concept can be traced back to 1943 when McCulloch and Pitts abstracted the biological nervous system into a neural network (McCulloch and Pitts 1943). Later, Rosenblatt expanded this idea and proposed the perceptron (Rosenblatt 1958). However, the perceptron could not solve the basic XOR problem. It was not until the introduction of the backpropagation algorithm (Rumelhart et al. 1986) that research on neural networks rapidly advanced. It is well known that the human brain contains neurons with diverse morphologies and functions, while artificial neural networks are almost exclusively built on a single type of neuron (Bassett and Gazzaniga 2011). The brain is the most intelligent system known to date, and the development of artificial neural networks has continuously been inspired and driven by our understanding of the brain (LeCun et al. 2015). In the human brain, neuronal diversity is a key factor in achieving various intelligent biological behaviors. Since artificial neural networks are a miniature model of the human brain, introducing neuronal diversity is of great significance in addressing the fundamental issues of artificial neural networks (Telley and Jabaudon 2018).

In recent years, researchers have proposed new types of neurons with diversity, such as spiking neural networks, which perform well in fields like pattern recognition and real-time signal processing, and dendritic neural networks, which have achieved excellent results in classification and prediction tasks. The MLP-Mixer framework is based on a simple fully connected layer structure, which has limitations in nonlinear processing capabilities (Goodfellow et al. 2016). Therefore, inspired by neuronal diversity, this work proposes a novel dendritic neural network to address the limitations of the MLP-Mixer framework composed solely of fully connected layers from the perspective of basic neurons firstly. The contributions of this paper are:

- We propose a dendritic neural unit (DNU), which considers the crucial role of synapses in information transmission within biological cells and integrate with the information-processing processes of biological neurons. The dendritic neural unit has enhanced nonlinear processing capability and greater flexibility. Its structure can be adjusted according to different functionalities to generate diverse dendritic neural modules tailored to different tasks.
- Inspired by biological architecture, we construct a dendritic multi-layered perceptron architecture. It flexibly employs DNU, capitalizing its advantages to enhance the framework's channel feature fusion capability and improve feature classification performance.
- We design a novel dendritic channel module $\mathrm{DNU_E}$, which is a feature extraction module built on DNU. Harnessing the structural characteristics of dendrites, $\mathrm{DNU_E}$ enhances channel information capacity. Compared to the traditional two fully connected layers, the $\mathrm{DNU_E}$ has fewer parameters, higher accuracy, and is more concise and efficient.
- We design a dendritic classifier $\mathrm{DNU_C}$, which is a feature classification module built on DNU. The $\mathrm{DNU_C}$ exhibits strong nonlinear processing capabilities, improving the classification performance of the model.

2 Related work

2.1 MLP-mixer

The MLP-Mixer consists of the token-mixer and channel-mixer, which jointly form the mixer layer (Tolstikhin et al. 2021). Both the token-mixer and the channel-mixer are composed of two fully connected layers. If the resolution of the original image is (*H*, *W*) and the resolution of each patch is *P*, the number of patches $S = HW/P^2$. The input of mixer layer $X \in \mathbb{R}^{S\times D}$ is obtained after the processing of the linear project. The channel-mixer performs information fusion between different channels, maps $\mathbb{R}^S \rightarrow \mathbb{R}^S$. The token-mixer performs spatial information fusion within each patch, maps $\mathbb{R}^D \rightarrow \mathbb{R}^D$. The whole process of the mixer layer is expressed as:

$$U_i = X_i + W_2\delta(W_1 LayerNorm(X_i)), \quad \text{for} \quad i = 1, ..., S \tag{1}$$

$$H_j = U_j + W_4\delta(W_3 LayerNorm(U_j)), \quad \text{for} \quad j = 1, ..., D \tag{2}$$

where δ is the activation function, W_1, W_2, W_3, and W_4 are connected weights of the fully connected layer, D is the sequence length of each patch.

In later studies, some researchers propose improvements to the MLP-Mixer framework (Liu et al. 2022). For example, Touvron et al. propose ResMLP by incorporating a residual module into the MLP-Mixer framework, which consists of a single hidden layer feedforward network and a linear patch interaction layer forming the residual module (Touvron et al. 2022). Liu et al. propose the gMLP framework, which combines the spatial gating unit and MLP-Mixer (Liu et al. 2021). Another research direction involves altering the token-mixer mechanism of the MLP-Mixer (Zhang et al. 2022). Sparse MLP replaces token-mixer by utilizing depthwise convolution and Sparse-MLP module (Tang et al. 2022). Lian et al. propose axial shifted MLP (AS-MLP), which focuses more on the interaction of local features. By shifting the channel axis of feature maps, AS-MLP can obtain information from different axes, allowing the network to capture local dependencies (Lian et al. 2021). Such token MLP-Mixer variants directly still keep the channel-mixer of the original MLP-Mixer. Whether based on framework-based improvements or fusion strategy-based enhancements, they employ network modules with stacked fully connected layers to achieve different functionalities. The nonlinear processing capability of stacked fully connected layers is limited and insufficient to adapt to variations in different functionalities. Inspired by biological neurons, we propose a dendritic neuron unit to overcome the limitations of fully connected layers in constructing diverse functional modules within the MLP-Mixer framework.

2.2 Dendritic Learning

Inspired by the neural cognition of the human brain, contemporary artificial neural networks are based on multi-layer perceptron, and their basic unit is a simple simulation of the biological cell. It is well-known that neurons in the human brain exhibit diversity, whereas artificial neural networks are primarily built upon a single type of neuron (Poirazi and Papoutsi 2020). Hence, the introduction of diverse neurons, coupled with extensive knowledge from neuroscience, imparts more excellent capabilities to networks in addressing the challenges faced by artificial neural networks (such as interpretability, efficiency, memory, etc.), catalyzing the next generation of artificial intelligence, which remains a hotpot in artificial intelligence research (Zador et al. 2023).

Biological studies have shown that dendrites play a crucial role in neurons in receiving, integrating, and transmitting signals, and they are essential components of neuronal function and are vital for the normal operation and information processing of neuronal networks (Stuart et al. 2016). Inspired by the synaptic signal transmission mechanism, Gao et al. proposed dendritic neuron model, which utilize comprehensive experimental validation of dendritic neural model to contrast the advantages over traditional multi-layer perceptron (Gao et al. 2018). The structural advantages of dendrites enable related models to achieve outstanding performance in many fields. The later research explored effective improvements in training dendritic neuron models using evolutionary algorithms, achieving excellent performance in the application of dendritic neuron models to classification problems (Egrioglu et al. 2023). Dendritic neuron models have achieved significant success in the fields of classification and prediction (Egrioglu and Bas 2024). Further, advanced researchers proposed the fully complex-valued dendritic neuron model (Lee et al. 2022), which extends the dendritic neural model from the real-value domain to the complex-value domain, demonstrating the immense potential inherent in dendritic learning. In the field of deep learning, dendritic deep learning has attracted the attention of many scholars. Zhang et al. combined dendritic learning with vision transformer, effectively improving the classification performance of the ViT (Zhang et al. 2024). Liu et al. proposed dendritic deep learning for medical segmentation (Liu et al. 2024). In this paper, we propose DMixNet, leveraging the advantages of dendritic learning to enhance the performance of MLP-Mixer framework.

3 Methodology

As illustrated in Fig. 1, we initially constructed the dendritic neural unit by emulating the patterns of biological neurons, ranging from selective to space-filling dendritic structures. Biological cells exhibit diverse dendritic arbor densities according to their functions. Intermediate arbor densities, termed sampling arborizations, possess complex dendritic patterns that allow neurons to gather a wide range of inputs while retaining specificity, as observed in cerebral cortex pyramidal cells (Shipp 2007). It can evolve in two directions. One direction lie selective arborizations. For example, in olfactory sensory cells (Elsaesser and Paysan 2007), the dendritic structure is sparse, with each dendrite connecting the cell body directly to a specific target. Another direction are space-filling arborizations. In extreme cases, like the cerebellar Purkinje cells, the dendrites extensively branch out to cover a large area, maximizing the input region and integrating a wide array of signals (Konnerth et al. 1990). The dendritic neural unit learns from the patterns of biological neurons, ranging from selective to space-filling dendritic structures, to simulate the morphological variations adopted for different functions.

Subsequently, different DNU submodules are developed based on their functional requirements, culminating in the DMixNet framework. The structure of DMixNet is shown in Fig. 2. Firstly, the model reshapes image patches as “Patches×Channels" using linear projection, which serves as the input for the dendritic mixer layer. The dendritic mixer layer inputs the signal into the dendritic channel module $\mathrm{DNU_E}$ for channel-mixer. Then, it processes the transposition signals obtained from the $\mathrm{DNU_E}$ through the token-mixer, which

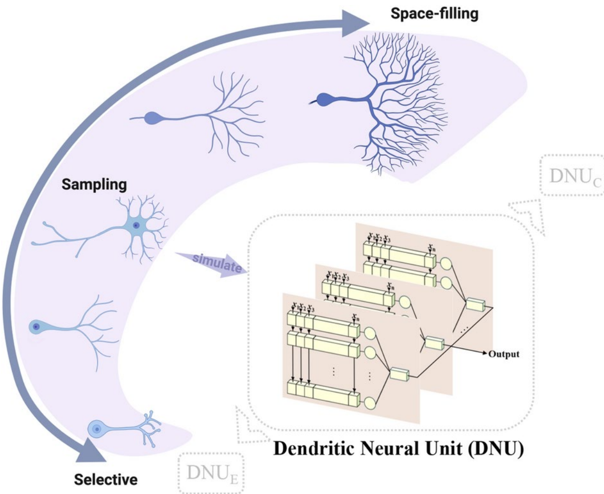

Fig. 1 The specificity of dendritic neural unit. The dendritic neural unit possesses a highly flexible structure, capable of simulating the evolutionary process by which biological cells develop different structures based on their functions. Depending on the functional requirements of each module, the dendritic neural unit can evolve along two pathways: selective and space-filling, dynamically adjusting its morphology and functionality. This adaptability enables the construction of dendritic neural modules with diverse functionalities

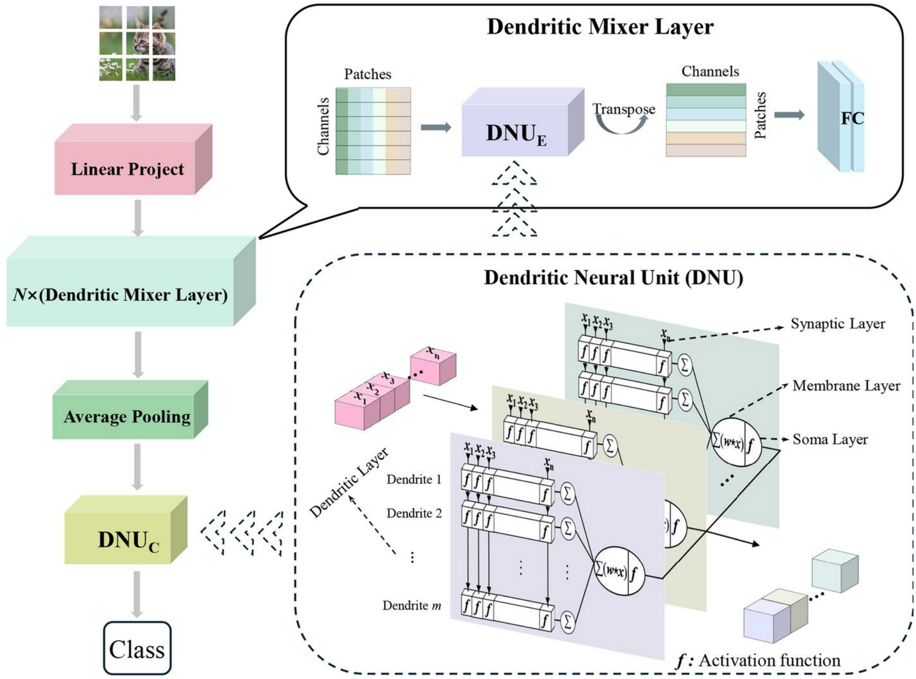

Fig. 2 The structure of DMixNet. The dendritic neural unit dynamically adjusts its morphology based on the functional requirements of each module. Specifically, the $\mathrm{DNU_E}$ module is designed for feature extraction, while the $\mathrm{DNU_C}$ module is tailored for feature classification

consists of two fully connected layers. The result obtained after multiple iterations of the dendritic mixer layer is fed into the dendritic classifier $\mathrm{DNU_C}$. Leveraging the structural advantages of dendrites, the $\mathrm{DNU_C}$ maps complex input feature combinations to derive the final classification results. The following sections will provide a detailed overview of the DMixNet framework.

3.1 Dendritic neural unit

Neural signal processing in biological neurons is a highly complex system where the joint action of synapses and dendrites plays a crucial role in information transmission and processing in the nervous system. The basic unit of artificial neural networks is a simplified simulation of biological cells, which overlooks the crucial role of dendrites in biological cell information transmission. The neuron doctrine suggests that as the intellectual development of organisms advances, dendritic arbours become more complex and extensive (Shepherd 2015). Due to the limited surface area of spherical cell bodies used for receiving inputs, dendrites effectively increase their surface area, thereby enhancing the signal processing capability of neurons without excessively increasing the volume of the cell body. Hence, based on the biological dendritic theory, we incorporate simulations of biological neurons and construct dendritic neural unit. The dendritic neural unit is divided into four layers, which are:

- Synaptic layer: Initially, the dendritic neural unit mathematically models the processing of synapses through threshold linear operations. Signals are inputted from its synaptic layer, where each synapse normalizes and linearly activates the input signals. This process can be represented as:

$$S_{id}^{t} = w_{id}^{t} LayerNorm(x_{id}^{t}) + b_{id}^{t} \tag{3}$$

where *t* is the number of dendritic neurons. x_{id}^{t} denotes the input of the *i*th synapse on the *d*th dendritic branch. w_{id}^{t} denotes the synaptic connectivity weight, while b_{id}^{t} represents the threshold. S_{id}^{t} represents the output of the *i*th synapse of the *d*th dendritic branch of the *t*th dendritic neuron.

- Dendritic layer: Subsequently, the dendritic layer processes signals from the synapses. Dendrites receive input signals from other neurons and undergo a certain degree of signal processing and integration between dendrites and the cell body, which is known as dendritic local computation (Koch and Poggio 1985). This local computation can influence whether the cell body generates neural impulses, as well as the frequency and intensity of the impulses produced. This type of local computation enables neurons to exhibit complex responses to input signals, thereby achieving information processing and transmission in neural networks. In order to simulate the local computation of the dendrites of biological neurons, we incorporate activation functions in the dendrite layer. It applies a non-linear activation function to the signals, followed by normalization of the activated results and separate summation to obtain the output of each dendritic layer. The entire process can be represented as:

$$D_{d}^{t} = \sum_{i=1}^{n} LayerNorm(\phi(S_{id}^{t})) \tag{4}$$

where *n* represents the number of synapses on each dendritic layer, $\phi(\cdot)$ is activation function of dendritic layer. D_{d}^{t} is the output of the *d*th dendritic layer.

- Membrane layer: The membrane layer performs a weighted sum of the results from the dendritic layers, assigning different weights based on the importance of dendritic layer signals to obtain the output of the membrane layer. Its mathematical expression is:

$$M^{t} = \sum_{d=1}^{m} \alpha_{d} D_{d}^{t} \tag{5}$$

where *m* is the number of dendritic layer. α_{d} is the adaptive weight for each dendritic layer. M^{t} is the output of membrane layer.

- Soma layer: Finally, the signal arrives at the soma layer and passes through the activation function of the soma layer. The activation function in the soma layer simulates the processing of signals by the cell body of biological neurons. The output of the dendritic neural unit is:

$$y^t = \psi(M^t) \tag{6}$$

$$Y = [y^1, y^2, ..., y^t] \tag{7}$$

where $\psi(\cdot)$ is activation function of soma layer. y^t is the output of individual dendritic neuron, Y is the output of dendritic neural unit.

The above is details of the four-layer structure of the dendritic neural unit. As shown in Fig. 1, the DNU demonstrates distinct advantages. Firstly, it considers the role of dendrites in the construction of basic neurons, enhancing the signal processing capabilities of individual neurons. Secondly, the DNU structure is highly flexible, allowing for adjustments to the layer structure according to various functional requirements, and enabling the construction of different types of dendritic structures, ranging from selective to space-filling.

Based on the DNU, we propose the $\mathrm{DNU_E}$ module of dendritic mixer layer and the $\mathrm{DNU_C}$ module into the DMixNet. Specifically, the design of $\mathrm{DNU_E}$ focuses on reinforcing the central role of the soma in information processing. By simplifying the complexity of the dendritic structure, the $\mathrm{DNU_E}$ allocates more resources to improving soma functions, such as enhancing signal integration, amplification, and transmission efficiency, aiming to boost the computational capabilities and adaptability of the network in a broader neural network environment. To achieve this, the $\mathrm{DNU_E}$ adopts a four-layer structure, aiming to enhance the performance of individual neuron units by increasing processing levels, thereby providing a more robust foundation for the neuron's information integration capabilities.

Conversely, the $\mathrm{DNU_C}$ adopts a distinctly different design strategy, inclined towards simulating the characteristics of highly differentiated single-cell neurons. By introducing more complex and enriched dendritic structures to mimic the intricate branching networks of actual neurons, the $\mathrm{DNU_C}$ relatively reduces the direct role of the soma in information processing, thereby promoting the specificity and efficiency of signal processing. To achieve this, the $\mathrm{DNU_C}$ employs a three-layer architecture design, carefully balancing dendritic complexity with local computational capabilities of the dendrites, striving to achieve breakthroughs in feature classification. We provide a detailed introduction to the $\mathrm{DNU_E}$ module and the $\mathrm{DNU_C}$ module in the following sections.

3.2 Dendritic channel module $\mathrm{DNU_E}$

The function of channel-mixer is to fuse cross-channel information, while the characteristics of dendrites can enhance the feature extraction across different channels. We construct a novel channel fusion network $\mathrm{DNU_E}$ to replace the traditional two fully connected layers as the channel-mixer. The synaptic layer of the $\mathrm{DNU_E}$ module receives the input signal. Each synapse normalizes the input signal and linearly activates it. Then, each dendritic layer accumulates the results from synaptic outputs to obtain the output. The membrane layer assigns different weights to the signals from the dendritic layer based on their importance and computes the weighted sum to obtain the output. The signal reaches the soma layer, where the GELU activation function is applied to obtain the output of the $\mathrm{DNU_E}$ module. The mathematical expression of the $\mathrm{DNU_E}$ is:

$$y_E^l = \psi(\sum_{d=1}^{m} \alpha_d^l (\sum_{i=1}^{n} w_{id}^l LayerNorm(x_{id}^l) + b_{id}^l)) \tag{8}$$

$$Y_E = [y_E^1, y_E^2, ..., y_E^l] \tag{9}$$

where ψ represents the activation function of the $\mathrm{DNU_E}$, which is the GELU function. *l* is the number of dendritic neurons equal to the number of channels. Y_E is the output of the $\mathrm{DNU_E}$.

The task of $\mathrm{DNU_E}$ is feature extraction, which serves as the intermediate layer of the entire neural network. It employs single neurons with high complexity and low dendritic density, designed to be computed by the soma layer. Therefore, DNUE adopts a complete four-layer structure. The high complexity of individual neurons is advantageous for capturing complex features of input data, while the low dendritic density helps reduce the computational cost of the entire network.

Then, we embed the dendritic channel module $\mathrm{DNU_E}$ into the mixer layer to form the dendritic mixer layer. To facilitate the rapid fusion of complex channel information extracted by the $\mathrm{DNU_E}$, we employ multiplication for linear acceleration of the results obtained from the $\mathrm{DNU_E}$. The entire process can be represented by:

$$U_i = X_i + X_i Y_E \tag{10}$$

3.3 Dendritic classifier $\mathrm{DNU_C}$

The role of linear classifier is to combine the learned features of the network with weighted coefficients and bias processing to generate the final classification result. It typically consists of single fully connected layer, where the number of neurons equals the number of classes in the dataset, with each neuron corresponding to a class. The capability of the single layer linear classifier is limited, particularly when dealing with complex nonlinear data. The structural advantages of dendrites enable them to better map complex input feature combinations. Therefore, we propose a dendritic classifier $\mathrm{DNU_C}$ to improve the model's classification accuracy.

Initially, the signal enters the synaptic layer of the $\mathrm{DNU_C}$. The synapses normalize the features and apply linear activation. Then, the dendritic layer uses the sigmoid function to nonlinearly activate the signals from the synapses. Subsequently, the activated results are normalized again and separately summed up to obtain the output of each dendritic branch layer. Finally, the dendritic layers are directly connected to the soma layer, where the outputs of the dendritic layers are summed up to obtain the final classification result. The expression of the $\mathrm{DNU_C}$ is:

$$y_C^k = \sum_{d=1}^{m} \sum_{i=1}^{n} LayerNorm(\phi(w_{id}^k LayerNorm(x_{id}^k) + b_{id}^k) \tag{11}$$

$$Y_C = [y_C^1, y_C^2, ..., y_C^k] \tag{12}$$

where ϕ represents the activation function of the $\mathrm{DNU_C}$, which is the softmax function. k is the number of dendritic neurons equal to the number of categories for categorization. Y_C is the classification result of the $\mathrm{DNU_C}$.

The function of the $\mathrm{DNU_C}$ is feature classification, requiring more transformations and mappings in the input feature space to effectively perform classification tasks. It employs single neurons with low complexity and high dendritic density, designed for local computation by the dendritic layer. Therefore, $\mathrm{DNU_C}$ adopts a simplified three-layer structure. By maintaining the low complexity of individual dendritic neuron, it facilitates simpler feature transformation and classification operations; meanwhile, increasing the dendritic density helps better capture the complex structure and relationships in the input data, thereby improving classification accuracy and robustness.

3.4 Advantage of DMixNet

The $\mathrm{DNU_E}$ and the $\mathrm{DNU_C}$ share similar macroscopic structures but differ in the detailed design of micro-modules. In biological neurons, different types of neurons have varying dendritic morphologies and densities of dendritic branches. Additionally, the degree of local computation by dendrites in biological neurons depends on the cell type and function (Häusser and Mel 2003). Certain types of cells are more inclined to perform local computations, while others rely more on central processing by the soma. As shown in Fig. 1, the DNU module simulates the patterns of biological cells, considering not only the structural design and local computational functions of dendrites but also the morphological variations of dendrites. Therefore, the DNU exhibits high flexibility, allowing its structure to be adjusted according to the functions of different modules to adapt to various changes. This characteristic endows DMixNet with higher biological interpretability and stronger nonlinear processing capabilities.

For channel information fusion, DMixNet introduces $\mathrm{DNU_E}$ to enhance the communication and information fusion capabilities of the dendritic Mixer Layer. For feature classification, we propose $\mathrm{DNU_C}$ to improve the classification performance of DMixNet. The DMixNet effectively leverages the advantages of DNU by designing different sub-models for different functions.

4 Experiment

4.1 Datasets

The experiments utilize commonly used benchmark datasets in the field of deep learning, including CIFAR-10, CIFAR-100 (Krizhevsky et al. 2009), Tiny-ImageNet (Fei-Fei et al. 2009), and Columbia Object Image Library (COIL-100) Dataset (Nene et al. 1996). CIFAR-10 is a relatively small dataset containing 10 classes, suitable for quickly testing and validating the performance of image classification algorithms. In contrast, CIFAR-100 consists of 100 classes with a richer hierarchical structure, providing more challenging classification tasks. The COIL-100 dataset, which comprises 100 different objects captured from 72 different angles, serves as an excellent resource for evaluating the robustness of algorithms to variations in object orientation and viewpoint. Tiny-ImageNet serves as an intermediate dataset between CIFAR and the original ImageNet, comprising 200 classes and higher-resolution images, allowing for the evaluation of algorithm performance on larger-scale and more complex datasets. Through experiments on these four datasets, we can comprehensively assess the generalization ability and effectiveness of the proposed method. To further underscore the model's potential, we extend the evaluation to PathMNIST (Yang et al. 2023), a medical image classification dataset derived from histopathology images. This additional evaluation highlights the model's potential and advantages in handling complex, real-world challenges, particularly in the context of healthcare-related applications.

4.2 Training details

All models are trained using AdamW optimizer (Loshchilov and Hutter 2018) with a batch size of 64, and trained for 100 epochs. None of the models have been pretrained. The data preprocessing and augmentation techniques are consistent across all models for the three datasets. The experiments are conducted on Ubuntu 18.04, using Python 3.9 and PyTorch 2.0, with NVIDIA RTX3090 GPU configuration.

4.3 Performance comparison

The experiments evaluated the performance of DMixNet against other state-of-the-art architectures, including AS-MLP, ResMLP, VGG19 (Simonyan and Zisserman 2014), ConvNeXt V2 (Woo et al. 2023), ViT, and ConvFormer (Yu et al. 2024), on the CIFAR-10, CIFAR-100, Tiny-ImageNet datasets, COIL-100, and PathMNIST. ACC-k1 (Top-1 Accuracy) measures the percentage of times the model's most likely prediction matches the true class, while ACC-k3 (Top-3 Accuracy) measures the percentage of times the true class is within the model's top three predictions. The results in Table 1 show that DMixNet achieves the highest accuracy across the board, whether compared to the classic CNN architectures VGG19 and ConvNeXt V2, the transformer-based ViT and ConvFormer, or the MLP-Mixer-based architectures AS-MLP and ResMLP.

Compared to other models such as MLP-Mixer, classic CNN architectures like VGG19 and ConvNeXt V2, and Transformer-based architectures like ViT and ConvFormer, DMixNet exhibits a significant reduction on Params. Specifically, DMixNet's parameter is 6.3M, and its FLOPs are 404 M, while MLP-Mixer has a paramete of 6.9M and FLOPs of 505 M, which are far higher than those of DMixNet. This indicates that DMixNet is much more lightweight in terms of computational resource consumption. Moreover, DMixNet not only reduced Params and FLOPs but also significantly improved accuracy, with increases of 2.13%, 4.79%, 4.71%, and 23.14% on CIFAR-10, CIFAR-100, Tiny-ImageNet, and COIL-100, respectively. Notably, DMixNet shows significant improvements compared to MLP-Mixer variants such as AS-MLP and ResMLP, especially on challenging datasets like Tiny-ImageNet. This highlights its ability to achieve an excellent balance between computational efficiency and performance. When compared to classic CNN architectures like VGG19 and ConvNeXt V2, DMixNet demonstrated significant advantages. It achieved higher accuracy while maintaining a lower parameter. In comparisons with Transformer-based models like ViT and ConvFormer, DMixNet's Params and FLOPs are much lower, yet it still achieved better accuracy across multiple datasets. This advantage underscores the

Table 1 Comparison results of different models on CIFAR-10, CIFAR100, Tiny-ImageNet, COIL-100, and PathMNIST

Model	Params (M)	FLOPs (M)	CIFAR10		CIFAR100		Tiny-ImageNet		COIL-100		PathMNIST	
			Top-1 (%) ACC-k1	Top-3 (%) ACC-k3	Top-1 (%) ACC-k1	Top-3 (%) ACC-k3	Top-1 (%) ACC-k1	Top-3 (%) ACC-k3	Top-1 (%) ACC-k1	Top-3 (%) ACC-k3	Top-1 (%) ACC-k1	Top-3 (%) ACC-k3
MLP-Mixer	6.9	505	89.89	97.85	63.55	79.44	41.69	57.38	52.04	73.79	60.32	83.23
AS-MLP	27.5	89	91.77	98.54	64.58	81.22	41.78	58.34	67.00	90.04	53.18	80.68
ResMLP	3.6	238	90.41	98.35	65.48	81.39	45.95	63.14	73.78	90.06	71.27	93.79
VGG19	20.0	400	91.60	91.62	65.11	78.97	42.87	62.18	43.43	72.07	70.60	92.28
ConvNeXt V2	9.0	284	88.36	97.36	58.59	75.50	40.99	56.28	48.22	71.43	74.83	93.33
ViT	7.2	468	91.05	98.44	67.04	83.36	45.77	62.53	73.5	91.65	70.78	92.98
ConvFormer	26.7	83	91.57	98.53	61.88	78.32	45.56	61.69	56.33	78.75	74.11	90.91
DMixNet	6.3	404	**92.02**	**98.65**	**68.34**	**83.42**	**46.40**	**63.25**	**75.18**	**92.69**	**75.10**	**94.51**

The metrics displayed include the model Params, FLOPs, and accuracy metrics, specifically top-1, and top-3 accuracies on each dataset, labeled as “ACC-k1", “ACC-k3"

The best values among all compared methods are shown in bold superiority of the DMixNet framework, which enhances nonlinear processing capabilities through its unique dendritic structure design.

Figure 3 shows radar charts of evaluation metrics for different models on each dataset. The radar charts indicate that DMixNet performs excellently on various performance metrics (ACC-k1, ACC-k3, AUC, Recall, Precision, F1) across CIFAR-10, CIFAR-100, Tiny-ImageNet, and COIL-100, PathMNIST datasets, with its advantages becoming more apparent on the more complex Tiny-ImageNet dataset. We can visually observe that as the dataset difficulty increases, DMixNet's advantages grow. The flexibility of the dendritic structure allows dendritic neural units to adjust their configuration to perform different functions, thereby enhancing DMixNet's ability to adapt to complex variations.

On the PathMNIST dataset, DMixNet demonstrated its strong potential in medical image processing, achieving a Top-1 accuracy of 75.10%, surpassing all other models. DMixNet not only performed excellently in general image classification tasks but also showed strong potential in the more challenging field of medical imaging. This result indicates that DMixNet possesses high generalization ability and adaptability.

4.4 Complexity analysis

Table 2 presents a comparison of the structures and complexity analysis between MLP-Mixer and DMixNet. The time complexity of MLPMixer is $O(P \times S \times (D_1 + D_2 + K))$, where the terms D_1 and D_2 correspond to the hidden layer sizes in the channel-mixer and token-mixer,respectively,and K is the number of output classes. For DMixNet, the time complexity is $O\left(P \times S \times (S \times m_1 + D_2 + m_2 \times K)\right)$. Here, m_1 and m_2 represent the number of dendrites in the $\mathrm{DNU_E}$ and $\mathrm{DNU_C}$, respectively, are custom-defined hyperparameters. The MLP-Mixer's channel-mixer and token-mixer both consist of two fully connected layers, and the hidden layer of these fully connected layers results in high computational overhead. In contrast, DMixNet introduces DNU, where the complexity primarily depends on the hyperparameters m_1. The value of m_1 is typically set much lower than the hidden layer size D_1. The unique architecture of DMixNet reduces the number of matrix multiplications and the number of nodes that need to be computed. Compared to MLP-Mixer, DMixNet significantly reduces computational complexity while improving model performance. This remarkable difference is primarily attributed to the $\mathrm{DNU_E}$ and $\mathrm{DNU_C}$ modules in DMixNet. Through bionic design and efficient information processing mechanisms, these modules not only reduce computational costs but also enhance the model's biological interpretability and nonlinear processing capabilities. Therefore, DMixNet not only outperforms the traditional MLP-Mixer architecture in terms of performance but also exhibits higher computational efficiency and scalability.

4.5 Ablation experiment

In this section, we independently discuss the effectiveness, structural configuration, and parameter analysis of the $\mathrm{DNU_E}$ and the $\mathrm{DNU_C}$. Meanwhile, we further validate the rationale behind the design of the $\mathrm{DNU_E}$ and $\mathrm{DNU_C}$ sub-modules through individual ablation experiments.

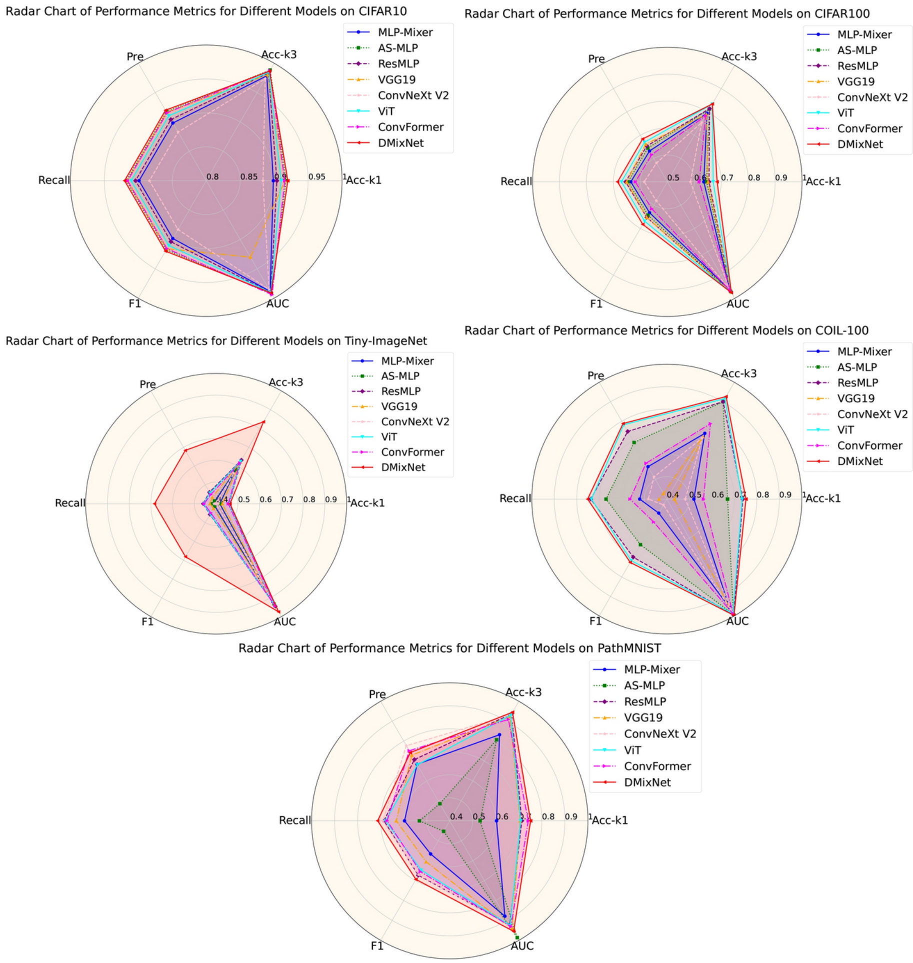

Fig. 3 Radar chart for different models performance metrics on different datasets. The metrics include precision (Pre), Recall, F1-score (F1), area under the curve (AUC), and accuracy at different thresholds (Acc-k1 and Acc-k3). Lines closer to the outer edge of the chart indicate better performance of the model on the corresponding metric

4.5.1 Effectiveness of $\mathrm{DNU_E}$

The experiment compared the performance of models utilizing the $\mathrm{DNU_E}$, traditional fully connected layer, and convolutional layer for three different channel-mixer methods. As shown in Table 3, the experiment firstly compares the performance of models using the $\mathrm{DNU_E}$ with dendritic branch m set to 2, two fully connected layers, and two convolutional layers as the channel-mixer. The results indicate $\mathrm{DNU_E}$ performs the best among them.

Furthermore, the experiment compared the performance of models using the $\mathrm{DNU_E}$ with dendritic branch m set to 1, single fully connected layer, and single convolutional layer as the channel-mixer. From Table 3, it can be observed that single fully connected layer and single convolutional layer outperform two fully connected layers and two convolutional

Table 2 Space complexity analysis of MLPMixer and DMixNet using CIFAR-10 as an example

Layer Type	MLPMixer	DMixNet	
Input Size	P × S (192 × 64)	P × S (192 × 64)	
Channel-mixer	Two fully connected layers (Hidden layer D_1 = 512)	$\mathrm{DNU_E}$	Synaptic layer Dendritic layer ($f=\varnothing$, $m_1=2$) Membrane layer Soma layer (f= *GELU*)
Token-mixer	Two fully connected layers (Hidden layer D_2= 2048)	Two fully connected layers (Hidden layer D_2 = 2048)	
Classifier	Single fully connected layer (Classes K=10)	$\mathrm{DNU_C}$	Synaptic layer Dendritic layer(f = *Softmax*, m_2 = 5) Soma layer ($f=\varnothing$)

It demonstrates the structural differences between MLPMixer and DMixNet

Table 3 Comparison results of different channel-mixer methods on CIFAR-10, including Params, FLOPs, Top-1, and Top-3 accuracy, labeled as "ACC-k1," "ACC-k3"

Channel-mixer	Params (K)	FLOPs (M)	Top-1(%) ACC-k1	Top-3(%) ACC-k3
Two fully connected layers	66.5	12.6	89.89	97.85
Two 1×1 convolution layers	66.5	12.6	89.59	98.03
$\mathrm{DNU_E}$(m=2)	16.9	3.2	91.35	98.47
Single fully connected layer	37.4	0.84	89.99	98.40
Single 1 × 1 convolution layer	8.7	0.84	90.03	98.41
$\mathrm{DNU_E}$(m=1)	8.6	0.05	90.68	98.54

layers, respectively. This suggests that simply stacking linear layers and increasing their size not only fails to effectively improve model accuracy but also reduces model performance due to the increased difficulty in network learning with the addition of layers. In contrast, for the $\mathrm{DNU_E}$, increasing the number of dendritic branch m enhances the learning capability of the entire network, thereby continuously improving model performance. The $\mathrm{DNU_E}$ achieves superior performance by enhancing the interpretability of the network, simulating the complex process of information processing by neural cells, and establishing a unique architecture. Whether with $\mathrm{DNU_E}$ with dendritic branch m set to 1 or 2, they outperform fully connected layers and convolutional layers in terms of performance. Moreover, $\mathrm{DNU_E}$ significantly reduces Params and FLOPs compared to traditional fully connected layers and convolutional layers. Whether with dendritic branch $m = 1$ or $m = 2$, $\mathrm{DNU_E}$ consistently demonstrates significant advantages in terms of Params and FLOPs, particularly when compared to the two fully connected layers.

Figure 4 shows the visualization comparison diagram of two fully connected layers and the $\mathrm{DNU_E}$ as channel-mixer on CIFAR10, Tiny-ImageNet, and PathMNIST. Compared to the feature maps generated by two fully connected layers, the feature maps produced by

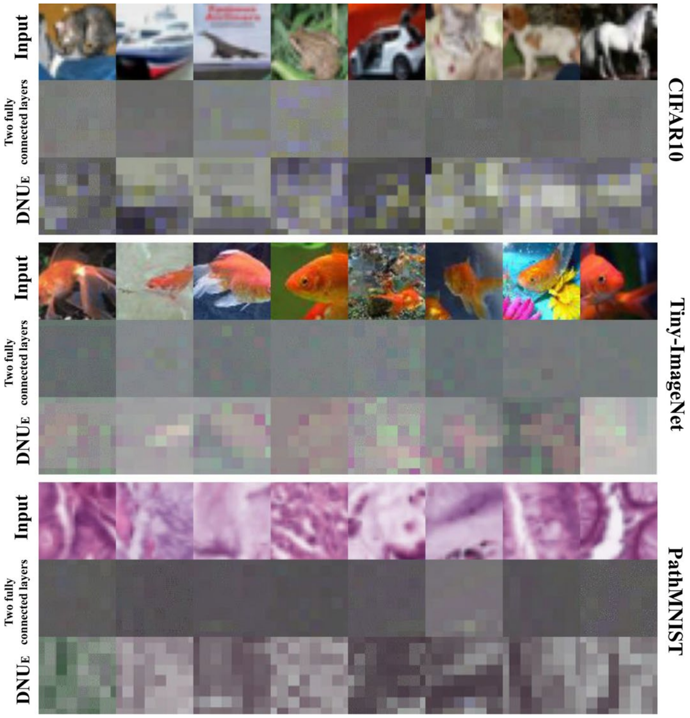

Fig. 4 The visualization comparison diagram of DMixNet. Input row shows the original input images from each dataset. Two-fully connected layers row represents the visualization results of images after channel-mixer processing through two fully connected layers. The $\mathrm{DNU_E}$ row shows the visualization results of images after channel-mixer processing through the $\mathrm{DNU_E}$ module

$\mathrm{DNU_E}$ exhibit clearer and more meaningful patterns, capturing more detailed and relevant features, whether on benchmark datasets like CIFAR-10, Tiny-ImageNet, or the medical image classification dataset PathMNIST. The specificity of dendrites enables the network to possess a more powerful feature extraction capability, allowing it to extract a wider range of global features. This empowers the $\mathrm{DNU_E}$ with a strong ability to fuse channel information effectively.

Table 4 displays the ablative experiments conducted on various modules within the $\mathrm{DNU_E}$. Through a combination of experiments, the experiments investigate whether the module uses activation functions, which type of activation function is employed, the position of the activation function, and whether the membrane layer is added. From the results in Table 4, it is evident that the design of the membrane layer effectively enhances the model's performance. Regarding activation functions, the GELU function significantly outperforms the sigmoid function. The GELU function has an output range identical to its input range, facilitating faster convergence. Additionally, the derivative of the GELU function approaches linearity near zero, aiding in alleviating the vanishing gradient problem. For the $\mathrm{DNU_E}$, whether GELU is used as the activation function or sigmoid is used as the activation function, the activation of the soma layer is superior to the activation of the dendritic layer. Through this ablation experiment, we determined the four-layer structure for $\mathrm{DNU_E}$, with GELU function in the soma layer.

Table 4 Internal ablation experiments of the DNU_E on CIFAR-10

Dendritic Layer Activation Function	Whether Membrane Layer Exists	Soma Layer Activation Function	Top-1 (%) ACC-k1	Top-3 (%) ACC-k3
✗	✗	✗	86.96	97.94
✗	✓	✗	87.66	97.75
GELU	✓	✗	89.18	97.77
RELU	✓	✗	86.19	97.63
✗	✓	GELU	**91.35**	**98.47**
✗	✓	RELU	90.59	98.44

It investigates the impact of different configurations in various layers within the dendritic neural unit on the DNU_E module

The best values among all variants are shown in bold

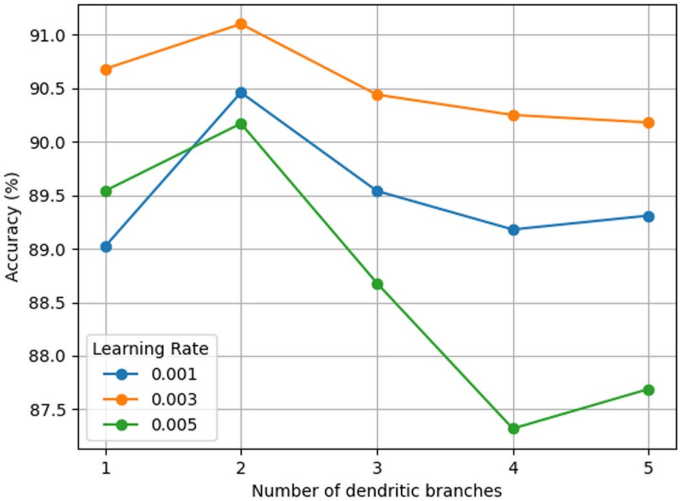

Fig. 5 The impact of dendritic density (number of dendritic branches) and network learning rate on the accuracy of the DNU_E module

Then, we delve into the discussion concerning the dendritic density of the DNU_E and the network's learning rate within the DNU_E as the channel-mixer. We set the dendritic branch m to be 1, 2, 3, 4, 5 and the learning rate to be 1e-1, 1e-3, 1e-5. Figure 5 illustrates the impact of dendritic layer and learning rate on the dendritic channel network. From Fig. 5, it is evident that increasing dendritic density does not necessarily improve performance. Instead, an increase in density after reaching the peak may lead to redundant synaptic nodes, consequently diminishing network performance. For the DNU_E, optimal settings are achieved with the number of the dendritic branch m set to 2 and the learning rate to 0.003.

The design of the DNU_E with high individual neuron complexity and low branch density enables the DNU_E to provide powerful feature extraction while maintaining efficient computation, which makes it perform well when dealing with complex data and tasks.

Table 5 Ablation study of the $\mathrm{DNU_C}$ on CIFAR10

Architecture	Params (M)	FLOPs (M)	Training Time (sec)	Inference Time (sec)	Top-1 (%) ACC-k1	Top-3(%) ACC-k3
MLP-Mixer	6.9	505	253.4	0.5479	89.89	97.85
MLP-Mixer +$\mathrm{DNU_E}$	6.4	430	579.9	0.5486	91.35	98.47
MLP-Mixer +$\mathrm{DNU_C}$	6.8	505	478.6	0.5482	90.66	98.58
DMixNet	6.3	404	663.5	0.5495	92.02	98.65

It summarizes the trade-off between computational cost and performance across different architectures

Table 6 Internal ablation experiments of the $\mathrm{DNU_C}$ on CIFAR-10

Dendritic Layer Activation Function	Whether Membrane Layer Exists	Soma Layer Activation Function	Top-1 ACC-k1	Top-3 ACC-k3
✗	✗	✗	89.38	98.16
✗	✓	✗	89.04	98.58
Sigmoid	✗	✗	90.19	98.44
Softmax	✗	✗	**90.66**	**98.78**
✗	✗	Softmax	90.41	98.44
✗	✗	Sigmoid	80.43	95.98

It investigates the impact of different configurations in various layers within the dendritic neural unit on the $\mathrm{DNU_C}$ module

The best values among all variants are shown in bold

4.5.2 Effectiveness of $\mathrm{DNU_C}$

Table 5 presents the results of the ablation study for $\mathrm{DNU_C}$. Whether $\mathrm{DNU_C}$ is added to the base MLP-Mixer or to the MLP-Mixer already enhanced with $\mathrm{DNU_E}$, $\mathrm{DNU_C}$ effectively improves the classification performance of the model. Benefiting from the structural advantages of dendrites, the $\mathrm{DNU_C}$ exhibits enhanced ability to capture nonlinear relationships between complex data. This is achieved through synaptic connections, enabling better combination mapping of input features and ultimately improving the classification performance of the model. However, while DMixNet reduces both the Params and FLOPs, its training time and inference time are significantly longer than the MLP-Mixer. This counterintuitive result is primarily due to the lack of parallelization in the multi-branch structure of the DNU. Therefore, although DMixNet demonstrates advantages in model accuracy and resource efficiency, there is still room for improvement in computational time.

Table 6 presents ablative experiments conducted on various modules of the $\mathrm{DNU_C}$. It is evident that increasing the complexity of individual dendritic neurons by adding the member layer module, leads to a decrease in classification accuracy. Regarding activation functions, softmax yields the best results. The softmax function transforms input real numbers into positive probability distributions, ensuring that all output values fall between 0 and 1, with their sum equaling 1. It is superior to the sigmoid function in multi-class classification problems. The $\mathrm{DNU_C}$ rely more on dendritic local computational capabilities, and activating the dendritic layer enhances the overall performance of the classifier. Through this ablation experiment, we determined the three-layer structure for $\mathrm{DNU_C}$, with softmax function in the dendritic layer.

Figure 6 illustrates the impact of dendritic density and the network's learning rate within the $\mathrm{DNU_C}$ as classifier. It can be observed that compared to the $\mathrm{DNU_E}$, the $\mathrm{DNU_C}$ requires higher dendritic complexity. However, an excessive increase in the number of dendritic

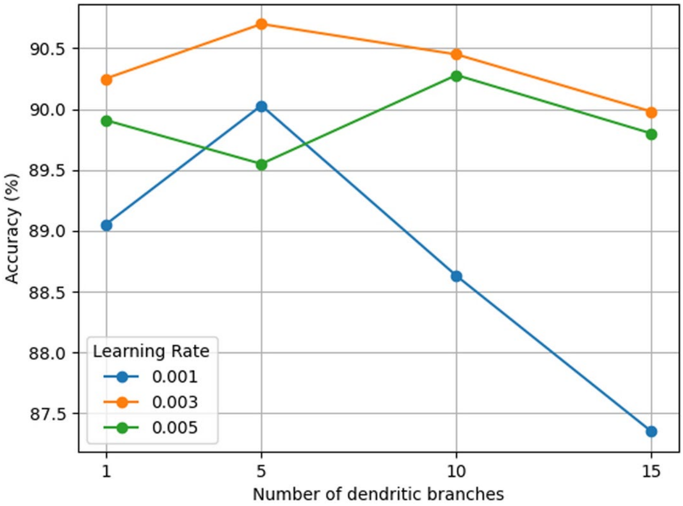

Fig. 6 The impact of dendritic density (number of dendritic branches) and network learning rate on the accuracy of the $\mathrm{DNU_C}$ module

layers, up to 15 layers, leads to a decline in classification performance due to redundant synaptic connections. For the $\mathrm{DNU_C}$, the optimal configuration involves setting the dendritic branch m to 5 and the learning rate to 0.003.

The design of $\mathrm{DNU_C}$, featuring low complexity for individual neurons, high dendritic density, and the use of local dendritic computation, enhances the mapping of complex input feature combinations, resulting in improved classification performance for $\mathrm{DNU_C}$.

5 Conclusion

In this paper, we propose a dendritic multi-layered perception architecture DMixNet. The DMixNet combines the characteristics of dendrites with the structural advantages of the dendritic neural unit, achieving enhanced channel fusion capabilities and improved feature classification performance. Compared to fully connected layers, the dendritic neural unit is based on the theory of biological dendritic neural systems and has stronger channel feature extraction capabilities and greater structural flexibility. Its structure is flexible and can be adjusted according to the needs of the module to cope with different functions. Based on the DNU, we propose the $\mathrm{DNU_E}$ module and $\mathrm{DNU_C}$ module to optimize the entire MLP-Mixer framework. Through network ablation experiments, we validate the effectiveness of the dendritic neural unit modules. Furthermore, we compare the performance of DMixNet with other state-of-the-art models on three widely used classification datasets, namely CIFAR-10, CIFAR-100, and Tiny-ImageNet. The experimental results demonstrate the superior performance of the DMixNet in classification tasks.

While the dendritic model offers higher accuracy compared to fully connected layers, the current multi-branch structure lacks parallel processing, resulting in excessively long training times. We are actively working on optimizing this issue. Additionally, as the dendritic model scales up, we have observed a tendency for dendritic redundancy, which can diminish the model's efficiency. To address this, we are focusing on pruning the dendritic branches to reduce redundancy. By refining these aspects, we aim to ensure that DMixNet remains competitive across various application scenarios.

Acknowledgements 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.

References

Bassett DS, Gazzaniga MS (2011) Understanding complexity in the human brain. Trends Cogn Sci 15(5):200–209

Benz P, Zhang C, Ham S, Karjauv A, Kweon IS (2021) Robustness comparison of vision transformer and MLP-mixer to CNNs. In: CVPR 2021 workshop on adversarial machine learning in real-world computer vision systems and online challenges (AML-CV), vol 7, pp 21–24

Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: International conference on learning representations (ICLR)

Egrioglu E, Bas E (2024) A new deep neural network for forecasting: deep dendritic artificial neural network. Artif Intell Rev 57(7):171

Egrioglu E, Grosan C, Bas E (2023) A new genetic algorithm method based on statistical-based replacement for the training of multiplicative neuron model artificial neural networks. J Supercomput 79(7):7286–7304

Elsaesser R, Paysan J (2007) The sense of smell, its signalling pathways, and the dichotomy of cilia and microvilli in olfactory sensory cells. BMC Neurosci 8:1–13

Fan F-L, Li Y, Peng H, Zeng T, Wang F (2023) Towards NeuroAI: introducing neuronal diversity into artificial neural networks. arXiv preprint arXiv:2301.09245

Fei-Fei L, Fergus R, Perona P (2009) Tiny imagenet challenge. In: Course: CS231N convolutional neural networks for visual recognition, Stanford University. http://cs231n.stanford.edu/2019/

Gao S, Zhou M, Wang Y, Cheng J, Yachi H, Wang J (2018) Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Trans Neural Networks Learn Syst 30(2):601–614

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Häusser M, Mel B (2003) Dendrites: bug or feature? Curr Opin Neurobiol 13(3):372–383

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the ieee conference on computer vision and pattern recognition, pp 770–778

Hou Q, Jiang Z, Yuan L, Cheng M-M, Yan S, Feng J (2022) Vision permutator: a permutable MLP-like architecture for visual recognition. IEEE Trans Pattern Anal Mach Intell 45(1):1328–1334

Koch C, Poggio T (1985) Biophysics of computation: neurons, synapses, and membranes. Biol Cybern 53(5):253–265

Konnerth A, Llano I, Armstrong CM (1990) Synaptic currents in cerebellar purkinje cells. Proc Natl Acad Sci 87(7):2662–2665

Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images

Le C, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

Lee C, Hasegawa H, Gao S (2022) Complex-valued neural networks: a comprehensive survey. IEEE/CAA J Automatica Sinica 9(8):1406–1426

Lian D, Yu Z, Sun X, Gao S (2021) AS-MLP: an axial shifted MLP architecture for vision. In: International conference on learning representations
Liu H, Dai Z, So D, Le QV (2021) Pay attention to MLPs. Adv Neural Inf Process Syst 34:9204–9215
Liu R, Li Y, Tao L, Liang D, Zheng H-T (2022) Are we ready for a new paradigm shift? A survey on visual deep MLP. Patterns **3**(7)
Liu Z, Zhang Z, Lei Z, Omura M, Wang R-L, Gao S (2024) Dendritic deep learning for medical segmentation. IEEE/CAA J Automatica Sinica 11(3):803–805
Loshchilov I, Hutter F (2018) Decoupled weight decay regularization. In: International conference on learning representations
McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133
Nene SA, Nayar SK, Murase H et al (1996) Columbia object image library (coil-20)
Poirazi P, Papoutsi A (2020) Illuminating dendritic function with computational models. Nat Rev Neurosci 21(6):303–321
Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386
Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
Shepherd GM (2015) Foundations of the Neuron doctrine. Oxford University Press, Oxford
Shipp S (2007) Structure and function of the cerebral cortex. Curr Biol 17(12):443–449
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
Stuart G, Spruston N, Häusser M (2016) Dendrites. Oxford University Press, Oxford
Tang C, Zhao Y, Wang G, Luo C, Xie W, Zeng W (2022) Sparse MLP for image recognition: is self-attention really necessary? Proc AAAI Conf Artif Intell 36:2344–2351
Telley L, Jabaudon D (2018) A mixed model of neuronal diversity. Nature 555(7697):452–454
Tolstikhin IO, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, Yung J, Steiner A, Keysers D, Uszkoreit J et al (2021) MLP-Mixer: an all-MLP architecture for vision. Adv Neural Inf Process Syst 34:24261–24272
Touvron H, Bojanowski P, Caron M, Cord M, El-Nouby A, Grave E, Izacard G, Joulin A, Synnaeve G, Verbeek J et al (2022) ResMLP: feedforward networks for image classification with data-efficient training. IEEE Trans Pattern Anal Mach Intell 45(4):5314–5321
Wei G, Zhang Z, Lan C, Lu Y, Chen Z (2023) Active token mixer. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, pp 2759–2767
Woo S, Debnath S, Hu R, Chen X, Liu Z, Kweon IS, Xie S (2023) ConvNeXt V2: co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16133–16142
Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, Pfister H, Ni B (2023) Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data 10(1):41
Yu T, Li X, Cai Y, Sun M, Li P (2022) S2-MLP: spatial-shift MLP architecture for vision. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 297–306
Yu W, Luo M, Zhou P, Si C, Zhou Y, Wang X, Feng J, Yan S (2022) Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10819–10829
Yu W, Si C, Zhou P, Luo M, Zhou Y, Feng J, Yan S, Wang X (2024) Metaformer baselines for vision. IEEE Trans Pattern Anal Mach Intell 46(2):896–912
Zador A, Escola S, Richards B, Ölveczky B, Bengio Y, Boahen K, Botvinick M, Chklovskii D, Churchland A, Clopath C et al (2023) Catalyzing next-generation artificial intelligence through neuro AI. Nat Commun 14(1):1597
Zhang H, Dong Z, Li B, He S (2022) Multi-scale MLP-mixer for image classification. Knowl-Based Syst 258:109792
Zhang Z, Lei Z, Omura M, Hasegawa H, Gao S (2024) dendritic learning-incorporated vision transformer for image recognition. IEEE/CAA J Automatica Sinica 11(2):539–541

Authors and Affiliations

Weixiang Xu[1] · Yaotong Song[1] · Shubham Gupta[2] · Dongbao Jia[3] · Jun Tang[4] · Zhenyu Lei[1] · Shangce Gao[1]

✉ Shangce Gao
gaosc@eng.u-toyama.ac.jp

Weixiang Xu
d2378002@ems.u-toyama.ac.jp

Yaotong Song
m23c1048@ems.u-toyama.ac.jp

Shubham Gupta
shubham.gupta@mnnit.ac.in

Dongbao Jia
dbjia@jou.edu.cn

Jun Tang
juntang@wicresoft.com

Zhenyu Lei
leizg@eng.u-toyama.ac.jp

1 Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan

2 Department of Mathematic, Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, India

3 School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China

4 Wicresoft Co Ltd, 13810 SE Eastgate Way, Bellevue, WA 98005, USA