

---

# Dendritic Integration Inspired Artificial Neural Networks Capture Data Correlation

---

Chongming Liu<sup>a,b</sup>, Jingyang Ma<sup>a,b</sup>, Songting Li<sup>a,b,c,1</sup>, and Douglas Zhou<sup>a,b,c,d,1</sup>

<sup>a</sup>School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, China

<sup>b</sup>Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, China

<sup>c</sup>Ministry of Education Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai, China

<sup>d</sup>Shanghai Frontier Science Center of Modern Analysis, Shanghai Jiao Tong University, Shanghai, China

## Abstract

Incorporating biological neuronal properties into Artificial Neural Networks (ANNs) to enhance computational capabilities is under active investigation in the field of deep learning. Inspired by recent findings indicating that dendrites adhere to quadratic integration rule for synaptic inputs, this study explores the computational benefits of quadratic neurons. We theoretically demonstrate that quadratic neurons inherently capture correlation within structured data, a feature that grants them superior generalization abilities over traditional neurons. This is substantiated by few-shot learning experiments. Furthermore, we integrate the quadratic rule into Convolutional Neural Networks (CNNs) using a biologically plausible approach, resulting in innovative architectures—Dendritic integration inspired CNNs (Dit-CNNs). Our Dit-CNNs compete favorably with state-of-the-art models across multiple classification benchmarks, e.g., ImageNet-1K, while retaining the simplicity and efficiency of traditional CNNs. All source code are available at <https://github.com/liuchongming1999/Dendritic-integration-inspired-CNN-NeurIPS-2024>.

## 1 Introduction

While the Artificial Neural Network (ANN) framework has made significant advancements towards solving complex tasks, it still faces problems that are rudimentary to real brains [6]. A notable distinction between the modern ANN framework and the human brain is that the former relies on a significant number of training samples, which consumes large amounts of energy, whereas the latter runs on extremely low power (<20 watts) and possesses a strong generalization capability based on few-shot learning. Studies have demonstrated that incorporating dendritic features in ANNs can alleviate these issues and enhance overall performance [52, 38, 30]. However, it is difficult to quantify the nonlinear integration of dendrites, which is an essential property that allows individual neurons to perform complex computations [43, 49]. As a result, dendritic-inspired models often employ linear integration with nonlinear activation functions such as ReLU and Sigmoid [21, 37].

In light of recent studies revealing that the somatic responses of biological neurons to multiple synaptic inputs on dendrites follow a quadratic integration rule [15, 28], we explore the computational benefit of the quadratic neuron model [41]. This model substitutes the linear integration and nonlinear

---

<sup>1</sup>Corresponding author, [songting@sjtu.edu.cn](mailto:songting@sjtu.edu.cn), or [zdz@sjtu.edu.cn](mailto:zdz@sjtu.edu.cn)

activation function of traditional point neurons with quadratic integration as follows<sup>1</sup>:

$$f(x) = \sigma(w \cdot x + b) \rightarrow f(x) = x^T Ax + w \cdot x + b. \quad (1)$$

The expected value of a quadratic term  $E_{x \sim \mathcal{D}}[a_{ij}x_i x_j]$  closely relates to the covariance between variables  $x_i$  and  $x_j$  ( $x_i$  and  $x_j$  are the  $i$ -th and the  $j$ -th components of vector  $x$ , respectively), suggesting an inherent capability for capturing input correlation. This attribute allows biological neurons to exhibit remarkable performance in correlation detection tasks [1, 29]. Here, we theoretically demonstrate that quadratic neurons indeed naturally capture correlation within structured data, which is pivotal in numerous machine learning applications [7], such as language generation [5] and video understanding [23]. This intrinsic property endows quadratic neurons with superior generalization capabilities compared to traditional neurons, as evidenced by few-shot learning experiments. We further propose a biologically plausible method to integrate them into Convolutional Neural Networks (CNNs). Applying this approach to ResNet [16] and ConvNeXt [34] results in novel CNN models—Dendritic integration inspired ResNet (Dit-ResNet) and Dendritic integration inspired ConvNeXt (Dit-ConvNeXt), respectively. Evaluations on CIFAR-10 and CIFAR-100 datasets demonstrate that Dit-ResNets significantly enhance test accuracy by merely replacing a single layer of point neurons with quadratic neurons. On the ImageNet-1K dataset [10], Dit-ConvNeXts demonstrate significant enhancements in top-1 accuracy with only a one percent increase in parameters, exhibiting competitive performance compared to state-of-the-art models. Ablation studies further substantiate the effectiveness of quadratic neurons and their capability for capturing data correlation.

This paper is structured as follows: Section 2 reviews previous works related to dendritic-inspired models and high-order interactive operations in neural networks. Section 3 presents a theoretical analysis of the computational benefits derived from quadratic neurons, as demonstrated through few-shot learning experiments. Section 4 describes the architecture of Dit-CNNs and evaluates their performance on several computer vision benchmark datasets. Section 5 concludes the paper. All numerical experiments in this paper are conducted using Python and executed on 8 Tesla A100 computing cards with a 7nm GA100 GPU, featuring 6,912 CUDA cores and 432 tensor cores.

## 2 Related work

**Dendritic-inspired computational models.** The significant role of dendrites in the nonlinear integration within neural systems is well-recognized, enabling neurons to perform intricate tasks [43, 49, 2]. Recently, numerous studies have explored the implementation of dendritic features in ANNs from different aspects, yielding encouraging results. In [52, 21, 37], the integration of dendritic morphology into ANNs has led to higher test accuracies on simple tasks compared to traditional ANNs. Furthermore, dendritic plasticity rules have been employed to develop learning algorithms in [38, 36], effectively replacing the non-biological backpropagation algorithm and achieving enhanced performance on classification tasks with small data sizes. Additionally, in [30], dendritic nonlinearity is considered by adding input to each layer with the Hadamard product rule, which has shown improvements in approximation and classification experiments.

**Neural networks with high-order interactions.** Recent advancements in the design of deep learning architectures have been predominantly driven by the ability to capture high-order statistics among inputs and features. Over recent decades, there has been a pivotal shift in the foundational neural networks used for computer vision tasks: moving from convolution-based architectures [16, 24, 56] to Transformer-based models [11, 33, 9]. This transition has been driven by the development of the self-attention mechanism, which facilitates quadratic interactions among inputs. Additionally, a novel category of neural networks, referred to as Mamba [14], has demonstrated exceptional performance in tasks that require the processing of extensive high-order statistical information, such as video understanding [32, 59, 27]. The backbone of Mamba’s architecture is a state space model that enables high-order interactions among inputs. Moreover, the integration of high-order spatial interactions into CNNs has also been shown to enhance performance significantly [40]. These developments underscore the significance of high-order statistical information in tackling large-scale complex tasks and suggest that incorporating high-order interactions as a foundational aspect of model design is an effective strategy.

---

<sup>1</sup> $A$  can always be considered a symmetric matrix, attributable to the quadratic form present in  $f(x)$ .

Table 1: The overview of current works with quadratic format.

Reference	Quadratic format	How quadratic operation is used	Biological interpretation	Theory for generalization
[54],[12],[4], [53]	$f(x) = (w_a x)(w_b x)$	Pixel-Wise	N	N
[48],[20],[35],[60]	$f(x) = x^T A x$	Pixel-Wise	N	N
Dit-CNNs	$f(x) = x^T A x$	Channel-Wise	Y	Y

Previous works have explored the concept of quadratic integration as a sophisticated alternative to traditional linear summation. Table 1 summarizes various neuron models formulated in quadratic formats, along with their relevant references. The one-rank format of quadratic neurons, when applied across entire networks, is documented in [54, 12, 4, 53], demonstrating enhanced performance in classification tasks. Studies such as [48, 20, 35, 60] have implemented quadratic neurons on a pixel-wise basis within convolution layers, yielding modest improvements in test accuracy. In contrast, Table 1 illustrates how our Dit-CNNs approach departs from these models by integrating a channel-wise quadratic operations with biological interpretation and providing a theoretical analysis of the computational advantages offered by quadratic neurons.

### 3 Quadratic neurons possess enhanced generalization capabilities

This section provides a theoretical analysis demonstrating how quadratic neurons enhance generalization capabilities by effectively capturing correlation within structured data. Additional numerical experiments corroborate this assertion. The detailed proofs can be found in Appendix A.

#### 3.1 Binary classification for normal distributions

Assume that the data points of two classes are equally sampled from two different non-degenerate normal distributions:  $class_1 \sim N(\mu_1, \Sigma_1)$ ,  $class_2 \sim N(\mu_2, \Sigma_2)$ , where  $\mu_j \in \mathbb{R}^d$ ,  $\Sigma_j \in \mathbb{R}^{d \times d}$  ( $j = 1, 2$ ). Then the optimal classifier  $y_{opt}(x) : \mathbb{R}^d \rightarrow \{1, 2\}$  can be defined according to the sampling probability:

$$y_{opt}(x) = \arg \max_{j \in \{1, 2\}} p_j(x), \quad (2)$$

where  $p_j(x)$  is the probability (density function) of sampling point  $x$  from distribution  $N(\mu_j, \Sigma_j)$ .

If a single quadratic neuron, as described in Equation (1) ( $f(x) = x^T A x + w \cdot x + b$ , where  $A$  is a symmetric matrix), is used to solve the above binary classification task, we can prove the following theorems:

**Theorem 1.** (Existence) *The critical points with respect to the cross-entropy loss  $L(A, w, b)$  are given as follows:*

$$A^* = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w^* = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b^* = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right).$$

i.e.

$$\frac{\partial L}{\partial A} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L}{\partial w} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L}{\partial b} \Big|_{A^*, w^*, b^*} = 0,$$

where

$$L(A, w, b) = \frac{1}{2} \left[ E_{x \sim class_1} \left( \log(1 + e^{f(x)}) \right) + E_{x \sim class_2} \left( \log(1 + e^{-f(x)}) \right) \right].$$

Moreover, the corresponding classifier generated by this formula is the same as the theoretically optimal classifier in Equation (2).

**Theorem 2.** (Uniqueness) *Consider the conditional cross-entropy loss defined on a set  $\Omega \in \mathcal{M}(\mathbb{R}^d)$ , where  $\mathcal{M}(\mathbb{R}^d)$  denotes the Lebesgue  $\sigma$ -algebra on  $\mathbb{R}^d$ :*

$$L(A, w, b | \Omega) = \frac{1}{2} \left[ E_{x \sim class_1, x \in \Omega} \left( \log(1 + e^{f(x)}) \right) + E_{x \sim class_2, x \in \Omega} \left( \log(1 + e^{-f(x)}) \right) \right],$$

where

$$E_{x \sim \text{class}_j, x \in \Omega} (g(x)) = \int_{\Omega} g(x) p_j(x) d\mu,$$

then the unique solution satisfying

$$\frac{\partial L(A, w, b | \Omega)}{\partial A} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b | \Omega)}{\partial w} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b | \Omega)}{\partial b} \Big|_{A^*, w^*, b^*} = 0$$

for every  $\Omega \in \mathcal{M}(\mathbb{R}^d)$  is given by:

$$A^* = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w^* = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b^* = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right).$$

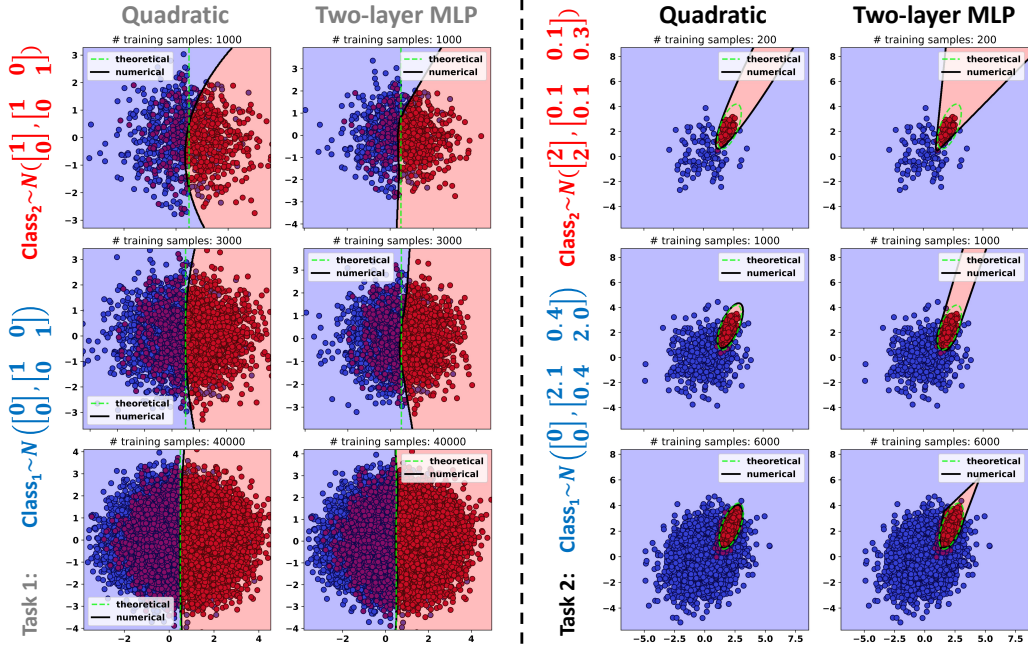


Figure 1: Visualization of classifier boundaries for two models (a single quadratic neuron with 2-dimensional inputs and a two-layer MLP comprising 2-ReLU(10)-1) across two distinct tasks. For each task, the impact of varying training sample sizes is examined. Model comparisons are performed under uniform conditions, utilizing identical random seeds and hyperparameters for fairness. Training is executed using gradient descent with a learning rate of 0.1 over 10,000 epochs. The term "theoretical" denotes the optimal classifier's boundary as specified in Equation (2), while "numerical" represents the empirical classification boundary obtained from the model.

Theorem 1 identifies an analytical critical point for a quadratic neuron with cross-entropy loss, which emerges as the optimal classifier. On the other hand, Theorem 2 establishes the uniqueness of this critical point under conditional cross-entropy loss, where  $\Omega$  denotes the regions occupied by distinct batches of data points. Under these conditions, the quadratic parameters will converge to this unique critical point when using the stochastic gradient descent (SGD) algorithm. It is demonstrated that, unlike traditional neuron, quadratic neuron inherently possesses the ability to converge to the optimal classification solution by directly including the covariance matrix. This finding is supported by simulated data in Task 1 (identical covariance) and Task 2 (non-identical covariance) shown in Figure 1, which indicates that quadratic neuron consistently achieves the theoretically optimal classification outcome. Moreover, in comparison to a multi-layer perceptron (MLP) with two layers, quadratic neuron surpass traditional neuron in Task 2 which require capturing correlation information from data distributions (require fewer training samples to converge to the optimal solution). This highlights the unique capability of quadratic neurons to identify internal correlations within structured data, offering a possible explanation for their superior generalization capability compared to traditional neurons, a topic that will be further explored in numerical experiments below.

### 3.2 Few-shot learning experiments on MNIST and Arabic MNIST

From the above classification tasks for normal distributions as depicted in Figure 1, it is suggested that quadratic neurons may require fewer training samples in comparison to traditional neurons. We next examine this few-shot learning capability of quadratic neurons on the MNIST [25] and Arabic MNIST datasets [19]. The results, presented in Figure 2, indicate that quadratic neurons outperform MLP with two layers when trained with a limited number of samples on both datasets. This evidence shows the enhanced generalization ability of quadratic neurons in applications. Further insights into the mechanism through which quadratic neurons capture data correlation in the MNIST dataset, and a related theorem concerning multi-class classification for normal distributions, are detailed in Appendix B.

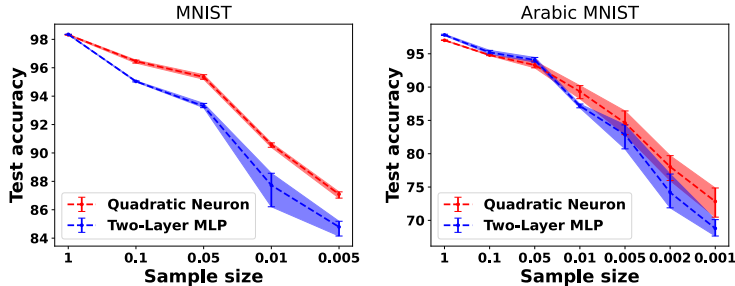


Figure 2: **Performance of two models on few-shot learning tasks with MNIST and Arabic MNIST datasets.** The first model consists of 10 quadratic neurons with a 784-dimensional input, while the second model is a MLP with the configuration of 784-ReLU(8000)-10. Both models are evaluated under identical conditions using the same training protocol, which includes SGD with a learning rate of 0.1 and a batch size of 100 across 20 epochs. The term 'Sample size' refers to the ratio of the number of training samples to the full training set. Experiments are conducted for each model and sample ratio across ten runs, and the resulting test accuracy is depicted through an error bar plot (error bar represents the upper and lower bound for test accuracy).

### 4 Integrating quadratic neurons into CNNs with biological plausibility

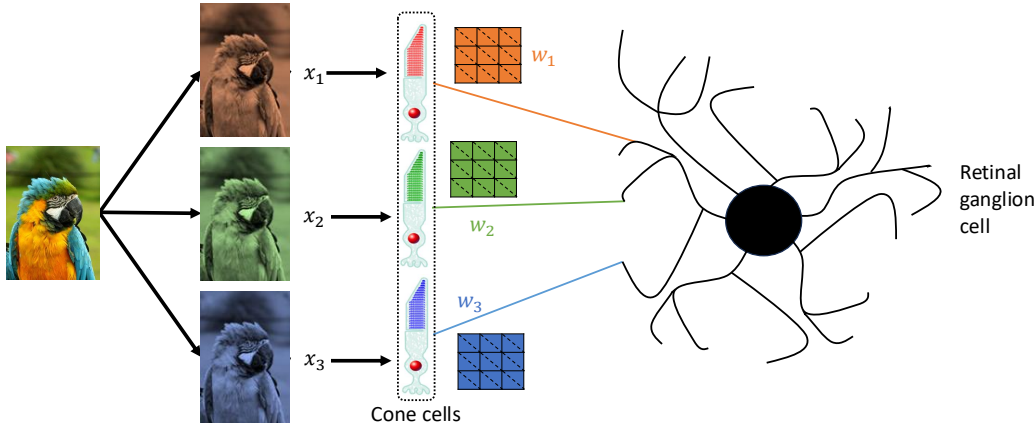


Figure 3: **Schematic of the biological interpretation of Dit-CNNs.** Dit-CNNs is inspired by neural networks in the visual system. For example, different types of cone cells encode various color (channel) information, and retinal ganglion cells receive inputs from multiple types of cone cells [22], the responses can be modeled as having receptive fields (convolutional kernels) related to different color channels ( $w_1 * x_1, w_2 * x_2, w_3 * x_3$ ). When multiple channel inputs are present, traditional CNNs simply linearly sum the corresponding responses. In contrast, dendritic neurons integrate these inputs with an additional quadratic term based on the dendritic quadratic integration rule. This approach leads to the formulation of Dit-CNNs after simplification.

The concept of convolution [13], inspired by the discovery of the receptive field in the cat visual cortex [18], laid the foundation for CNNs [26] which have been designed to efficiently tackle large-scale vision tasks. The underlying biological interpretation of convolution is that each neuron exhibits varying responses at different image locations based on its receptive field (convolution kernel). Traditionally, CNNs have focused solely on the linear integration of inputs from presynaptic neurons. Specifically, for an input  $X \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ , where  $C_{in}$  represents the number of input layer neurons (channels), and  $H_{in}$  and  $W_{in}$  denote the height and width of the input feature, respectively. Given convolution kernels  $w \in \mathbb{R}^{C_{in} \times C_{out} \times (2l+1) \times (2l+1)}$ , the convolution output,  $Y = Conv(X) \in \mathbb{R}^{C_{out} \times H_{out} \times W_{out}}$ , represents the response of neuron  $i$  at location  $[j, k]$  to inputs from locations  $[j-l : j+l, k-l : k+l]$  across all input layer neurons:

$$Y[i, j, k] = \sum_{m=1}^{C_{in}} w[m, i, :, :] * X[m, j-l : j+l, k-l : k+l]. \quad (3)$$

However, acknowledging that neurons with dendrites obey a quadratic integration rule, we propose a novel approach that incorporates quadratic neurons into CNNs as depicted in Figure 3. This adaptation incorporates a quadratic integration term among inputs from different neurons (channels). Specifically, with a quadratic integration coefficient  $A \in \mathbb{R}^{C_{out} \times C_{in} \times C_{in}}$ , the output neuron’s response in Equation (3) is modified as follows:

$$Y[i, j, k] = \sum_{m=1}^{C_{in}} w[m, i, :, :] * X[m, j-l : j+l, k-l : k+l] + \underbrace{X^T[:, j, k] A [i, :, :] X[:, j, k]}_{\text{Quadratic integration}}. \quad (4)$$

#### 4.1 Evaluations on CIFAR-10 and CIFAR-100

**Dataset and models.** Our initial experiments utilize the CIFAR dataset, which includes 50,000 training images and 10,000 testing images. The experimental setup follows the ResNet configurations as outlined in [16], comprising models such as ResNet-20, ResNet-32, ResNet-56 and ResNet-110. To reduce computational demands, we incorporate quadratic neurons specifically into one layer—namely, the second layer of the first block in the second stage of the ResNet architecture [16]. Additionally, to address issues related to gradient dynamics, such as gradient explosion, a Layer Normalization (LN) layer [3] is implemented immediately before the layer equipped with quadratic neurons.

Table 2: Comparative performance of Dit-ResNets and structurally similar models on CIFAR.

Model	# Param. (CIFAR10)	Acc. (CIFAR10)	# Param. (CIFAR100)	Acc. (CIFAR100)
ResNet-20[16]	0.27M	91.25%	0.30M	67.26±0.68%
QResNet-20[12]	0.81M	92.22%	0.84M	67.82±0.52%
QuadraResNet-20[54]	0.81M	92.21%	0.84M	68.02±0.44%
Dit-ResNet-20	0.30M	<b>92.66%</b>	0.33M	<b>68.66±0.34%</b>
ResNet-32[16]	0.46M	92.49%	0.49M	68.52±0.55%
QResNet-32[12]	1.39M	93.10%	1.42M	69.41±0.48%
QuadraResNet-32[54]	1.39M	93.11%	1.42M	69.54±0.44%
Dit-ResNet-32	0.49M	<b>93.17%</b>	0.52M	<b>69.68±0.32%</b>
ResNet-56[16]	0.86M	93.03%	0.89M	70.17±0.67%
QResNet-56[12]	2.55M	93.66%	2.58M	71.21±0.44%
QuadraResNet-56[54]	2.55M	93.79%	2.58M	70.98±0.76%
Dit-ResNet-56	0.89M	<b>93.90%</b>	0.92M	<b>71.40±0.35%</b>
ResNet-110[16]	1.73M	93.57%	1.76M	70.84±0.76%
QResNet-110[12]	5.17M	93.88%	5.20M	71.58±0.87%
QuadraResNet-110[54]	5.17M	93.77%	5.20M	71.72±0.81%
Dit-ResNet-110	1.76M	<b>94.33%</b>	1.79M	<b>72.40±0.85%</b>

**Experimental settings.** The training process incorporates SGD with a weight decay of 0.0001, a batch size of 128, and a momentum of 0.9. Following the recommendations from [12], quadratic

integration parameters are initialized to zero, and a distinct learning rate is adopted for these parameters. Initially, the learning rate for the quadratic integration matrix is set at 1, and 0.1 for all other parameters. These rates are reduced by a factor of ten at 80 and 120 epochs, with training concluding at 160 epochs. Data augmentation procedures mirror the original protocol: each image is padded by 4 pixels on each side, followed by random cropping of a  $32 \times 32$  section from the padded image or its horizontal flip. For testing, a single view of the original  $32 \times 32$  image is evaluated.

**Results.** Table 2 presents a comparative analysis of CIFAR performance between our Dit-ResNets and the original ResNet model [16], as well as two adaptations that incorporate quadratic neurons [12, 54], which have shown superior outcomes among existing approaches with quadratic formats [48, 20, 35, 60, 4]. For the CIFAR-10 dataset, given the extensive benchmarking in prior studies, we conduct ten runs of our Dit-ResNets, reporting the optimal outcome for comparative analysis. For CIFAR-100, we independently execute all models, documenting average test accuracy and variance (mean $\pm$ std). Our experiments indicate significant performance enhancements on both CIFAR-10 and CIFAR-100 datasets after integrating quadratic neurons into a single layer of ResNet, confirming their efficacy. Furthermore, when comparing a deeper ResNet model with our Dit-ResNet (e.g., ResNet-110 vs. Dit-ResNet-56), our approach not only boosts performance but also reduces training overhead. This suggests that augmenting the model with quadratic integration parameters is more effective than merely increasing network depth—a traditionally favored strategy. In comparison to other models employing quadratic neurons [12, 54], our Dit-ResNets consistently achieve the highest test accuracy with the fewest parameters. This underscores the efficacy of our approach in leveraging the computational advantages of quadratic neurons. These neurons, as discussed in Section 3, demonstrate enhanced generalization capabilities.

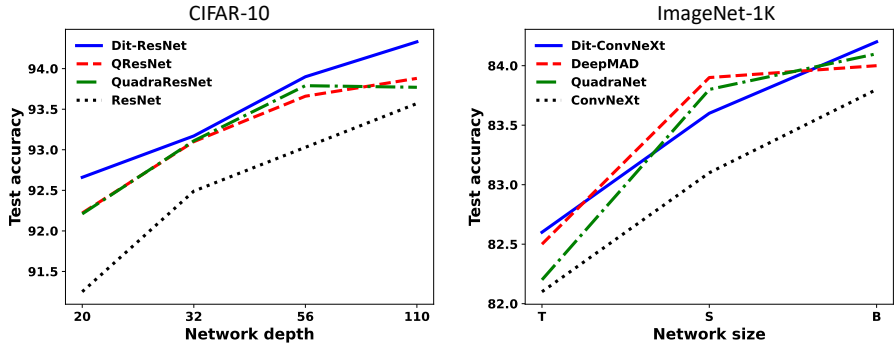


Figure 4: Visualization of some performance results presented in Table 2 (left) and Table 3 (right).

## 4.2 Evaluations on ImageNet-1K

**Dataset and settings.** We extend our investigations to the ImageNet-1K dataset [10], which comprises 1.28 million training images and 50,000 validation images across 1,000 categories. Our analysis primarily focuses on top-1 accuracy on the validation set. Training is conducted at a resolution of  $224 \times 224$ , supplemented by a comprehensive suite of data augmentation and regularization strategies. These include RandAugment [8], mixup [57], cutmix [55], label smoothing [44], layer scale [47], random erasing [58], exponential moving average (EMA) [39], and stochastic depth [17]. These techniques are inspired by the *timm* library [50] and methodologies from Touvron et al. [45]. Detailed descriptions of these hyperparameters are provided in Appendix C.

**Model description.** Mirroring our approach with the CIFAR dataset, we integrate quadratic neurons into a single layer of the ConvNeXt model [34] to conserve computational resources. This adaptation results in the creation of three distinct models: Dit-ConvNeXt-T, Dit-ConvNeXt-S, and Dit-ConvNeXt-B. Specific details about the layer replacement are discussed in the ablation study.

**Results.** Table 3 presents a comparative analysis of Dit-ConvNeXts against state-of-the-art models from the existing literature. Compared to the original ConvNeXt counterparts, our Dit-ConvNeXts exhibit an average increase of 0.5% in top-1 accuracy, with only a marginal average increase of 1% rise

Table 3: Dit-ConvNeXts versus state-of-the-art (SOTA) models on ImageNet-1K. All models listed in the table are trained and validated at a resolution of  $224 \times 224$ .

Arch.	Model	# Param.	FLOPs	Top-1 acc. (%)
Transformers	Swin-T [33]	29M	4.5G	81.3
	DeiT-S [46]	22M	4.6G	79.8
State Space Models	VMamba-T [32]	22M	5.6G	82.2
	VideoMamba-S [27]	26M	4.3G	81.2
CNNs	ResNet-50 [51]	26M	4.1G	80.4
	SLaK-T [31]	30M	5.0G	82.5
	QuadraNet36-T [53]	24M	4.1G	82.2
	DeepMAD-29M [42]	29M	4.5G	82.5
	ConvNeXt-T [34]	29M	4.5G	82.1
	Dit-ConvNeXt-T	29M	5.0G	<b>82.6</b>
Transformers	Swin-S [33]	50M	8.7G	83.0
State Space Models	VMamba-S [32]	44M	11.2G	83.5
	VideoMamba-M [27]	74M	12.7G	82.8
CNNs	ResNet-101 [51]	45M	7.8G	81.5
	ResNet-152 [51]	60M	11.5G	82.0
	SLaK-S [31]	55M	9.8G	83.8
	QuadraNet36-S [53]	50M	8.9G	83.8
	DeepMAD-50M [42]	50M	8.7G	<b>83.9</b>
	ConvNeXt-S [34]	50M	8.7G	83.1
	Dit-ConvNeXt-S	50M	9.2G	83.6
Transformers	Swin-B [33]	88M	15.4G	83.5
	DeiT-B [46]	87M	17.6G	81.8
State Space Models	VMamba-B [32]	75M	18.0G	83.7
CNNs	SLaK-B [31]	95M	17.1G	84.0
	QuadraNet36-B [53]	90M	15.8G	84.1
	DeepMAD-89M [42]	89M	15.4G	84.0
	ConvNeXt-B [34]	89M	15.4G	83.8
	Dit-ConvNeXt-B	90M	16.7G	<b>84.2</b>

in parameter count. Moreover, our models maintain competitive performance against Transformer-based, SSM-based, and CNN-based architectures, highlighting the efficacy and adaptability of Dit-ConvNeXts. Figure 4 further shows that our model consistently improves in accuracy as the model size increases, while other methods tend to saturate, indicating superior scaling property for our model.

### 4.3 Ablation study

#### 4.3.1 Dit-CNNs capture data correlation

The expectation of the quadratic term for Gaussian variables can be expressed as follows:

$$E_{x \sim N(\mu, \Sigma)} [x^T A x] = \mu^T A \mu + \text{tr}(A \Sigma).$$

This equation highlights that the term  $\text{tr}(A \Sigma)$  encapsulates the information derived from data correlation. Consequently, Table 4 examines the impact of this term on the performance of Dit-CNNs in tackling complex tasks. The data presented in Table 4 show a notable decline in accuracy for Dit-CNNs when this term is omitted. This underscores the significance of the quadratic neurons within Dit-CNNs, demonstrating their pivotal role in effectively capturing data correlation, which is essential for task performance.



Table 4: The performance of Dit-CNNs and their counterparts, from which the covariance term  $\text{tr}(A\Sigma)$  and the quadratic term  $x^T Ax$  are omitted in quadratic neurons ( $\Sigma$  is estimated from training samples).

Model	Dataset	Performance (Original)	Performance (minus $\text{tr}(A\Sigma)$ )	Performance (minus $x^T Ax$ )
Dit-ResNet-32	CIFAR-10	93.17%	37.78%	12.11%
Dit-ResNet-32	CIFAR-100	69.68%	22.33%	1.32%
Dit-ConvNeXt-T	ImageNet-1K	82.6%	73.7%	70.6%

### 4.3.2 Incorporate quadratic neurons with minimal computational overhead

To manage computational costs effectively, only three layers within the ConvNeXt architecture have been identified as suitable candidates for the integration of quadratic neurons, as depicted in Figure 5. We explore the most effective layer for the integration of quadratic neurons and elucidate the rationale behind this choice. Figure 5 indicates that substituting traditional neurons with quadratic neurons in the first layer of the block 3 yields the most significant performance improvement. Additionally, a negative correlation is observed between the model’s final accuracy and the accuracy after omitting the quadratic term from Dit-ConvNeXt-T post-training. This correlation underscores the critical influence of quadratic neurons on the model’s efficacy. These findings suggest that Dit-CNNs demonstrating enhanced performance are those where quadratic neurons play a crucial role, highlighting the superior generalization capabilities of quadratic neurons compared to traditional neurons.

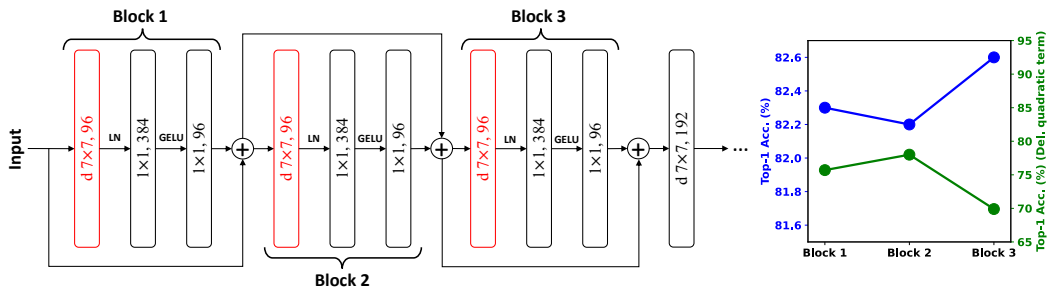


Figure 5: **Left:** Structure of ConvNeXt highlighting three candidate layers (in red) for integrating quadratic neurons. **Right:** ImageNet-1k performance on different Dit-ConvNeXt-T, blue dots indicates top-1 accuracy (left blue vertical axis) while green dots indicates the accuracy post-removal of the quadratic term from Dit-ConvNeXt-T after training (right green vertical axis).

### 4.3.3 Channel/Pixel-wise quadratic neuron utilizations on ImageNet-1K

Table 5: Comparison of quadratic neurons between channel-wise and pixel-wise

Model	Channel/Pixel-wise	Top-1 acc. (%)
ConvNeXt-T	Channel-wise (Our Method)	<b>82.6</b>
	Pixel-wise	82.2
ConvNeXt-S	Channel-wise (Our Method)	<b>83.6</b>
	Pixel-wise	82.9
ConvNeXt-B	Channel-wise (Our Method)	<b>84.2</b>
	Pixel-wise	83.7

As previously discussed, our Dit-CNNs employ quadratic neurons in a channel-wise manner, which provides a clear biological interpretation, as detailed in Section 4. Meanwhile, employing quadratic neurons on a pixel-wise basis, akin to the quadratic convolution outlined in [20], [35], and [60], leads to a scenario where dendritic integration occurs only when neurons receive synaptic inputs from the

same neuron. This setup simplifies interactions between different neurons to a linear approximation, a method that diverges from biological plausibility within the brain. The superior performance achieved through the channel-wise application of quadratic neurons, as evidenced in Table 5, underscores the efficacy of our model. This result supports the hypothesis that models which more closely mirror brain-like mechanisms tend to exhibit enhanced performance.

## 5 Conclusion

In this paper, we first provide a theoretical demonstration of the computational advantages of quadratic neurons in capturing internal correlation within structured data. These neurons model dendritic integration rules observed in biological neurons. Our empirical evaluations using the MNIST and Arabic MNIST datasets for few-shot learning validate our theoretical assertions. Drawing from the biological interpretation of CNNs, we introduce a biologically plausible method to integrate quadratic neurons into CNN architectures, resulting in Dit-CNNs. These Dit-CNNs not only exhibit significant performance enhancements with minimal modifications to their original counterparts but also compete favorably with state-of-the-art models. The potential applicability of our approach to other architectures, such as Deep-MAD [42], hints at further performance improvements. The promising results of this research underscore the vast potential of brain-inspired models. Given that the quadratic integration rule of neurons could be confined to specific brain areas [15, 28], future electrophysiological experiments could reveal other neuronal integration rules. Our findings could guide the development of brain-inspired deep neural networks by incorporating various integration rules corresponding to different brain areas in distinct layers. Additionally, how to theoretically analyze these new brain-inspired models will be an important issue. It is our aim to extend our analysis concerning high-order statistical information to explore the generalization error of these innovative brain-inspired models. We hope that our work will stimulate further investigations into brain-inspired models, ultimately contributing to the quest for artificial general intelligence (AGI).

## Limitations and Discussions

**Theoretical results for quadratic neurons.** Our theoretical framework is established on the assumption that training samples are normally distributed, a simplification that might not fully encapsulate the complexity inherent in real-world datasets. Despite this assumption, our models have demonstrated exceptional performance on various tasks including ImageNet-1K. This underscores the potential of quadratic neurons in capturing correlation within more intricate data distributions. Future efforts will focus on developing a more comprehensive theoretical foundation to better understand the mechanisms through which quadratic neurons achieve this capability.

**Computational cost of quadratic neurons.** While our Dendritic integration inspired Convolutional Neural Networks (Dit-CNNs) strategically limit the increase in the number of learnable parameters by selectively incorporating quadratic neurons in a singular layer, this modification undeniably raises the computational complexity, as evidenced by an increase in Floating Point Operations (FLOPs). Nevertheless, given the substantial enhancements our Dit-CNNs contribute to model performance, it warrants further investigation into optimizing the efficiency of quadratic neuron deployment. For instance, drawing inspiration from the inherent sparsity observed in the quadratic coefficients of biological neurons—specifically, the pronounced quadratic interactions among synaptic inputs on the same dendritic branch [29]—it is conceivable to predefine a sparse configuration for the quadratic coefficients within our Dit-CNNs. This approach could potentially reduce computational demands while maintaining performance gains.

## Acknowledgments

We thank Ruoyu Sun for his helpful discussions and Zhi-Qin John Xu for his comments. This work was supported by Science and Technology Innovation 2030 - Brain Science and Brain-Inspired Intelligence Project with Grant No. 2021ZD0200204; National Natural Science Foundation of China Grant 12271361, 12250710674 (S.L.); National Natural Science Foundation of China with Grant No. 12225109, 12071287 (D.Z.) and the Student Innovation Center at Shanghai Jiao Tong University (S.L., D.Z.).

## References

- [1] Hagai Agmon-Snir, Catherine E Carr, and John Rinzel. The role of dendrites in auditory coincidence detection. *Nature*, 393(6682):268–272, 1998.
- [2] Gal Ariav, Alon Polsky, and Jackie Schiller. Submillisecond precision of the input-output transformation function mediated by fast sodium dendritic spikes in basal dendrites of cal pyramidal neurons. *Journal of Neuroscience*, 23(21):7750–7758, 2003.
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [4] Jie Bu and Anuj Karpatne. Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 675–683. SIAM, 2021.
- [5] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz—a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.
- [6] Spyridon Chavlis and Panayiota Poirazi. Drawing inspiration from biological dendrites to empower artificial neural networks. *Current Opinion in Neurobiology*, 70:1–10, 2021.
- [7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8126–8135, 2021.
- [8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [9] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [12] Fenglei Fan, Wenxiang Cong, and Ge Wang. A new type of neurons for machine learning. *International journal for numerical methods in biomedical engineering*, 34(2):e2920, 2018.
- [13] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.
- [14] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [15] Jiang Hao, Xu-dong Wang, Yang Dan, Mu-ming Poo, and Xiao-hui Zhang. An arithmetic rule for spatial summation of excitatory and inhibitory inputs in pyramidal neurons. *Proceedings of the National Academy of Sciences*, 106(51):21906–21911, 2009.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [17] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.
- [18] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106, 1962.
- [19] Weiwei Jiang. Mnist-mix: a multi-language handwritten digit recognition dataset. *IOP SciNotes*, 1(2):025002, 2020.
- [20] Yiyang Jiang, Fan Yang, Hengliang Zhu, Dian Zhou, and Xuan Zeng. Nonlinear cnn: improving cnns with quadratic convolutions. *Neural Computing and Applications*, 32:8507–8516, 2020.
- [21] Ilenna Simone Jones and Konrad Paul Kording. Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree? *Neural Computation*, 33(6):1554–1571, 2021.
- [22] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.
- [23] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [25] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [26] Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.
- [27] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.
- [28] Songting Li, Nan Liu, Xiao-hui Zhang, Douglas Zhou, and David Cai. Bilinearity in spatiotemporal integration of synaptic inputs. *PLoS computational biology*, 10(12):e1004014, 2014.
- [29] Songting Li, Nan Liu, Xiaohui Zhang, David W McLaughlin, Douglas Zhou, and David Cai. Dendritic computations captured by an effective point neuron model. *Proceedings of the National Academy of Sciences*, 116(30):15244–15252, 2019.
- [30] Gang Liu and Jing Wang. Dendrite net: A white-box module for classification, regression, and system identification. *IEEE Transactions on Cybernetics*, 52(12):13774–13787, 2021.
- [31] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022.
- [32] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.
- [33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

- [35] Pranav Mantini and Shishr K Shah. Cqnn: Convolutional quadratic neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9819–9826. IEEE, 2021.
- [36] Toviah Moldwin, Menachem Kalmenson, and Idan Segev. The gradient clusteron: A model neuron that learns to solve classification tasks via dendritic nonlinearities, structural plasticity, and gradient descent. *PLoS computational biology*, 17(5):e1009015, 2021.
- [37] Varun Ojha and Giuseppe Nicosia. Backpropagation neural tree. *Neural Networks*, 149:66–83, 2022.
- [38] Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, 24(7):1010–1019, 2021.
- [39] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.
- [40] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35:10353–10366, 2022.
- [41] Sanjeevakumar Redlapalli, Madan M Gupta, and K-Y Song. Development of quadratic neural unit with applications to pattern classification. In *Fourth International Symposium on Uncertainty Modeling and Analysis, 2003. ISUMA 2003.*, pages 141–146. IEEE, 2003.
- [42] Xuan Shen, Yaohua Wang, Ming Lin, Yilun Huang, Hao Tang, Xiuyu Sun, and Yanzhi Wang. Deepmad: Mathematical architecture design for deep convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6173, 2023.
- [43] Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206–221, 2008.
- [44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Re-thinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [45] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *Advances in neural information processing systems*, 32, 2019.
- [46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.
- [47] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021.
- [48] Nikolaos Tsapanos, Anastasios Tefas, Nikolaos Nikolaidis, and Ioannis Pitas. Neurons with paraboloid decision boundaries for improved neural network classification performance. *IEEE transactions on neural networks and learning systems*, 30(1):284–294, 2018.
- [49] Balázs B Ujfalussy, Judit K Makara, Máté Lengyel, and Tiago Branco. Global and multiplexed dendritic computations under in vivo-like conditions. *Neuron*, 100(3):579–592, 2018.
- [50] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [51] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.
- [52] Xundong Wu, Xiangwen Liu, Wei Li, and Qing Wu. Improved expressivity through dendritic neural networks. *Advances in neural information processing systems*, 31, 2018.

- [53] Chenhui Xu, Fuxun Yu, Zirui Xu, Chenchen Liu, Jinjun Xiong, and Xiang Chen. Quadranet: Improving high-order neural interaction efficiency with hardware-aware quadratic neural networks. In *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 19–25. IEEE, 2024.
- [54] Zirui Xu, Fuxun Yu, Jinjun Xiong, and Xiang Chen. Quadralib: A performant quadratic neural network library for architecture optimization and design exploration. *Proceedings of Machine Learning and Systems*, 4:503–514, 2022.
- [55] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.
- [56] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.
- [57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- [58] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.
- [59] Lianghai Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.
- [60] Georgios Zoumpourlis, Alexandros Doumanoglou, Nicholas Vretos, and Petros Daras. Non-linear convolution filters for cnn-based learning. In *Proceedings of the IEEE international conference on computer vision*, pages 4761–4769, 2017.

## A Proofs of theorems for binary classification task

Assume that the data points of two classes are equally sampled from two distinct non-degenerate normal distributions:  $class_1 \sim N(\mu_1, \Sigma_1)$ ,  $class_2 \sim N(\mu_2, \Sigma_2)$ , where  $\mu_j \in \mathbb{R}^d$ ,  $\Sigma_j \in \mathbb{R}^{d \times d}$  ( $j = 1, 2$ ). The optimal classifier  $y_{opt}(x) : \mathbb{R}^d \rightarrow \{1, 2\}$  is then defined based on the probability of sampling any point  $x$  from these distributions. Specifically, the classifier can be expressed as:

$$y_{opt}(x) = \arg \max_{j \in \{1, 2\}} p_j(x), \quad (5)$$

where  $p_j(x)$  denotes the probability (density function) of sampling point  $x$  from the distribution  $N(\mu_j, \Sigma_j)$ . The decision rule effectively assigns  $x$  to the class with the highest probability density, reflecting the most likely class membership based on the normal distribution parameters.

If a single quadratic neuron, as described in Equation (1) ( $f(x) = x^T A x + w \cdot x + b$ , where  $A$  is a symmetric matrix), is employed to solve the binary classification task described above, we can prove the following theorems.

### A.1 Existence of the optimal solution

**Theorem 3.** (Existence) *The critical points with respect to the cross-entropy loss  $L(A, w, b)$  are given as follows:*

$$A^* = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w^* = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b^* = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right).$$

i.e.

$$\frac{\partial L}{\partial A} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L}{\partial w} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L}{\partial b} \Big|_{A^*, w^*, b^*} = 0,$$

where

$$L(A, w, b) = \frac{1}{2} \left[ E_{x \sim class_1} \left( \log(1 + e^{f(x)}) \right) + E_{x \sim class_2} \left( \log(1 + e^{-f(x)}) \right) \right].$$

Moreover, the corresponding classifier generated by this formula is the same as the theoretically optimal classifier in Equation (5).

*Proof.* We express the generalized cross-entropy loss  $L(A, w, b)$  as:

$$L(A, w, b) = \frac{1}{2} \int_{\mathbb{R}^d} \left( \log(1 + e^{f(x)}) p_1(x) + \log(1 + e^{-f(x)}) p_2(x) \right) d\mu.$$

The gradients are computed explicitly as follows:

$$\frac{\partial L}{\partial A} = \frac{1}{2} \int_{\mathbb{R}^d} \left( \frac{e^{f(x)}}{1 + e^{f(x)}} p_1(x) - \frac{1}{1 + e^{f(x)}} p_2(x) \right) \frac{\partial f}{\partial A} d\mu,$$

and similarly for  $\frac{\partial L}{\partial w}$  and  $\frac{\partial L}{\partial b}$ . Given the probability density function of the multivariate normal distribution:

$$p_j(x) = (2\pi)^{-d/2} |\Sigma_j|^{-1/2} \exp \left( -\frac{1}{2} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \right) \quad j \in \{1, 2\}.$$

If the parameters  $A, w, b$  are set as:

$$A = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right),$$

one can obtain

$$f(x) = x^T (\Sigma_1^{-1} - \Sigma_2^{-1}) x - 2(\mu_1^T \Sigma_1^{-1} - \mu_2^T \Sigma_2^{-1}) x + \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right)$$

which ensures that

$$e^{f(x)} = \frac{p_2(x)}{p_1(x)}, \quad \forall x \in \mathbb{R}^d.$$

Consequently,

$$\frac{e^{f(x)}}{1 + e^{f(x)}} p_1(x) - \frac{1}{1 + e^{f(x)}} p_2(x) = 0, \forall x \in \mathbb{R}^d \Rightarrow \frac{\partial L}{\partial A} = 0, \frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0.$$

The classifier generated by the quadratic neuron is defined as:

$$y_{model}(x) = \begin{cases} 1 & \text{if } f(x) < 0, \\ 2 & \text{if } f(x) > 0. \end{cases}$$

Therefore,

$$y_{model}(x) = 1 \Leftrightarrow y_{opt}(x) = \arg \max_{j \in \{1,2\}} p_j(x) = 1.$$

Similarly,  $y_{model}(x) = 2$  implies  $y_{opt}(x) = 2$ . Hence,  $y_{model}(x) = y_{opt}(x)$  for all  $x$  in  $\mathbb{R}^d$ , demonstrating the consistency between these two classifiers.  $\square$

## A.2 Uniqueness of the optimal solution

To prove the theorem for uniqueness, we first introduce the following preliminary lemmas:

**Lemma 1.** *Let  $\mathcal{M}(\mathbb{R}^d)$  denote the Lebesgue  $\sigma$ -algebra on  $\mathbb{R}^d$ . Consider  $f(x)$ , a Lebesgue measurable function on  $\mathbb{R}^d$ , satisfying:*

$$\int_{\Omega} f(x) d\mu = 0, \forall \Omega \in \mathcal{M}(\mathbb{R}^d).$$

It follows that  $f(x) = 0$  almost everywhere on  $\mathbb{R}^d$ .

*Proof.* To demonstrate this via contradiction, let us assume, without loss of generality, that:

$$\mu(\{x \mid f(x) > 0\}) > 0.$$

Since

$$0 < \mu(\{x \mid f(x) > 0\}) = \mu\left(\bigcup_{n=1}^{+\infty} \{x \mid f(x) > \frac{1}{n}\}\right) \leq \sum_{n=1}^{+\infty} \mu(\{x \mid f(x) > \frac{1}{n}\}),$$

then we know there exist  $n_0 \in \mathbb{Z}^*$  such that  $\mu(\{x \mid f(x) > \frac{1}{n_0}\}) > 0$ . If we set  $\Omega_0 = \{x \mid f(x) > \frac{1}{n_0}\} \in \mathcal{M}(\mathbb{R}^d)$ , then by the property of  $f(x)$  we can derive:

$$0 = \int_{\Omega_0} f(x) d\mu > \frac{\mu(\{x \mid f(x) > \frac{1}{n_0}\})}{n_0} > 0.$$

Hence, we have reached a contradiction.  $\square$

**Lemma 2.** *Assume  $f(x) = x^T A x + w \cdot x + b$ , a Lebesgue measurable function, equals zero almost everywhere ( $f(x) = 0$  a.e.), with  $A$  being a symmetric matrix. It then follows that  $A = 0$ ,  $w = 0$ , and  $b = 0$ .*

*Proof.* Since  $f(x)$  is continuous, we know  $f(x) = 0$  for  $\forall x \in \mathbb{R}^d$ . Then we have:

$$b = f(0) = 0.$$

Setting  $x = \varepsilon w$  for some  $\varepsilon > 0$ , we obtain:

$$\|w\|_2^2 + \varepsilon w^T A w = 0,$$

letting  $\varepsilon$  go to zero, we have:

$$\|w\|_2^2 = 0 \Rightarrow w = 0.$$

Having established that  $f(x) = x^T A x$ , consider  $x = e_i$ , where  $e_i$  is the unit vector with a value of 1 at the  $i$ -th position and 0s elsewhere. Evaluating the function at this vector, we have  $f(e_i) = e_i^T A e_i = a_{ii} = 0$ . Subsequently, by letting  $x = e_i + e_j$ , where  $e_j$  is another unit vector corresponding to the  $j$ -th position, we obtain the following equation:

$$a_{ii} + a_{ij} + a_{ji} + a_{jj} = 0.$$

Since  $a_{ii} = a_{jj} = 0$  and  $a_{ij} = a_{ji}$  ( $A$  is a symmetric matrix), we know:

$$a_{ij} = 0, \forall i, j \in \{1, 2, \dots, d\} \Rightarrow A = 0.$$

$\square$



**Theorem 4.** (Uniqueness) Consider the conditional cross-entropy loss defined on a set  $\Omega \in \mathcal{M}(\mathbb{R}^d)$ , where  $\mathcal{M}(\mathbb{R}^d)$  denotes the Lebesgue  $\sigma$ -algebra on  $\mathbb{R}^d$ :

$$L(A, w, b \mid \Omega) = \frac{1}{2} \left[ E_{x \sim \text{class}_1, x \in \Omega} \left( \log(1 + e^{f(x)}) \right) + E_{x \sim \text{class}_2, x \in \Omega} \left( \log(1 + e^{-f(x)}) \right) \right],$$

with

$$E_{x \sim \text{class}_j, x \in \Omega} (g(x)) = \int_{\Omega} g(x) p_j(x) d\mu,$$

then the unique solution satisfies

$$\frac{\partial L(A, w, b \mid \Omega)}{\partial A} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b \mid \Omega)}{\partial w} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b \mid \Omega)}{\partial b} \Big|_{A^*, w^*, b^*} = 0$$

for every  $\Omega \in \mathcal{M}(\mathbb{R}^d)$  given by:

$$A^* = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w^* = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b^* = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right).$$

*Proof.* Following the derivation outlined in the proof of Theorem 3, we obtain:

$$\begin{aligned} \frac{\partial L(A, w, b \mid \Omega)}{\partial A} &= \frac{1}{2} \int_{\Omega} \left( \frac{e^{f(x)}}{1 + e^{f(x)}} p_1(x) - \frac{1}{1 + e^{f(x)}} p_2(x) \right) x \cdot x^T d\mu, \\ \frac{\partial L(A, w, b \mid \Omega)}{\partial w} &= \frac{1}{2} \int_{\Omega} \left( \frac{e^{f(x)}}{1 + e^{f(x)}} p_1(x) - \frac{1}{1 + e^{f(x)}} p_2(x) \right) x d\mu, \\ \frac{\partial L(A, w, b \mid \Omega)}{\partial b} &= \frac{1}{2} \int_{\Omega} \left( \frac{e^{f(x)}}{1 + e^{f(x)}} p_1(x) - \frac{1}{1 + e^{f(x)}} p_2(x) \right) d\mu. \end{aligned}$$

Hence, if  $A^*$ ,  $w^*$ , and  $b^*$  satisfy:

$$\frac{\partial L(A, w, b \mid \Omega)}{\partial A} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b \mid \Omega)}{\partial w} \Big|_{A^*, w^*, b^*} = 0, \quad \frac{\partial L(A, w, b \mid \Omega)}{\partial b} \Big|_{A^*, w^*, b^*} = 0$$

for every  $\Omega \in \mathcal{M}(\mathbb{R}^d)$ . Subsequently, invoking Lemma 1 yields:

$$e^{x^T A^* x + (w^*)^T x + b^*} = \frac{p_2(x)}{p_1(x)}, \quad a.e.$$

Simplifying this equation, we have

$$x^T (\Sigma_1^{-1} - \Sigma_2^{-1} - A^*) x - [2(\mu_1^T \Sigma_1^{-1} - \mu_2^T \Sigma_2^{-1}) + (w^*)^T] x + \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right) - b^* = 0,$$

which holds almost everywhere. Applying Lemma 2, we obtain:

$$A^* = \Sigma_1^{-1} - \Sigma_2^{-1}, \quad w^* = -2(\Sigma_1^{-1} \mu_1 - \Sigma_2^{-1} \mu_2), \quad b^* = \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_2^T \Sigma_2^{-1} \mu_2 + \log \left( \frac{|\Sigma_1|}{|\Sigma_2|} \right).$$

□

## B Quadratic neurons capture correlation on multi-class classification task

For a classification task with  $k$  classes, assume the training samples for these classes are equally drawn from  $k$  normal distributions:  $\text{class}_j \sim N(\mu_j, \Sigma_j)$ ,  $j \in [k]$ , the optimal classifier  $y_{opt}(x) : \mathbb{R}^d \rightarrow \{1, 2, \dots, k\}$  is defined by:

$$y_{opt}(x) = \arg \max_{j \in [k]} p_j(x), \quad (6)$$

where  $p_j(x)$  represents the probability (density function) of sampling point  $x$  from the normal distribution  $N(\mu_j, \Sigma_j)$ . Employing  $k$  quadratic neurons for this multi-class classification task allows for the derivation of a theorem analogous to the one discussed in the previous section:

**Theorem 5.** (Existence) The critical points with respect to the cross-entropy loss  $L(\theta)$  are given as follows:

$$A_j^* = \Sigma_j^{-1}, w_j^* = -2\Sigma_j^{-1}\mu_j, b^* = \mu_j^T \Sigma_j^{-1}\mu_j + \log(|\Sigma_j|), j \in [k],$$

where

$$L(\theta) = \frac{1}{k} \sum_{j=1}^k E_{x \sim \text{class}_j} \left[ \log \left( 1 + \sum_{i=1, i \neq j}^k e^{f_i(x) - f_j(x)} \right) \right].$$

Moreover, the corresponding classifier generated by this formula is the same as the theoretically optimal classifier as defined in Equation (6).

*Proof.* The gradients can be computed explicitly as follows:

$$\frac{\partial L}{\partial A_j} = \frac{1}{k} \int_{\mathbb{R}^d} \left( \sum_{i=1, i \neq j}^k \frac{e^{f_j(x)}}{\sum_{i=1}^k e^{f_i(x)}} p_i(x) - \sum_{i=1, i \neq j}^k \frac{e^{f_i(x)}}{\sum_{i=1}^k e^{f_i(x)}} p_j(x) \right) x \cdot x^T d\mu, j \in [k],$$

and similarly for  $\frac{\partial L}{\partial w_j}$  and  $\frac{\partial L}{\partial b_j}$  that can be calculated. From the derivation outlined in the proof of Theorem 3 and the choice for critical points as follows

$$A_j^* = \Sigma_j^{-1}, w_j^* = -2\Sigma_j^{-1}\mu_j, b^* = \mu_j^T \Sigma_j^{-1}\mu_j + \log(|\Sigma_j|), j \in [k],$$

then we have

$$e^{f_j(x) - f_i(x)} = \frac{p_j(x)}{p_i(x)}, \forall x \in \mathbb{R}^d, \forall i, j \in [k],$$

which is equivalent to

$$\frac{e^{f_j(x)}}{\sum_{i=1}^k e^{f_i(x)}} p_i(x) = \frac{e^{f_i(x)}}{\sum_{i=1}^k e^{f_i(x)}} p_j(x), \forall x \in \mathbb{R}^d, \forall i, j \in [k].$$

Thus, we can derive:

$$\frac{\partial L}{\partial A_j} = 0, \frac{\partial L}{\partial w_j} = 0, \frac{\partial L}{\partial b_j} = 0, \forall j \in [k].$$

The classifier generated by quadratic neurons is defined as:

$$y_{\text{model}}(x) = \arg \max_{j \in [k]} f_j(x),$$

Therefore,

$$y_{\text{model}}(x) = j \Leftrightarrow f_j(x) > f_i(x), \forall i \in [k] / \{j\} \Leftrightarrow y_{\text{opt}}(x) = j.$$

Hence,  $y_{\text{model}}(x) = y_{\text{opt}}(x)$  for all  $x$  in  $\mathbb{R}^d$ , demonstrating the consistency between these two classifiers.  $\square$

Theorem 5 demonstrates that in multi-class classification tasks, each neuron captures data correlation by directly relating to the covariance matrix of their respective class. This concept is empirically validated through numerical experiments on the MNIST dataset, as depicted in Figure 6. The eigenvectors associated with the largest eigenvalues of the covariance matrices, representing the principal components of each class’s distribution, reveal the spatial correlation within that class’s data. The marked similarity between these eigenvectors and the eigenvectors of the quadratic integration matrices of neurons confirms that quadratic neurons effectively capture correlation from the data distribution. This intrinsic ability of quadratic neurons to capture correlations provides a possible explanation for the superior generalization capability of quadratic neurons.

## C Experimental settings of ImageNet training

The training settings for our Dit-ConvNeXt models on ImageNet-1K are detailed in Table 6. While all Dit-ConvNeXt variants adhere to a unified configuration, certain parameters—namely, the stochastic depth rate, learning rate, and weight decay—are individually customized for each model variant. These hyperparameters were optimized based on performance outcomes, utilizing a grid search over potential learning rates from the set  $\{0.004, 0.006, 0.008\}$  and weight decay values options within  $\{0.04, 0.06, 0.08\}$ .

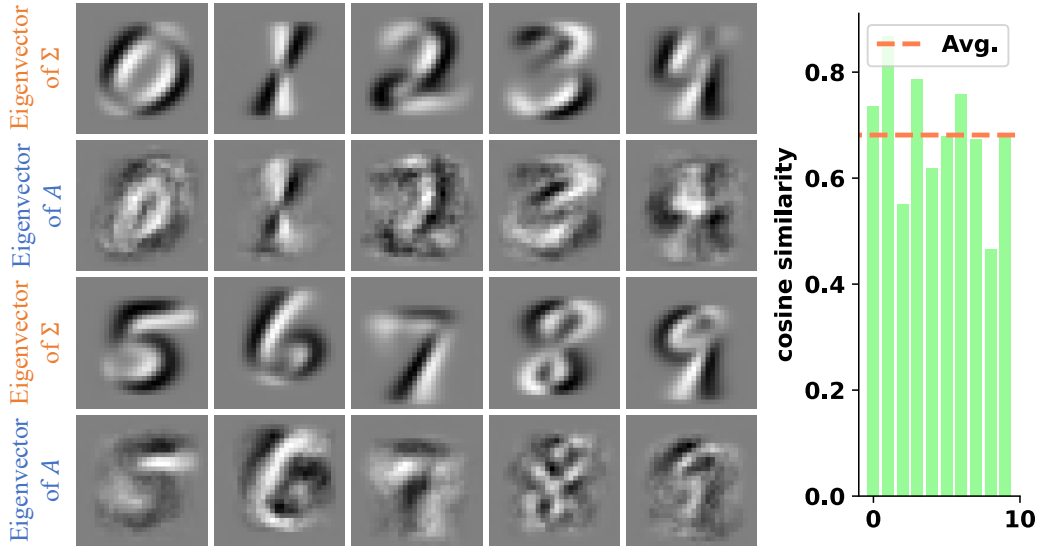


Figure 6: Comparison of Eigenvectors between covariance matrices  $\Sigma_j$  and quadratic weights  $A_j$  for  $j \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . **Left:** Visualization of eigenvectors corresponding to the largest eigenvalue of  $\Sigma_j$  alongside the most similar eigenvectors of  $A_j$ . **Right:** Cosine similarity metrics for ten eigenvector pairs depicted on the left.

Table 6: ImageNet-1K training settings.

Training config	Dit-ConvNeXt-T	Dit-ConvNeXt-S	Dit-ConvNeXt-B
Optimizer	AdamW	AdamW	AdamW
Base learning rate	6e-3	6e-3	4e-3
Weight decay	0.08	0.06	0.06
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.999$
Batch size	4096	4096	4096
Training epochs	300	300	300
Learning rate schedule	cosine	cosine	cosine
Warmup epochs	20	20	20
Warmup schedule	linear	linear	linear
RandAugment [8]	rand-m9-mstd0.5	rand-m9-mstd0.5	rand-m9-mstd0.5
Mixup [57]	0.8	0.8	0.8
Cutmix [55]	1.0	1.0	1.0
Random erasing [58]	0.25	0.25	0.25
Label smoothing [44]	0.1	0.1	0.1
Stochastic depth [17]	0.1	0.4	0.4
Layer scale [47]	1e-6	1e-6	1e-6
Exp. mov. avg. (EMA) [39]	0.9998	0.9998	0.9998

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: See Section 1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: See the Limitations and Discussions part.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: See Section 2 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main text, along with Appendix C, contains comprehensive details necessary for replicating the study's findings, including hyperparameters, data augmentation techniques, etc.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The main text, along with Appendix C, contains comprehensive details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Several experiments, including the comparison of top-1 accuracy on ImageNet-1K, do not feature error bars due to adherence to the convention established in prior studies, which omitted such reporting.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: See the last paragraph in Section 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics [https://neurips.cc/public/EthicsGuidelines?](https://neurips.cc/public/EthicsGuidelines)

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper focuses exclusively on fundamental scientific research and does not address societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: See Reference section.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.



- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

**13. New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

**14. Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.