


Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning

Received: 4 April 2024

Accepted: 13 January 2025

Published online: 22 January 2025

 Check for updatesSpyridon Chavlis  & Panayiota Poirazi  

Artificial neural networks (ANNs) are at the core of most Deep Learning (DL) algorithms that successfully tackle complex problems like image recognition, autonomous driving, and natural language processing. However, unlike biological brains who tackle similar problems in a very efficient manner, DL algorithms require a large number of trainable parameters, making them energy-intensive and prone to overfitting. Here, we show that a new ANN architecture that incorporates the structured connectivity and restricted sampling properties of biological dendrites counteracts these limitations. We find that dendritic ANNs are more robust to overfitting and match or outperform traditional ANNs on several image classification tasks while using significantly fewer trainable parameters. These advantages are likely the result of a different learning strategy, whereby most of the nodes in dendritic ANNs respond to multiple classes, unlike classical ANNs that strive for class-specificity. Our findings suggest that the incorporation of dendritic properties can make learning in ANNs more precise, resilient, and parameter-efficient and shed new light on how biological features can impact the learning strategies of ANNs.

The biological brain is remarkable in its ability to quickly and accurately process, store, and retrieve vast amounts of information while using minimal energy¹. Artificial Intelligence (AI) systems, on the other hand, are notoriously energy hungry^{2–4} and often fail on tasks where biological systems excel, such as continual and transfer learning^{5–7}. The most widely used AI method is deep learning (DL)⁸, which is applied in areas like computer vision⁹ and natural language processing¹⁰ and can even achieve superhuman performance in very specific tasks^{11,12}. However, the number of trainable parameters needed to achieve such performance is large leading to generalization failures due to overfitting¹³, as well as energy consumption levels that are not sustainable¹⁴. Moreover, unlike the brain, DL methods still fail to achieve high-performance accuracy under noisy settings^{15,16} and tasks where information changes in a continuous manner¹⁷. This dichotomy between biological and artificial intelligence systems suggests that drawing inspiration from the brain may help enhance the efficiency of DL models, bringing them one step closer to emulating the biological way of information processing.

DL architectures rely heavily on multilayered artificial neural networks (ANNs) inspired by their biological counterparts. In these networks, artificial nodes are typically constructed as linearly weighted sums of their inputs followed by a nonlinearity, roughly imitating how the soma or axon of biological neurons integrates inputs¹⁸, and learning occurs via changes in the connection strengths (weights) between these nodes¹⁹. In contrast, biological neurons are much more complex, consisting of a soma, an axon, and numerous dendrites that enable them to process thousands of synaptic inputs in parallel, in ways that differ extensively between cell types²⁰. Although the somatic and axonal functionalities of biological neurons are well captured in artificial neurons, the dendritic computations are currently missing.

Biological dendrites, because of their ability to generate local regenerative events (dendritic spikes)^{21,22}, share a similar spiking profile as the neuronal soma. As a result, biological neurons can act as multi-layer ANNs^{23–26}, able to perform complex computations^{27,28}, such as logical operations^{29,30}, signal amplification and segregation^{31,32}, coincidence detection^{33–36}, multiplexing³⁷ and filtering of irrelevant or

noisy stimuli^{38,39}. Consequently, dendrites are thought to underlie complex brain functions, including perception^{40,41}, motor behavior^{42,43}, fear learning^{44–46}, and memory linking⁴⁷. Moreover, dendrites can help achieve such functions in an efficient manner. For example, they enable learning with few plastic synapses⁴⁸, forming memories using small neuronal populations²⁴, and increasing storage capacity^{49,50}. Given the high computational power of dendrites and the associated benefits in biological networks^{27,51}, the current design of artificial neurons seems outdated. Incorporation of dendritic properties would likely empower ANNs^{52–54}, fostering more effective, efficient, and resilient learning behaviors like those seen in biological networks.

The above proposition is supported by recent studies that have integrated dendritic structures and their properties into traditional ANNs^{55–58}, showing promising results on machine learning (ML) tasks^{59–64}. For instance, adding active dendrites in ANNs was shown to enhance the network's ability to learn continually⁶³, while including a specific dendritic nonlinearity improved performance in a multitask learning scenario⁶⁵. However, to achieve improved performance, these studies have either sacrificed biological plausibility⁶⁴, used a very large number of trainable parameters⁶³, or were applied to very simple tasks⁵⁸.

Here, we propose a bio-realistic dendritic architecture that aims to improve learning in ANNs trained with the backpropagation algorithm. In the proposed architecture, inputs are partially sampled and fed into the dendritic layer, which is, in turn, connected to the somatic layer in a sparse and highly structured manner (Fig. 1). Input sampling is inspired by the receptive fields of neurons in the visual cortex^{66,67} and amounts to sampling a restricted part of the input, similar to a specific form of convolutional networks, the so-called locally-connected networks⁶⁸, as opposed to the entire image that is typically done in ANNs. By incorporating dendritic structural and sampling features, the proposed dendritic ANN models match or outperform traditional ANNs on several image classification tasks and counteract overfitting, while using orders of magnitude fewer trainable parameters. These improvements are likely due to a more extensive utilization of trainable weights and a different learning strategy used by dendritic versus traditional ANNs. Overall, our findings suggest that dendrites can augment the computational efficiency of ANNs without sacrificing their performance accuracy, opening new avenues for developing bio-inspired ML systems that inherit some of the major advantages of biological brains.

Results

To explore the role of dendritic properties in efficient learning, we developed a dendritic ANN (dANN) model with structured connectivity that loosely mimics the morphology and sampling characteristics of biological neurons (Fig. 1a). In this model, each dendrite acts as a typical point neuron: it linearly sums its weighted inputs (synapses) and passes the sum through a nonlinearity. The dendritic activations are subsequently multiplied by the cable weights and summed at the soma before going through a second nonlinearity (Fig. 1b). To train the model using ML platforms (e.g., TensorFlow, PyTorch, Jax), we implemented it as a traditional ANN with two sparsely connected hidden layers, representing the dendritic and somatic units, respectively, and a fully connected output layer (Fig. 1c). For comparison purposes, we also implemented a fully connected, vanilla ANN (vANN) with the same number of layers (Fig. 1d).

In addition to their structured connectivity, the dendrites of biological neurons receive only partial information from a visual scene⁶⁹. To investigate the contribution of this property, we implemented three types of input sampling for dendritic ANN models: a) random sampling of input features (R), b) local receptive fields (LRF) where each dendrite samples from a spatially restricted part of the image, and c) global receptive fields (GRF) where all dendrites belonging to a soma sample from the same spatially restricted part of the image. Finally, for completeness, we also explored an all-to-all, fully connected type of

sampling (F), which is also used by the vANN (Fig. 1e). As this type of sampling is not biologically realistic, we term the respective model a *partly-dendritic ANN* (pdANN). We then tested the learning capabilities of our models on various image classification tasks (Supplementary Fig. 1) using the same (default) hyperparameters, optimization algorithm, and loss function (see Methods). To ensure fair comparisons, we tested equivalent network architectures for all models, i.e., consisting of the same number of nodes in each hidden layer. It should be noted that the conjunction of structured connectivity and sparse input sampling under the term *dendritic ANNs* is meant to describe an architectural analogy between the artificial dendritic neurons presented here and real biological neurons. It is not meant to represent a direct biological equivalence or the mechanisms behind this architectural analogy.

Bio-inspired dendritic ANNs are more accurate, robust, and efficient than vanilla ANNs on image classification

We first tested the learning capabilities of all models against the Fashion MNIST (FMNIST) dataset (Fig. 2a). We found that the three dendritic ANNs achieve better learning and combat overfitting much more effectively than vANN models for both size-matched and larger vANN architectures (in terms of trainable parameters). This is evidenced by a consistently lower test loss for all dANN models compared to vANNs of the same number of trainable parameters. The pdANN, on the other hand, requires a larger number of parameters than the vANN to match its minimum loss (Fig. 2b). Importantly, the vANN models exhibit overfitting as the model size increases (Fig. 2b and Supplementary Fig. 2), while this does not occur for dANNs or the pdANN (for the model sizes tested), suggesting that the structured connectivity of dendrites may serve as a natural regularizer⁷⁰. Indeed, reductions in overfitting are also evident in vANNs when using various regularization methods or hyperparameter tuning such as dropout (Supplementary Fig. 3), different learning rates (Supplementary Fig. 4) or an early stopping criterion (Supplementary Fig. 5). These overfitting reductions, however, are not as large as those seen in dANNs, suggesting that the bio-inspired dendritic structures that were tested here may serve as more effective regularizers. While these findings do not constitute a comparison to all traditional regularization methods, they suggest that the observed regularization is an emerging property of the bio-inspired architecture.

In addition to overfitting benefits, dANNs match the best performance of vANNs via the use of much fewer trainable parameters (Fig. 2c), suggesting that dendritic features render ANNs more efficient. Among the three dANN configurations tested, the dANN with local receptive fields (dANN-LRFs) is the most efficient: it reaches maximum accuracy and minimum loss with over one order of magnitude fewer trainable parameters than the vANN. The dANN models with random (dANN-R) and global receptive field (dANN-GRF) sampling are slightly worse but significantly more efficient than the vANN. Finally, the pdANN shows reduced overfitting but no efficiency gains with respect to accuracy. Overall, these findings suggest that both features, i.e., the structured dendritic connectivity and the restricted input sampling, contribute to the efficiency gains of dANN models.

Finally, we found that, as expected, learning in dANN and pdANN models improves with network size (lower loss: Fig. 2d, better accuracy: Fig. 2e). More importantly, unlike other bio-inspired architectures⁷¹, dANNs (with LRFs and GRFs) and pdANN models exhibit improved performance and/or stability as the number of layers increases (Supplementary Fig. 6), revealing their potential for use in deeper architectures.

To substantiate our results on the beneficial role of dendritic features, we tested the dANN models on five additional benchmark datasets (Fig. 3, Table 1, see Methods). As with FMNIST, we found that the best dANN models consistently matched (or outperformed, e.g.,

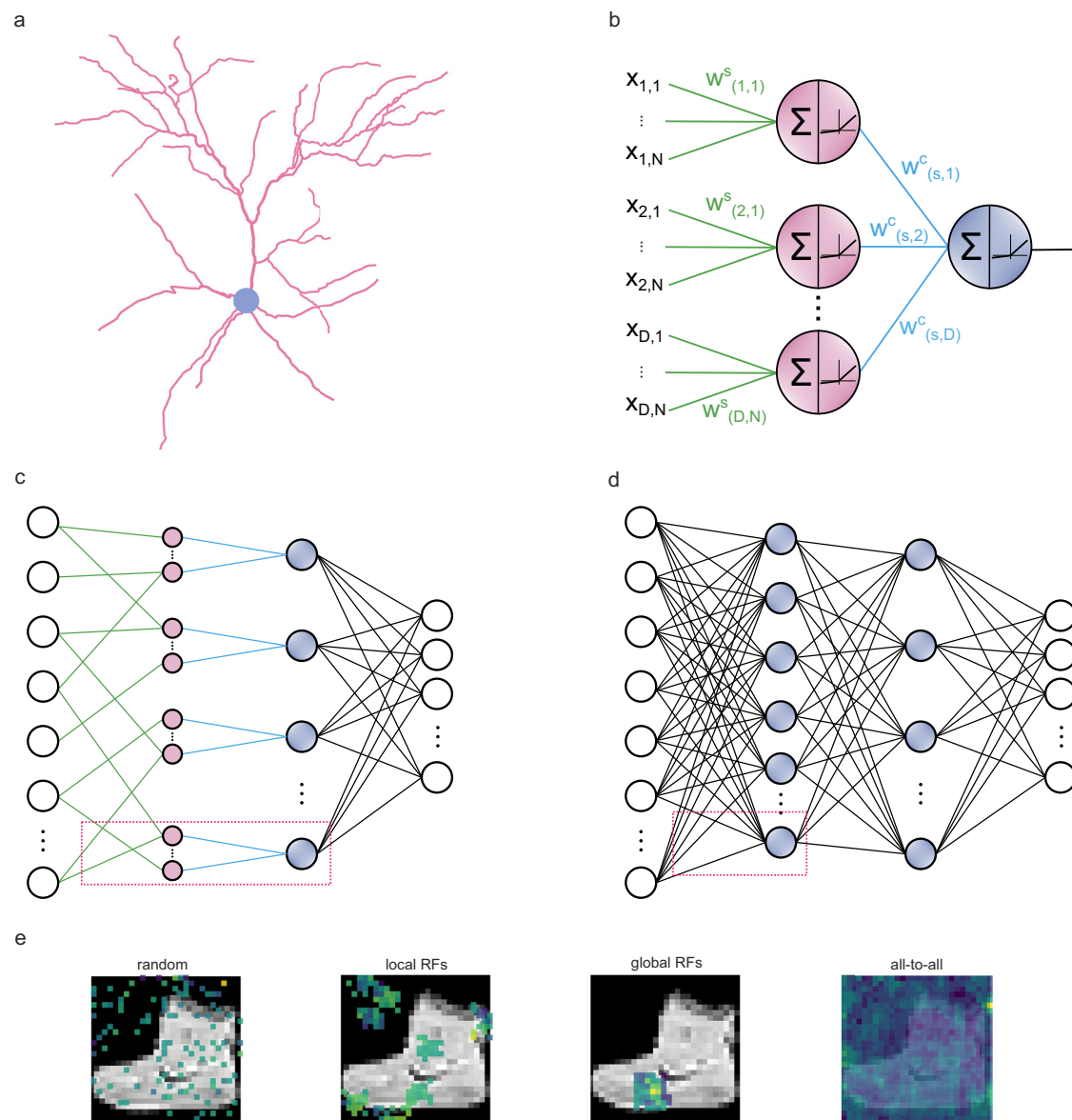


Fig. 1 | Schematic representation of the dendritic ANN (dANN) compared to a classical vanilla ANN (vANN). **a** Example of a layer 2/3 pyramidal cell of the mouse primary visual cortex (dendrites: pink; soma: grey) that served as inspiration for the artificial dendritic neuron in **b**. The morphology was adopted from Park et al.¹²⁷. **b** The dendritic neuron model consists of a somatic node (blue) connected to several dendritic nodes (pink). All nodes have a nonlinear activation function. Each dendrite is connected to the soma with a (cable) weight, $w_{(d,s)}^c$, where d and s denote the dendrite and soma indices, respectively. Inputs are connected to dendrites with (synaptic) weights, $w_{(d,n)}^s$, where d and n are indices of the dendrites and input nodes, respectively. $d \in \{1, D\}$, $n \in \{1, N\}$, N denotes the number of synapses each dendrite receives, and D the number of dendrites per soma s . **c** The dendritic ANN

architecture. The input is fed to the dendritic layer (pink nodes), passes a non-linearity, and then reaches the soma (blue nodes), passing through another non-linearity. Dendrites are connected solely to a single soma, creating a sparsely connected network. **d** Typical fully connected ANN with two hidden layers. Nodes are point neurons (blue) consisting only of a soma. **e** Illustration of the different strategies (LRF) to sample the input space: random sampling (R), local receptive fields (LRF), global receptive fields (GRF), and fully connected (F) sampling of input features. Examples correspond to the synaptic weights of all nodes that are connected to the first unit in the second layer. The colormap denotes the magnitude of each weight. The image used in the background is from the Fashion MNIST (FMNIST) dataset (see Methods).

on CIFAR10) the best vANN in terms of both accuracy and loss (Table 1 and Supplementary Table 1). Moreover, similarly to FMNIST, we found that dANNs are much more efficient than vANNs for all datasets. Specifically, they can match the accuracy (Fig. 3a) and loss (Fig. 3b) of the best vANN using 1–3 orders of magnitude fewer trainable parameters. It is worth noting that for more difficult tasks, like the CIFAR10 dataset, the difference in both the number of trainable parameters (Fig. 3a, b) and the best accuracy (Table 1) between these dANNs and vANNs is more prominent. Finally, the pdANN achieves slightly higher accuracy than vANNs on two datasets (Table 1, EMNIST and CIFAR10), albeit with a much larger number of trainable parameters compared to dANNs

—and often vANNs— (Fig. 3a, b, purple bars), thus diminishing any efficiency gains.

To quantify the efficiency differences between dendritic, partly-dendritic and vANN models we formulated two efficiency scores: a) the *accuracy efficiency score*, which is calculated by normalizing the highest test accuracy attained by a model against the product of its trainable parameters and the number of epochs required to achieve the lowest validation loss and b) the *loss efficiency score*, where the minimum loss attained by a model is similarly normalized using the product of its trainable parameters and the epochs required to achieve the lowest validation loss (see Methods). These metrics

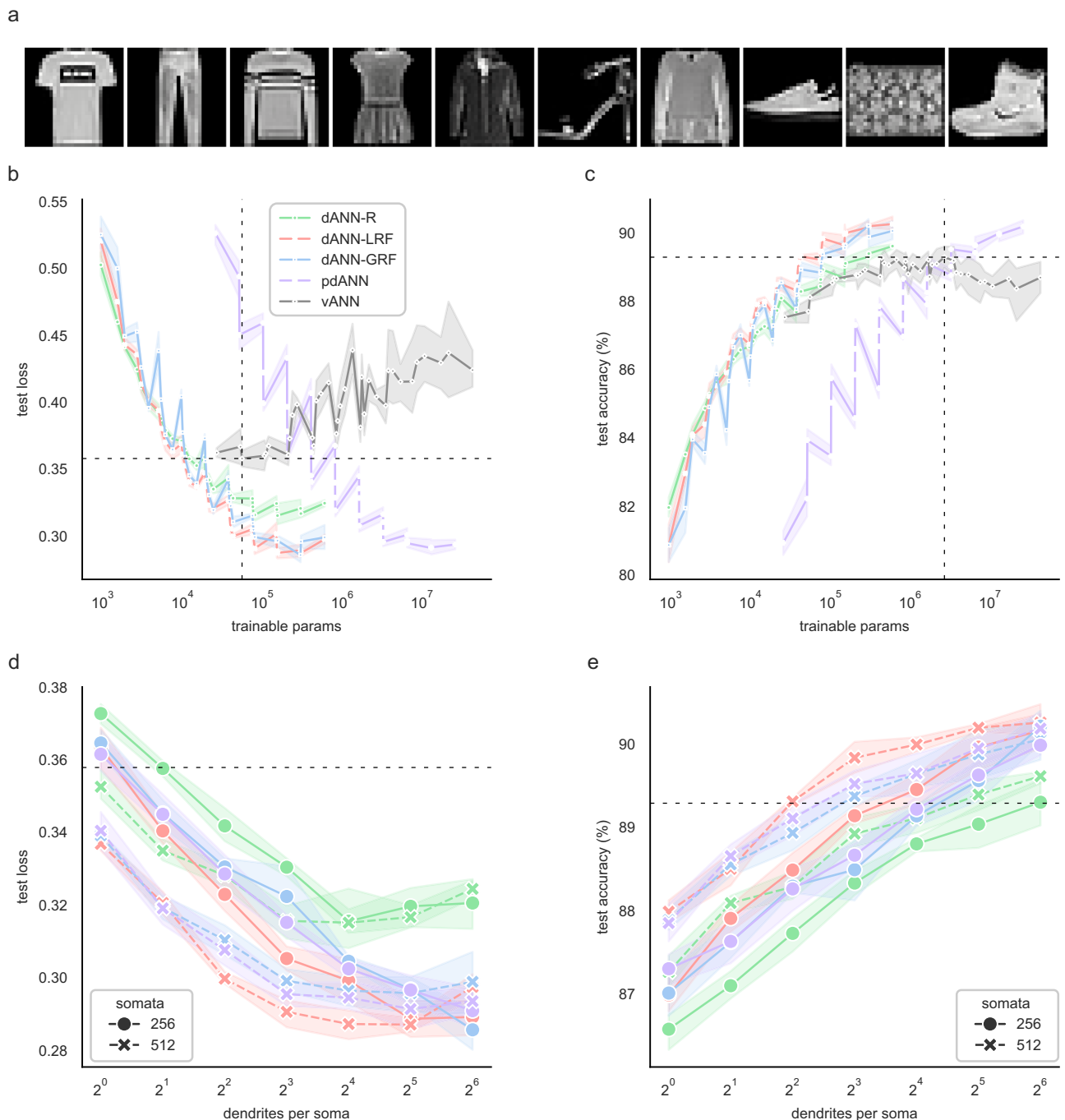


Fig. 2 | Dendritic features improve learning on Fashion MNIST classification.

a The Fashion MNIST dataset consists of 28×28 grayscale images of 10 categories. **b** Average test loss as a function of the trainable parameters of the five models used: A dendritic ANN with random inputs (dANN-R, green), a dANN with LRFs (red), a dANN with GRFs (blue), a partly-dendritic ANN with all-to-all inputs (pdANN, purple), and the vANN with all-to-all inputs (grey). Horizontal and vertical dashed lines denote the minimum test loss of the vANN and its trainable parameters, respectively. The x-axis is shown in a logarithmic scale (log10). **c** Similar to **b**, but depicting the test accuracy instead of the loss. **d** Test loss as a function of the number of dendrites per somatic node for the three dANNs and the pdANN model. The

linestyle (solid and dashed) represents different somatic numbers. The dashed horizontal line represents the minimum test loss of the vANN (512-256 size of its hidden layers, respectively). The x-axis is shown in a logarithmic scale (log2). **e** Similar to **d**, but showing the test accuracy instead of the loss. The dashed horizontal line represents the maximum test accuracy of the vANN (2048-512 size of its hidden layers, respectively). Note that while all models have the same internal connectivity structure, the pdANN model (purple) has a much larger number of trainable parameters due to its all-to-all input sampling. For all panels, shades represent the 95% confidence interval across $N = 5$ initializations for each model.

consider both the size of a network (trainable parameters) and a proxy of its convergence speed (training epochs). We found that dANN models exhibit higher efficiency than both vANNs and pdANNs across all datasets tested (Fig. 3c). Of note, the pdANN, which is a hybrid between dANNs and vANNs, is substantially less efficient than

the vANNs in terms of accuracy (Fig. 3c) but more efficient in terms of loss (Fig. 3d). This is because the pdANN model has a very large number of parameters and thus needs a larger number of epochs than vANNs for training (Supplementary Fig. 7). Yet, it is less susceptible to overfitting than vANNs (Fig. 2b and Supplementary Fig. 2),

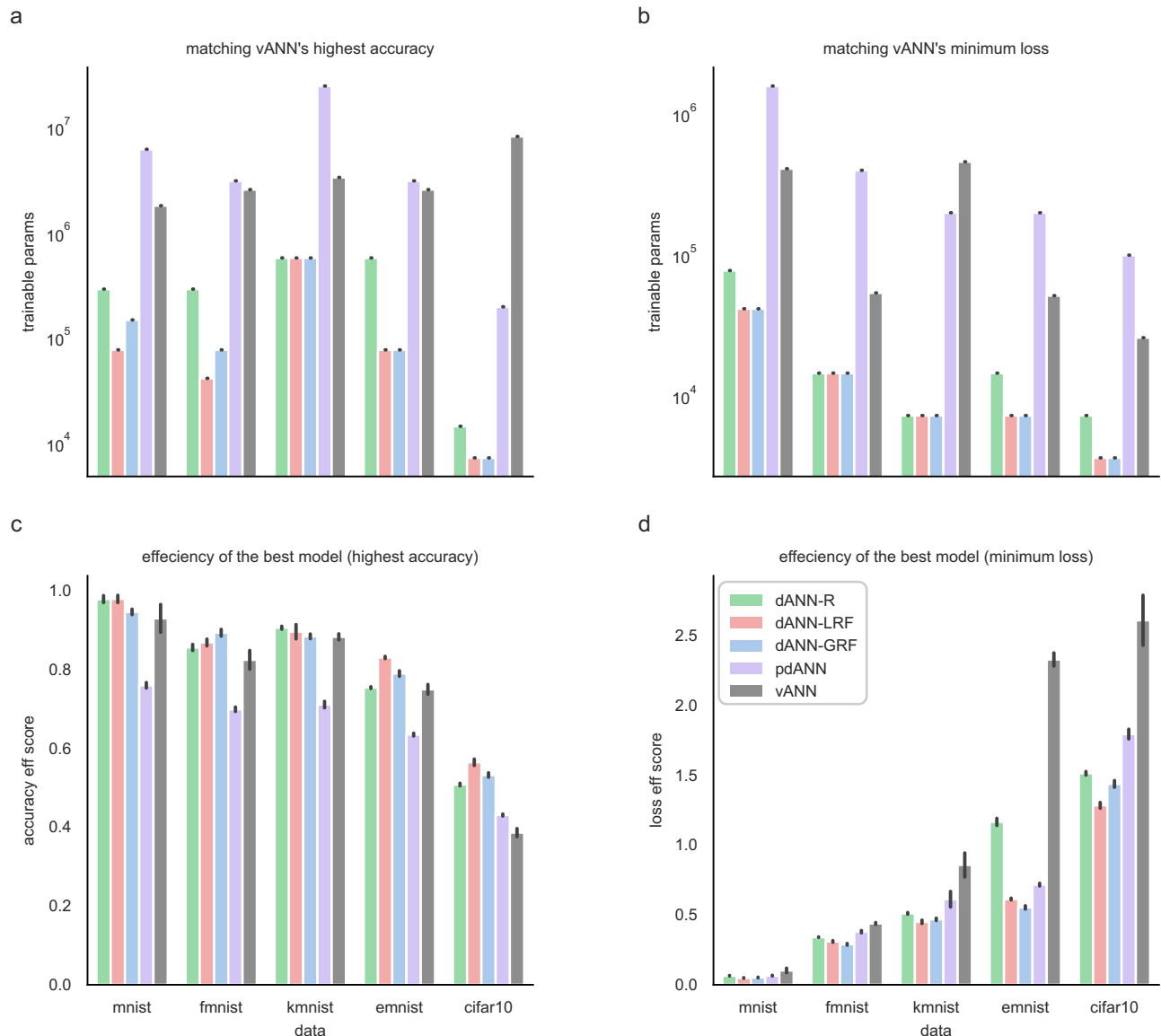


Fig. 3 | Dendrites improve efficiency across various benchmark datasets. The comparison is made between the three dendritic models, dANN-R (green), dANN-LRF (red), dANN-GRF (blue), the partly-dendritic model pdANN (purple) and the vANN (grey). **a** Number of trainable parameters that each model needs to match the highest test accuracy of the vANN. **b** The same as in **a**, but showing the number of trainable parameters required to match the minimum test loss of the vANN. **c** Accuracy efficiency scores of all models across the five datasets tested. This score reports the best test accuracy achieved by a model, normalized with the logarithm

of the product of trainable parameters with the number of epochs needed to achieve minimum validation loss. The score is bounded in $[0, 1]$. **d** Same as in **c**, but showing the loss efficiency score. Here the minimum loss achieved by a model is normalized with the logarithm of the trainable parameters times the number of epochs needed to achieve minimum validation loss. The score is bounded in $[0, \infty)$. In all barplots the error bars represent one standard deviation across $N = 5$ initializations for each model.

thus achieving a smaller loss for similar accuracy in the majority of the datasets tested.

The above findings illustrate that the two dendritic features implemented here, i.e., the structured connectivity and the restricted input sampling, provide important efficiency gains in image classification compared to classical vANNs. To further dissect the relative contributions of these two features, we also compared the dendritic and partly-dendritic ANN models to: **a**) densely connected vANN models (Fig. 4 and Supplementary Fig. 8) and **b**) sparsely connected sANN models (Supplementary Figs. 9 and 10), furnished with the four types of input sampling. We found that in densely connected ANNs, restricted input sampling substantially improves efficiency, as fewer trainable parameters are required to match the accuracy and loss of the best-performing vANN (Fig. 4a, b). However, these improvements

are not as high as those seen in the dendritic ANNs. Both the accuracy and loss efficiency scores of dANNs are better than the respective dense ANNs with restricted input sampling (Fig. 4c, d). As already shown in Fig. 3, the pdANN does not exhibit a consistent behavior with respect to efficiency gains: it shows reduced accuracy (Fig. 4c, purple) but improved loss (Fig. 4d, purple) in its efficiency scores compared to vANNs. Overall, these results suggest that while restricted input sampling can offer some efficiency gains when incorporated in classical ANNs, it cannot fully account for the improved efficiency of the dendritic ANN models.

To assess the contribution of structured dendritic connectivity, we next compared dANNs to randomly connected, sparse ANN models (sANNs). Sparse neural networks were previously shown to exhibit improved performance⁷² and since dANNs are a specific subset of

Table 1 | The table shows the maximum performance achieved by each type of model across the five datasets tested

Models	Model performance—top accuracy				
	MNIST	FMNIST	KMNIST	EMNIST	CIFAR10
Test accuracy (%)					
dANN-R	98.090 ± 0.0583	89.612 ± 0.0870	91.076 ± 0.0811	82.745 ± 0.2568	52.458 ± 0.4347
dANN-LRF	98.466 ± 0.1058	90.256 ± 0.2237^a	91.928 ± 0.0643^a	83.166 ± 0.0893	56.966 ± 0.9796
dANN-GRF	98.576 ± 0.0809^a	90.182 ± 0.2164	90.046 ± 0.2915	83.779 ± 0.2064	56.998 ± 0.4875
pdANN	98.466 ± 0.0862	90.184 ± 0.2203	90.61 ± 0.6927	84.121 ± 0.1880^a	60.12 ± 0.3906^a
vANN	98.034 ± 0.2742	89.288 ± 0.3654	91.552 ± 0.6629	83.381 ± 0.2681	49.082 ± 1.2092
Test loss					
dANN-R	0.0644 ± 0.0013	0.3245 ± 0.0028	0.5068 ± 0.0045	1.0591 ± 0.0225	1.4612 ± 0.0172
dANN-LRF	0.0483 ± 0.0018	0.2975 ± 0.0042	0.4374 ± 0.0037^a	0.6105 ± 0.0048	1.2684 ± 0.0230^a
dANN-GRF	0.0471 ± 0.0022^a	0.2947 ± 0.0052	0.5757 ± 0.0335	0.5605 ± 0.0077	1.3398 ± 0.0217
pdANN	0.0503 ± 0.0023	0.2935 ± 0.0043^a	0.4789 ± 0.0378	0.5390 ± 0.0060^a	1.2811 ± 0.0247
vANN	0.0967 ± 0.0116	0.4040 ± 0.0066	0.8246 ± 0.0791	2.0860 ± 0.0476	2.0455 ± 0.1375

Reported scores correspond to the models that achieved the best test accuracy. The corresponding test loss values for these models are also depicted. Performance accuracy and loss are listed as mean ± standard deviation over $N = 5$ initializations for each model.

^aBold fonts denote the top metric per column.

sANNs, it is likely that efficiency gains stem primarily from internal sparsity. We found that the *bio-inspired dANNs*, namely those with structured restricted input sampling (LRF and GRF), consistently outperform sANNs in terms of efficiency gains. When comparing dANNs and sANNs with matched input sampling types—whereby the only difference is the structured vs. random internal connectivity—differences in efficiency gains are reduced, especially for larger models (Supplementary Fig. 9). However, most of the dANNs stills exhibit higher efficiency gains than respective sANNs (Supplementary Figs. 9 and 10) suggesting that a structured, tree-like internal sparsity provides additional efficiency benefits. The models with fully connected input sampling (pdANN and psANN) are very similar.

Overall, these experiments confirm that both dendritic features, namely the structured connectivity and the restricted input sampling, contribute significantly to improving the performance accuracy of dANNs, their efficiency and their ability to reduce overfitting. Given these findings, in the following sections, we focus our analysis on the three dANN models as compared to fully connected vANN models.

Dendritic ANNs employ a distinct learning strategy and fully exploit their available resources

To better understand why dANN models outperform vANNs, we analyzed their weight distributions post-learning of the Fashion MNIST dataset. We found a broader distribution, i.e., larger range of values of synaptic (layer 1) weights for dANNs compared to the vANN (Supplementary Table 2 and Fig. 5a, top row) and a bimodal distribution of dANN cable (layer 2) weights, all centered around zero (Supplementary Table 2 and Fig. 5a, middle row). For the dANN-LRF model, in particular, there were very few cable weights close to zero, indicating that the model effectively utilizes all trainable parameters of this layer. In contrast, in the vANN model, weights follow a Gaussian-like distribution centered around zero, suggesting that many weights are not as effectively utilized. Finally, the distribution of the output layer weights in dANNs is broader than in the vANN model (Supplementary Table 2 and Fig. 5a, bottom row). These observations suggest that dANNs fully exploit their trainable parameters, especially their cable (second layer or dendrosomatic) weights, compared to the second hidden layer of the vANN.

To delineate how the nodes of the different models contribute to a decision, we looked into their selectivity. First, we calculated the information entropy, which measures how class-specific a node is. High entropy values indicate mixed selectivity, whereby the node is

active for more than one class, while low values indicate class specificity. We found opposite entropy distributions between the dANNs and the vANN. This means that the dANN models primarily have mixed-selective nodes in both hidden layers, while vANNs primarily have class-specific nodes. This difference was even more pronounced for dANNs with global or local RFs (Fig. 5b).

To assess whether the observed differences in entropy map onto node specificity, we formulated the selectivity index, which counts how many classes a given node responds to. Specifically, if a node is active (activation greater than zero for a given image) for more than 400 images of a specific class, corresponding roughly to 40% of testing images, its selectivity index for that class is set to one. As with entropy distributions, we found that in dANNs, both layers consist primarily of mixed-selective nodes, while the vANN contains primarily class-specific nodes (Fig. 5c). These observations suggest that dANN and vANN models employ different strategies to solve the same classification task.

To complete our interpretability analysis, we visualized the hidden representations of the compared dANN and vANN models post-learning. The goal was to assess the amount of high-level information that is extracted by the first and second hidden layers across models (i.e., dendritic and somatic layers for dANNs, respectively). We applied the T-distributed stochastic neighbor embedding (TSNE), an algorithm that reduces the dimensionality and allows visualization of high-dimensional data⁷³. By visual inspection, we observed a change in the representation between the dendritic and somatic layers of dANN models, similar to representations of vANN between its two hidden layers (Fig. 6a–d). We quantified the separability of the representations using the silhouette and the neighborhood (nh) scores, which measure the global and local degree of separability, respectively (see Methods). In the three dANN models, global and local separability was increased from the dendritic to the somatic layer (Fig. 6e, f), something that we also observed in the hidden layers of the vANN. This means that the discriminatory power of both dANNs and vANN increases across layers in a similar way. This is in line with the findings of Fig. 5, whereby the vANN is shown to have higher class-specificity than the dANNs in the first layer, and thus higher separability scores. Importantly, our results regarding the properties of the representations in low-dimensional space reflect the properties of the high-dimensional data as shown by their high trustworthiness scores (Fig. 6g). The latter measures the extent to which the local structure of the data is retained after projection to the lower-dimensional space. Values close to 1 indicate high

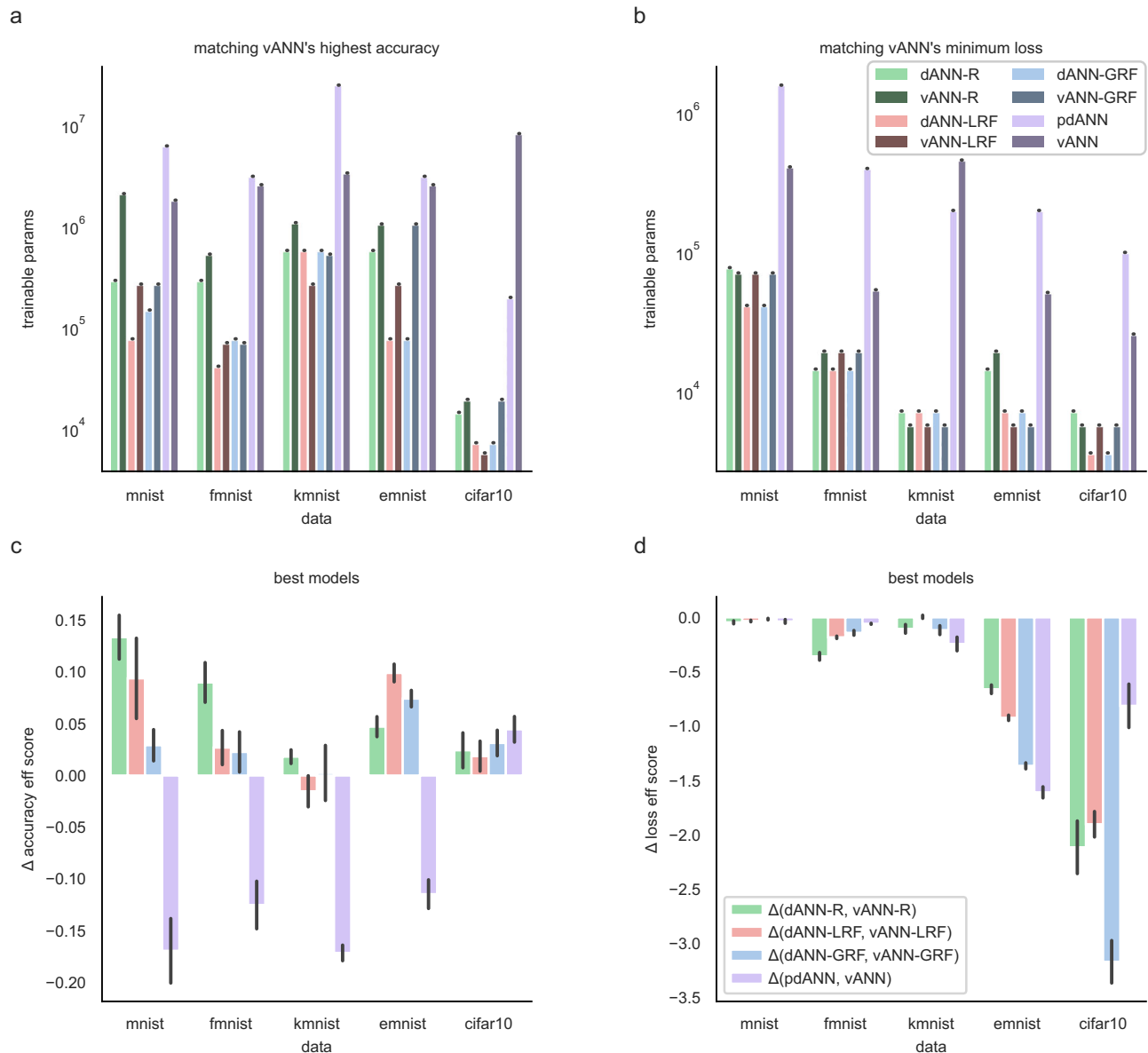


Fig. 4 | Comparison of dendritic, partly-dendritic, and vanilla ANNs with different types of input sampling. The following models were compared: dANN-R and vANN-R with random input sampling (light and dark green), dANN-LRF and vANN-LRF with local receptive field sampling (light and dark red), dANN-GRF and vANN-GRF with global receptive field sampling (light and dark blue), and pdANN and vANN with all-to-all sampling (light and dark purple). **a** Number of trainable parameters that each model needs to match the highest test accuracy of the respective vANN. **b** The same as in **a**, but showing the number of trainable parameters required to match the minimum test loss of the vANN. **c** Difference (Δ) in

accuracy efficiency score between the structured (dANN/pdANN) and vANN models. Test accuracy is normalized with the logarithm of trainable parameters times the number of epochs needed to achieve minimum validation loss. The score is bounded in $[0, 1]$. **d** Same as in **c**, but showing the difference (Δ) of the loss efficiency score. Again, we normalized the test score with the logarithm of the trainable parameters times the number of epochs needed to achieve minimum validation loss. The score is bounded in $[0, \infty)$. In all barplots the error bars represent one standard deviation across $N = 5$ initializations for each model.

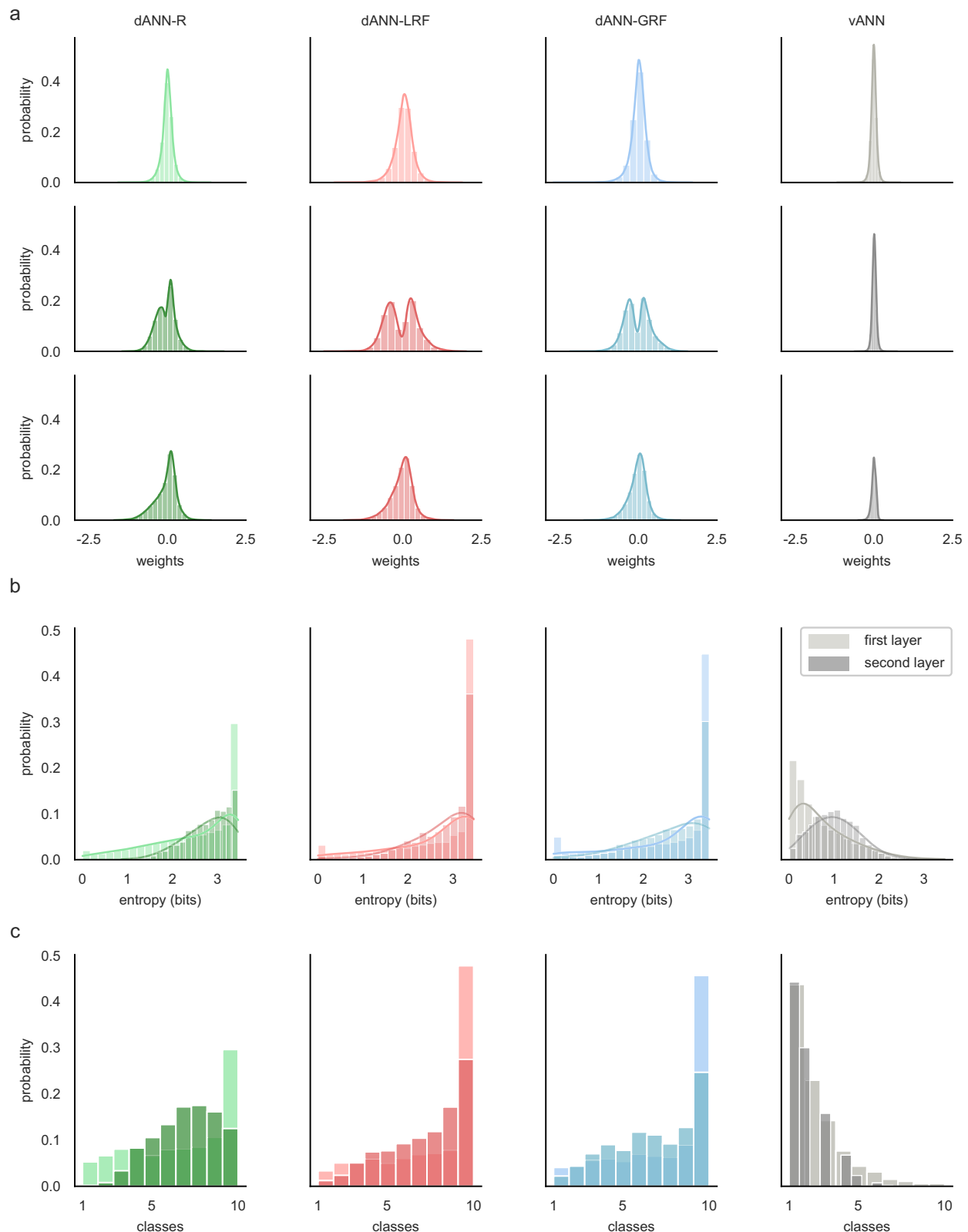
reliability. Figure 6g suggests that the three dANNs do a better job in retaining the structure of the original data in their representations, as measured by TSNE, compared to vANNs. This is probably a result of the different strategies implemented by these networks.

Overall, our interpretability analysis reveals that dANNs use a different strategy than the vANN model to achieve accurate, robust, and efficient image classification: rather than becoming class-specific early on like the vANN, the dANN models exhibit mixed-selectivity in both layers. This strategy may underlie their ability to create trustworthy representations of the input data, and achieve high performance accuracy and reduced overfitting while using significantly fewer yet fully utilized trainable parameters.

Dendritic benefits are more pronounced as the task difficulty increases

Our image discrimination results suggest that the difference between dANN and vANN models may be larger for more difficult tasks/datasets (see results for CIFAR10 in Table 1). To test this hypothesis, we constructed learning scenarios that are known to be challenging for traditional ANN models.

First, we added Gaussian noise (with a variable standard deviation σ and zero mean) to all images in the FMNIST dataset, thus creating new datasets of increasing classification difficulty (Fig. 7a). We then selected the best vANN and the corresponding dANNs that matched its performance accuracy on FMNIST (from Figs. 2 and 3) and



tested their performance on the noisy datasets. We found that, while the performance of all models declined with increasing noise levels, dANNs demonstrated higher efficiency and resilience. This is evident by a slower increase rate for the loss and a slower drop rate for the accuracy efficiency scores, respectively, compared to vANNs (Fig. 7b). In all cases, the best-performing dANN was the one with local RFs (dANN-LRF).

To confirm the advantages of dANN models on challenging tasks, we constructed a second learning scenario that remains challenging for traditional ANNs. In this task, models were fed with batches of inputs belonging to the same class in a sequential manner (Fig. 7c). This process, which was repeated 50 times (epochs), resulted in models receiving information only from images of a single class during gradient calculation. As with the noisy task, the three dANN models

Fig. 5 | Dendritic ANN models fully exploit their available resources and solve the task using a different learning strategy. a Weight probability density functions after training for dANN-R, dANN-GRF, dANN-LRF, and vANN. The density functions are built by concatenating all weights across $N = 5$ initializations for each model. First hidden layer (top row), second hidden layer (middle row), and output layer (bottom row) weights are shown. Both x and y axes are shared across all subplots for visual comparison among the density plots. Supplementary Table 2 contains the kurtosis, skewness, and range of all KDE plots. **b** Probability density

function of the entropy (bits) for the first (normal color) and second (shaded color) hidden layer, respectively. Entropies are calculated using the activations of each layer for all test images of FMNIST (see Methods). Silent nodes have been excluded from the visualization. Higher values signify mixed selectivity, whereas low values indicate class specificity. **c** Probability density functions of selectivity for both layers (different color shades) and all models (columns). For all histograms, the bins are equal to the number of classes, i.e., for the FMNIST dataset.

were more accurate (Table 2), less variable across different initializations, and much more efficient than the vANN, as evidenced by their loss and accuracy efficiency scores (Fig. 7d). The best-performing dANN was again the one with local RFs (dANN-LRF). Overall, these findings suggest that incorporating dendritic features in ANNs may result in even greater robustness, accuracy, and efficiency gains when the task difficulty is increased.

Discussion

Bio-inspired machine learning is one of the most dynamic branches of AI. Biological dendrites and their learning rules are among the top candidates being explored, already showing highly promising results in artificial neural networks^{55,56,74,75}. Recent studies in dendritic networks focus on their potential to tackle difficult problems such as continual^{60,63} and multitask learning⁶⁵ and propose solutions to the credit assignment problem without backpropagation^{56,57}. However, the processing power and efficiency of biological networks, largely endowed by their dendrites, are still far from being matched by respective ANNs. Towards this goal, we focused on the structured connectivity and restricted sampling characteristics of dendrites, two prominent features that are conserved across brain regions and species^{76,77}, suggesting that their role in information processing is likely to be very important.

We constructed a set of dendritic ANNs that leverage the structured connectivity and restricted sampling features of biological dendrites to enhance learning. dANNs are constructed as typical ANNs with two hidden layers, in which the first (dendritic) layer is connected in a sparse and structured manner to the second (somatic) layer so that it resembles the structured connectivity of biological dendrites to their respective somata. In line with experimental data, input sampling to the dendritic layer is also sparse and inspired by the receptive fields of neurons in the visual cortex^{67,77}, whereby dendrites (and neurons) sample only a restricted part of the visual space. We compared our models to vanilla ANNs across numerous image classification tasks and found that they are superior in performance accuracy, degree of overfitting, robustness to noise, and sequential learning. Compared to vANN, these benefits are achieved with orders of magnitude fewer trainable parameters, making dANNs more efficient and effective. As there is growing concern that the demand for computational resources to develop and apply AI models could lead to a significant increase in the electricity consumption of data centers worldwide⁷⁸, dANNs are particularly valuable in space-constrained and other energy-constrained settings. This concerns hardware implementations where the zeroed-out connections in dANNs can be omitted, offering significant energy and space savings for both training and inference. Even though we focused on simple ANNs that form the foundation of nearly all DL architectures, it is important to note that our study also offers a framework for integrating dendritic features into various models, such as convolutional neural networks, transformers, and others⁷⁹. This involves replacing the fully connected layers commonly found in deep learning architectures with our dANN model.

To dissect the relative contributions of structured connectivity vs. restricted input sampling, we also compared dANN models to fully connected ANNs and sparse (randomly-connected) ANNs furnished with the input sampling types that were used in the dendritic models. We found that both features are important for reducing overfitting and

achieving accuracy efficiency gains, with their relative contribution varying with the dataset and task used.

Of note, while some of the sparse networks approximate the performance of dANNs, the space of all sparse networks of a given size is vast, and finding those networks that work best is non-trivial. This work demonstrates that a biologically-inspired structured architecture (namely the dendritic one) can result in improved and more efficient performance. This finding is unexpected, as the bio-inspired architecture—including tree-like hidden layer connectivity and sampling with receptive fields—is just one of the too many possible sparse architectures. Knowing that nature has identified a network architecture that is more efficient than classical neuronal network architectures is important for designing efficient systems, without the need for extensive and expensive grid searches to identify such architectures.

Throughout these comparisons, the dANN model with LRF subsampling emerged as the most efficient configuration from a computational perspective. LRF subsampling assumes that individual dendrites of a given soma preferentially receive inputs from focused regions of a visual scene and that these regions are not necessarily close in visual space (Fig. 1). In the visual cortex, this could amount to receiving clustered inputs from presynaptic neurons that share similar feature selectivity, allowing for selective and distributed information processing within the dendritic tree. Several studies in both the cortex and the hippocampus provide evidence that dendrites can sample from the same feature space and receive inputs with correlated tuning properties^{40,80–83}. Additionally, dendrites can exhibit local, branch-specific integration and plasticity⁸⁴. This feature-based input organization is further supported by observations of branch-specific dendritic depolarizations and the generation of local dendritic spikes in response to specific input combinations^{29,85}. This structured organization of inputs has been suggested to confer important computational advantages for learning and memory storage^{49,86–88}. Overall, our findings demonstrate that dANNs with restricted input sampling consistently perform more efficiently than vANNs and are comparable to all variants of sANNs. This highlights the benefits of biologically-inspired approaches in improving model effectiveness, pointing towards promising directions for future research and applications in neural network design.

Our interpretability analysis revealed that dANN models—unlike vANN models—form primarily mixed-selective nodes. This finding aligns with experimental evidence of mixed-selective neurons in various cortical regions of the mouse and primate brains^{89–92}. Such neurons can encode multiple task-relevant features simultaneously, a property believed to be important for flexible decision-making and information processing, especially in higher cortical regions such as the prefrontal cortex. The ability to form and utilize mixed-selective nodes suggests that dANNs may enable more efficient and adaptable information processing, similar to biological neural networks. Further research into the similarities and differences between mixed-selective nodes in ANNs and mixed-selectivity neurons in the brain could provide valuable insights into the principles underlying intelligent behavior.

Beyond the findings reported here, other studies have also adopted dendritic properties in ANNs and DL models. These studies are different and complementary to ours in several ways. For example,

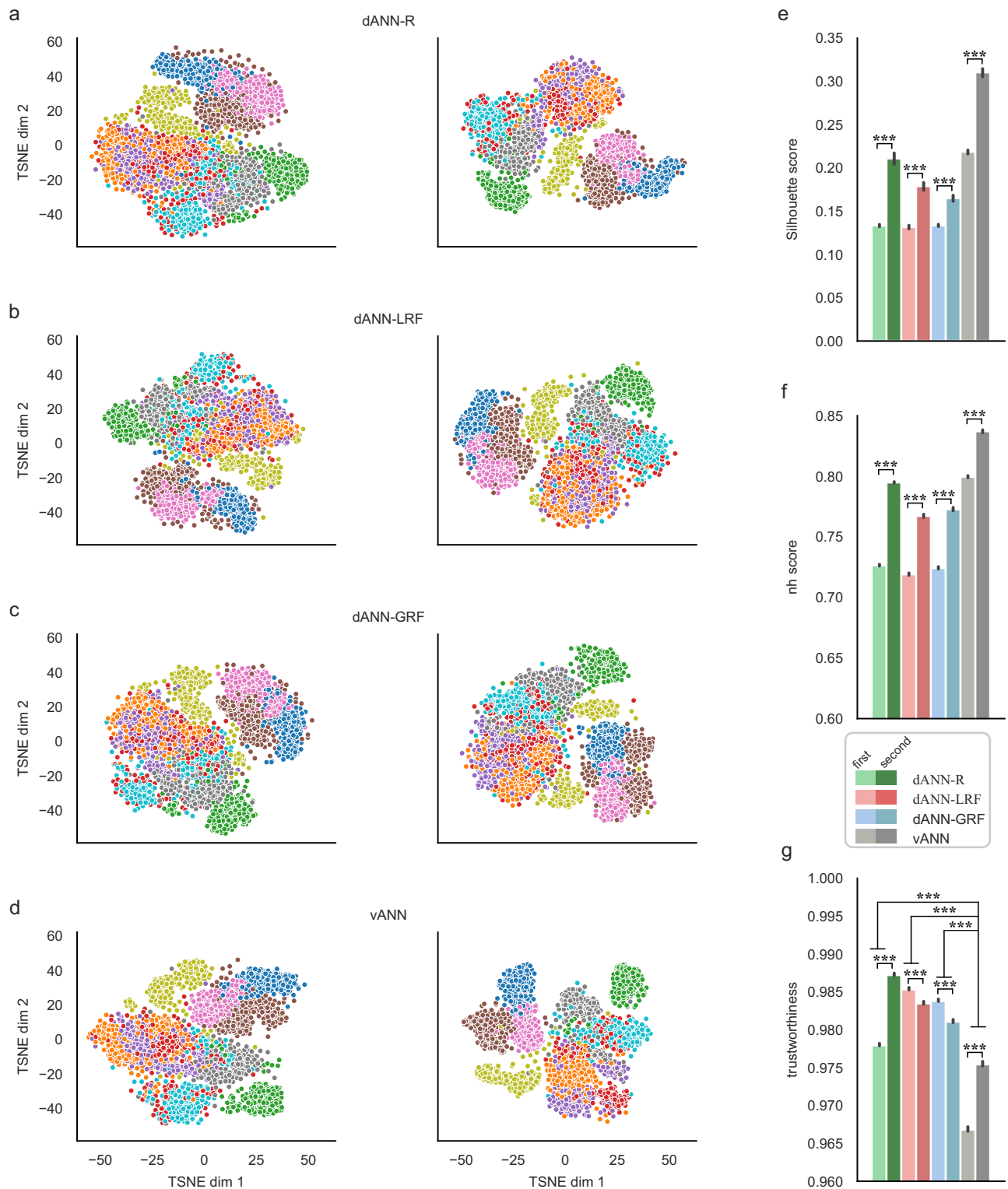


Fig. 6 | Learned representations. **a–d** TSNE projections of the activations for the first (left column) and the second (right column) hidden layers corresponding to the three dANN and the vANN models. Different colors denote the image categories of the FMNIST dataset. While the figure shows the results of one run, the representations are consistent across 10 runs of the TSNE algorithm (data not shown). **e** Silhouette scores of the representations (2-way ANOVA: model $F(3,32) = 1598.31$, $p < 10^{-3}$, layer $F(1,32) = 2130.39$, $p < 10^{-3}$, model \times layer $F(3,32) = 105.20$, $p < 10^{-3}$). **f** Neighborhood scores of the representations,

calculated using 11 neighbors (2-way ANOVA: model $F(3,32) = 8624.78$, $p < 10^{-3}$, layer $F(1,32) = 18512.42$, $p < 10^{-3}$, model \times layer $F(3,32) = 299.51$, $p < 10^{-3}$).

g Trustworthiness of the representations, calculated using 11 neighbors (2-way ANOVA: model $F(3,32) = 6187.66$, $p < 10^{-3}$, layer $F(1,32) = 1856.84$, $p < 10^{-3}$, model \times layer $F(3,32) = 1777.98$, $p < 10^{-3}$). In all barplots the error bars represent the 95% confidence interval across $N = 5$ initializations for each model and 10 runs of the TSNE algorithm per initialization. Stars denote significance with unpaired t-test (two-tailed) with Bonferroni's correction.

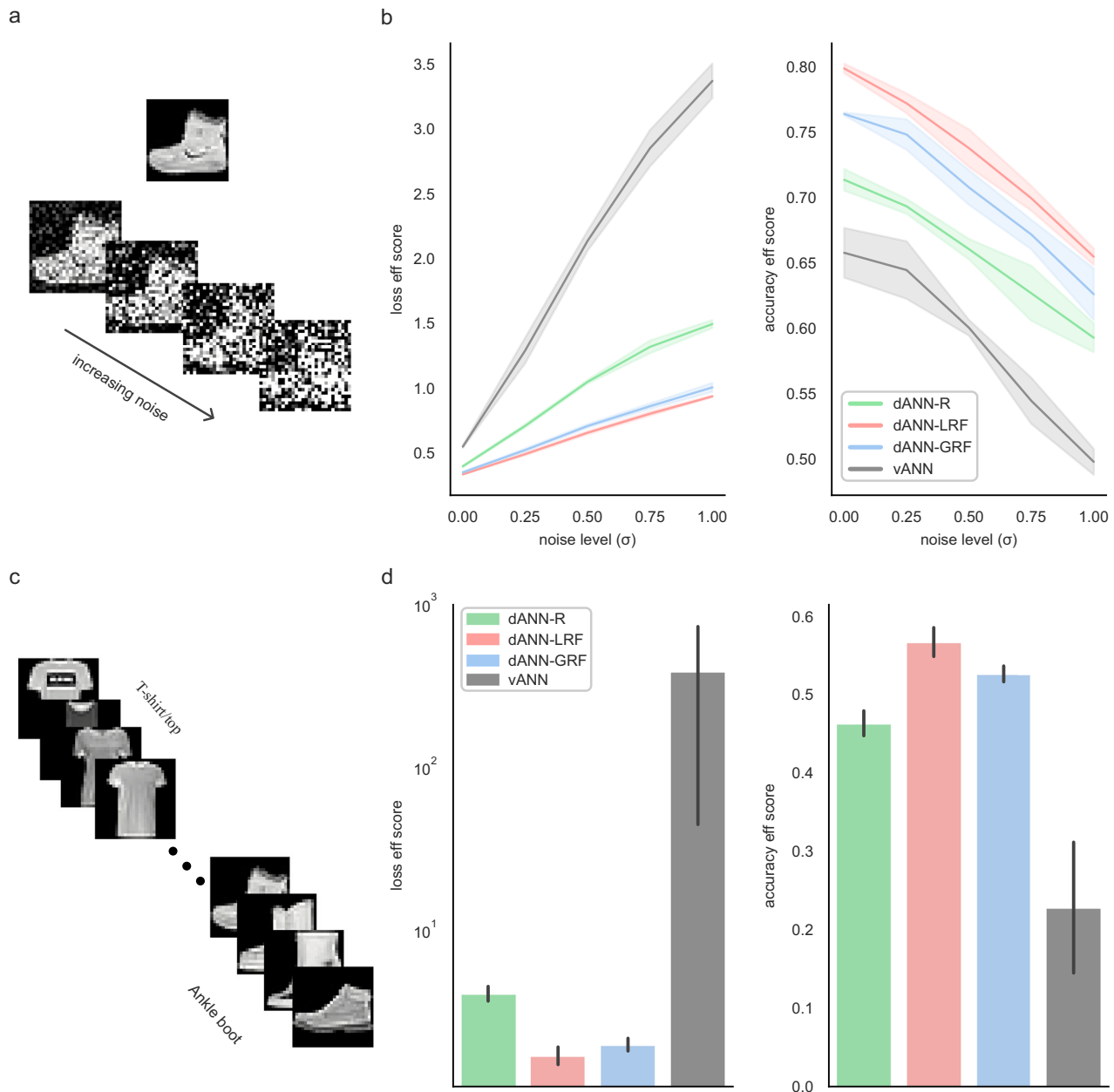


Fig. 7 | Dendritic ANNs are more accurate and efficient than vANNs when inputs are noisy or presented in a sequential manner. **a** An example of one FMNIST image with variable Gaussian noise. Sigma (σ) is the standard deviation of the Gaussian noise. **b** Testing loss (left) and accuracy (right) efficiency scores for all models and noise levels. Shades represent one standard deviation across $N = 5$

network initializations for each model. **c** The sequential learning task. **d** As in (b), but showing the loss (left) and accuracy (right) efficiency scores for the sequential task. Errorbars denote one standard deviation across $N = 5$ initializations for each model. See Table 2 and Supplementary Table 3 for the accuracy and loss values.

one approach implemented dendrites as max-pooling or average-pooling layers^{63,64,93}, two methods that are extensively used in DL models⁹⁴. Other approaches were more abstract, modeling dendrites as a multiplicative component and/or using all-to-all connectivity from the input layer onto dendrites^{62,95–97}. Some methods use specific dendritic components, such as normalization of the weights or specific dendritic spikes, but their connectivity matrices become sparse using, for example, evolutionary algorithms, or they contain fully connected layers^{65,98,99}. Lastly, dendritic models incorporating local learning rules have been used to study how the brain solves the credit assignment problem, showing promising learning capabilities but are not easily applicable to large ML applications^{55,56,74}.

Similar to Jones and Kording⁵⁸, our modeling approach considers dendrites as an additional layer that provides weighted inputs to the

somatic nodes. This means that dANN can be viewed as a sparse ANN from an ML perspective. In ANNs, sparsity can be achieved by pruning after training^{100,101}, using evolutionary algorithms^{102,103}, specific regularization^{104,105} during training, or iterative methods applied before training^{106,107}. Similar to the latter approaches, in dANNs, the connectivity sparsity is handcrafted from the start, creating a fixed architecture that is significantly smaller than typical vANNs undergoing a pruning process. This makes training faster and more efficient, as no pruning is required. Furthermore, inputs to dendrites are not randomly allocated but can be constructed based on RFs, setting our model apart from traditional sparse networks that rely on random connectivity initialization. Finally, RFs are created before training and can be modified to capture the most essential characteristics of a dataset. These attributes lend biological inspiration to our dANN

Table 2 | The table shows the maximum performance achieved by each type of model on the sequential learning task

Model performance— sequential learning		
Models	Test accuracy (%)	Test loss
dANN	56.834 ± 1.6748	3.3947 ± 0.3063
dANN-LRF	59.658 ± 1.5005^a	1.6532 ± 0.1783^a
dANN-GRF	59.348 ± 1.3412	1.8007 ± 0.1537
vANN	28.458 ± 9.4994	313.466 ± 240.8529

Reported scores correspond to the models that achieved the best test accuracy.

The corresponding test loss values for these models are also depicted. Test accuracy and loss are listed as mean ± standard deviation across $N = 5$ initializations for each model.

^aBold fonts denote the top metric per column.

models and are expected to be advantageous for neuromorphic hardware implementation, particularly when faced with space limitations and increased energy consumption resulting from lengthy node connections.

It is crucial to acknowledge the boundaries and limitations of our dANN architecture. In the implementation presented here, to maintain their initial connectivity, dANNs necessitate an extra boolean mask multiplication after every gradient descent step. This additional step results in a higher computational expense regarding floating-point operations. Moreover, during training, we employ the back-propagation algorithm and discard some gradients that aren't linked to an existing connection, potentially losing vital information from other gradient directions that could result in faster convergence. Using locally computed gradients that are dependent only on the connected nodes would overcome these limitations and further improve the efficiency/performance of our dANN models, but such an implementation is currently not possible with existing ML platforms and requires custom-made code. Of note, the same limitations apply to sparse network architectures that are implemented using the existing ML platforms. Finally, the cable weights used here are unconstrained. A more bio-realistic approach would be to restrict these weights to positive values or set them to fixed, positive values. Nevertheless, we believe our work is valuable as it offers new insights into the benefits, i.e., improved accuracy, less overfitting, and much fewer parameters, that can be gained by adopting dendritic features in classical ANNs. These advantages combined with their ability to scale relatively well with depth, render the proposed dANN networks a potentially powerful alternative to classical ANNs through their incorporation in DL architectures such as convolutional neural nets or transformers.

Overall, we show that implementing dendritic properties can significantly enhance the learning capabilities of ANNs, making them both accurate and efficient. These findings hold great promise as they suggest that integrating biological characteristics could be crucial for optimizing the sustainability and effectiveness of ML algorithms.

Methods

Network architectures

We have developed a range of traditional ANNs consisting of two hidden layers and an output layer that matches the number of classes. To create the dANN model, we first create two boolean masks that determine the synaptic weights between the input and dendritic layers, as well as the cable weights between dendritic and somatic layers. Once we have initialized the model, we apply these masks to achieve a sparse network with structured synaptic and cable weights (Eq. 1).

$$\mathbf{W}_k \rightarrow \mathbf{W}_k \odot \mathbf{M}_k \quad (1)$$

where \mathbf{W}_k denotes the weights, \mathbf{M}_k the boolean mask associated k -th layer, and \odot is the Hadamard (elementwise) product.

The calculation for the forward pass is obtained by linearly combining the inputs with the weights and adding the bias in each node of all layers. Finally, the output is obtained by passing the summation through a (nonlinear) activation function (Eq. 2).

$$\mathbf{A}_k = f(\mathbf{W}_k \mathbf{X}_k + \mathbf{b}_k) \quad (2)$$

where \mathbf{X}_k and \mathbf{A}_k denote the inputs to the k -th layer and its activations, respectively. $f(\cdot)$ is the activation function.

In the output layer, we calculate the loss, which is then propagated back to calculate the gradients with respect to all trainable parameters respectively. To ensure the same connectivity as the original model, we zero out all gradients calculated in non-existent connections (Eq. 3).

$$\frac{\partial \mathcal{L}}{\partial \theta} \leftarrow \frac{\partial \mathcal{L}}{\partial \theta} \odot \mathbf{M}_k \quad (3)$$

where \mathcal{L} is the loss and θ denotes the trainable parameters.

Synaptic connections

To define the input to the dendritic connectivity matrix, we utilize three distinct strategies. Our first approach involves random allocation, where each dendrite receives 16 inputs (pixels) which are randomly selected from the image (dANN-R). Our second approach utilizes local-constructed receptive fields (dANN-LRF), where each dendrite again receives 16 inputs (pixels), but this time they are sampled from a restricted part of the image. To do so, we randomly select a pixel to represent the center of the receptive field for each dendrite. In particular, the central pixel is drawn from a uniform distribution. Then, the 16 inputs are chosen from the 4×4 neighborhood of that pixel. The process is repeated for all dendrites. Finally, we utilize a global-constructed receptive field (dANN-GRF). In this approach, we select a pixel to represent the receptive field center for each soma instead of each dendrite. Then, the central pixel of each dendrite belonging to that soma has a central pixel drawn from a uniform distribution around the central somatic pixel. Finally, dendrites receive 16 inputs from the 4×4 neighborhood of their central pixel, as before.

Datasets

The dANN models are trained to classify images into one of their respective classes. The MNIST¹⁰⁸ consists of handwritten digits from 0 to 9. Fashion MNIST¹⁰⁹ is an alternative to MNIST and consists of clothing images: T-shirt/top, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Kuzushiji MNIST¹¹⁰ is a drop-in replacement for the MNIST dataset consisting of one Japanese character representing each of the ten rows of Hiragana. All of these datasets come with 60,000 training and 10,000 test images. Extended MNIST¹¹¹ follows the same conversion paradigm used to create the MNIST dataset. The result is a set of datasets that constitute more challenging classification tasks involving letters and digits. Here, we used the 47 balanced classes with 731,668 training and 82,587 testing images. All MNIST variants consist of 28×28 grayscale images. Finally, CIFAR-10¹¹² consists of images of objects or animals in ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset contains 50,000 training and 10,000 test images. The images are 32×32 pixels in three color channels.

For our experiments, we trained the models with 90% of the training data, keeping the remaining 10% for validation. Once the training was complete, we evaluated the performance on the test set.

Hyperparameters

Our models were trained using the Adam optimization algorithm, with the default parameters of a learning rate of 0.001 and betas of 0.9 and 0.999. To ensure efficient training, we utilized a minibatch size of 128. The number of epochs was variable for each dataset but the same

across models. Specifically, for MNIST, FMNIST, and KMNIST we used 15, 20, and 20 epochs, respectively, whereas for EMNIST and CIFAR10 we used 50 epochs. For the sequential learning scenario, we used 50 epochs to train the models. In our dANN models, each dendrite receives inputs from 16 input neurons. We calculate the loss using the cross-entropy function and have set the activation function of all nodes to the Leaky Rectified Linear Unit (ReLU) with a negative slope of 0.1, except for the output nodes, which utilize the softmax activation function.

Efficiency scores

To calculate the efficiency scores for all networks, we formulated the accuracy (*aes*) and loss (*les*) efficiency scores, respectively. To do so, we normalized the accuracy and loss with a factor f that takes values from $[1, \infty]$ and is the ratio of the logarithm, with base 10, of the number of trainable parameters of a model i with the minimum number of trainable parameters of the compared models times the number of epochs needed to reach minimum validation loss (Eq. 4).

$$f = \frac{\log_{10} k_i}{\min_m (\log_{10} (k_m))} \quad (4)$$

where k_i denotes the number of trainable parameters multiplied by the number of epochs needed to reach the minimum validation loss of the model i , and m denotes the compared models.

By dividing the accuracy by the factor f , the score remains in $[0, 100]$ while multiplying the loss with f the loss score remains in $[0, \infty]$ (Eq. 5).

$$aes = \frac{\text{accuracy}(\%)}{f}, les = loss \cdot f \quad (5)$$

Using these scores, models with large numbers of trainable parameters and training epochs have lower accuracy efficiency scores and higher loss efficiency scores compared to models with fewer trainable parameters and fewer training epochs.

Model comparison

In the comparisons shown in Figs. 3, 4, and 7 and Supplementary Fig. 10, we chose the best model based on either accuracy or loss, independently. This means that the best model for accuracy might be different from the best model for loss. Accuracy measures the percentage of correct predictions out of the total predictions made and is a simple metric for classification tasks. On the other hand, loss (here measured by the cross-entropy function) assesses how well the model's predictions match the true labels, considering the confidence of the predictions, not just their correctness. This evaluation is important because a model could achieve high accuracy with confident yet incorrect predictions, which can be misleading. Additionally, certain applications may prioritize minimizing loss over maximizing accuracy, so it's important to choose the model based on the specific task requirements.

Interpretability analysis

Synaptic, cable, and output weights: To display the learned weights, we constructed histograms with 20 bins and utilized kernel density estimation (KDE) to approximate the underlying distribution using a continuous probability density curve. The KDE plot smooths the observations with a Gaussian kernel, generating a continuous estimation. The histograms are built by concatenating the learned weights across $N = 5$ initializations for each ANN model.

Entropy: To calculate the entropy for the first and second hidden layers, we used the activations of all nodes during evaluation against the test set. First, we created the hit matrix ($\mathbf{HM} \in \mathbb{R}^{n_{\text{class}} \times D}$) for each

layer that assigns the number of times a node was activated (activity above zero) for images belonging to the same category.

To calculate normalized probabilities, we add an extra row to \mathbf{HM} containing the number of times a node remained inactive. Thus, the summation across rows is equal to the number of data samples in the test set.

Obtaining the probability matrix by dividing the \mathbf{HM} by the number of images, we calculate the entropy for each node in the k -th layer (Eq. 6).

$$H_k = - \sum_{i=1}^D p(x) \log_2 p(x) \quad (6)$$

We plot the entropy distributions using histograms with 20 bins and the KDE method to estimate a continuous probability density function. We removed inactive nodes from the analysis.

Selectivity: To calculate how selective a node is, we calculated how many categories it was activated for. We consider activity significant if a node was activated for over 400 images of a specific category. Thus, we ended with integers in $[1, n_{\text{class}}]$, with 1 denoting class-specificity and n_{class} total mixed-selectivity. As selectivity is a discrete metric, we plot the histograms with n_{class} bins and without using KDE.

Dimensionality reduction and analysis

To calculate the representations of the hidden layers, we used the t-distributed stochastic neighbor embedding (TSNE) dimensionality reduction method⁷³ with perplexity equal to 50. We chose this technique based on its widespread popularity and proven ability to preserve neighborhoods and clusters in projections. Activations for a given layer, the subject of our analysis, are extracted strictly for a random subset of 2000 observations from the test sets to aid visual presentation. We visualize projections as scatterplots, with points colored to show their class. To assess the quality of the projection and its discriminatory power, we employ two metrics. The Silhouette score calculates the global structure of the projection. It shows if activations of images belonging to the same category are close in the reduced space¹¹³, and the neighborhood hit score (*nh*) shows the local structure in the projection and indicates how well classes are visually separated^{114,115}.

Silhouette score is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample (Eq. 7).

$$S_i = \frac{b - a}{\max(a, b)}, i \in \{1, 2, \dots, n\} \quad (7)$$

and takes values in $[-1, 1]$, with values closer to 1 denoting good clustering and values close to 0 indicating overlapping clusters. Negative values indicate a sample has been assigned to the wrong cluster, as a different cluster is more similar. The Silhouette score is the average over all samples.

The *nh* score denotes how similar a point is to its k nearest neighbors in the reduced space. For a given k , the *nh* score for a point x is the percentage of the k -nearest neighbors that belong to the same class as x (Eq. 8).

$$nh(k) = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{\sum_{j=1}^k l_j}{k} \right\}, l_j = 1 \text{ if } r_j \in \mathcal{C}^{[x_i]} \text{ else } 0 \quad (8)$$

where \mathcal{C}^{x_i} denotes the category of the data point x_i , and r_j is the point in its neighborhood. Then the neighborhood hit score is calculated as an average across all points x in the dataset. The score is bounded in $[0, 1]$, with higher values denoting better neighborhood compactness and small values of misplaced data points. Here, we use $k = 11$.

Trustworthiness is a metric that expresses the extent to which the local structure is retained after the dimensionality reduction¹¹⁶ (Eq. 9).

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in \mathcal{N}_k^i} \max(0, (r(i, j) - k)) \quad (9)$$

where $r(i, j)$ denotes the rank of the datapoint j according to the pairwise distances between the low-dimensional datapoints, and \mathcal{N}_k^i represents the k -nearest neighbors of datapoint i in the low-dimensional space, but not in the high-dimensional space. Thus, any unexpected nearest neighbors in the low-dimensional space are penalized proportionally to their rank in the high-dimensional space. The trustworthiness is within $[0, 1]$. Here, we use $k = 11$.

Computing resources and software

All simulations were performed on a custom machine under the Debian GNU/Linux Trixie/sid (kernel version 6.6.15-2) operating system with 64GB of RAM, Intel® Core™ i5-10400 CPU @ 2.90 GHz, and an NVIDIA GeForce RTX 3080 Ti GPU @ 12GB. We implemented all models using the Keras 2.15.0 functional API¹¹⁷ with TensorFlow 2.15.0 backend¹¹⁸ under Python 3.9.18 (conda 23.7.4). For better handling of the training process, we used a custom training loop. For data analysis and visualization, we utilized various Python modules, including NumPy 1.24.4¹¹⁹, scikit-learn 1.4.1¹²⁰, pandas 1.5.3¹²¹, matplotlib 3.8.3¹²², seaborn 0.13.2¹²³ and seaborn-image 0.8.0¹²⁴.

Statistical analysis

For all standard statistical tests (detailed in figure legends), the significance level α was 0.05. To correct for multiple comparisons, α was divided by the number of tests according to the Bonferroni procedure. Throughout the figures, p values are denoted by * ($p < 0.05$), ** ($p < 0.01$), and *** ($p < 0.001$). To compare the dependent value among different groups (models \times layers), we used a two-way analysis of variance (ANOVA) followed by an unpaired t-test (two-tailed) with Bonferroni's correction whenever statistical difference was observed for post hoc comparisons. The statistical analysis was performed using the pingouin 0.5.4 library¹²⁵.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The source code that generates all Figures and the data that support this study are accessible on GitHub¹²⁶: https://github.com/Poirazi-Lab/dendritic_anns.

Code availability

The code underlying this study is available on GitHub¹²⁶: https://github.com/Poirazi-Lab/dendritic_anns.

References

- Attwell, D. & Laughlin, S. B. An energy budget for signaling in the grey matter of the brain. *J. Cereb. Blood Flow Metab.* **21**, 1133–1145 (2001).
- Luccioni, A. S., Jernite, Y. & Strubell, E. Power hungry processing: watts driving the cost of AI deployment? In *Proc. 2024 ACM Conference on Fairness, Accountability, and Transparency* 85–99 (FACCT, 2024).
- Strubell, E., Ganesh, A. & McCallum, A. Energy and policy considerations for deep learning in NLP. In *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650 (Association for Computational Linguistics, Florence, Italy, 2019).
- Mehonic, A. & Kenyon, A. J. Brain-inspired computing needs a master plan. *Nature* **604**, 255–260 (2022).
- McCloskey, M. & Cohen, N. J. Catastrophic interference in connectionist networks: the sequential learning problem. in *Psychology of Learning and Motivation* vol. 24 109–165 (Elsevier, 1989).
- Abraham, W. C. & Robins, A. Memory retention—the synaptic stability versus plasticity dilemma. *Trends Neurosci.* **28**, 73–78 (2005).
- Mesnil, G. et al. Unsupervised and transfer learning challenge: a deep learning approach. In *Proc. ICML Workshop on Unsupervised and Transfer Learning* 97–110 (JMLR, 2012).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- Guo, Y. et al. Deep learning for visual understanding: a review. *Neurocomputing* **187**, 27–48 (2016).
- Chai, J. & Li, A. Deep learning in natural language processing: a state-of-the-art survey. In *2019 International Conference on Machine Learning and Cybernetics*, 1–6 (ICMLC, 2019).
- Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
- Fuchs, F., Song, Y., Kaufmann, E., Scaramuzza, D. & Durr, P. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **6**, 4257–4264 (2021).
- Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **1168**, 022022 (2019).
- Yang, T.-J., Chen, Y.-H., Emer, J. & Sze, V. A method to estimate the energy consumption of deep neural networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers* 1916–1920 (IEEE, 2019).
- Nazaré, T. S., Da Costa, G. B. P., Contato, W. A. & Ponti, M. Deep convolutional neural networks and noisy images. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (eds. Mendoza, M. & Velastin, S.) vol. 10657 416–424 (Springer International Publishing, Cham, 2018).
- Dodge, S. & Karam, L. Understanding how image quality affects deep neural networks. In *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)* 1–6 (IEEE, 2016).
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C. & Wermter, S. Continual lifelong learning with neural networks: a review. *Neural Netw.* **113**, 54–71 (2019).
- McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133 (1943).
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
- Spruston, N. Pyramidal neurons: dendritic structure and synaptic integration. *Nat. Rev. Neurosci.* **9**, 206–221 (2008).
- Major, G., Larkum, M. E. & Schiller, J. Active properties of neocortical pyramidal neuron dendrites. *Annu. Rev. Neurosci.* **36**, 1–24 (2013).
- Larkum, M. E., Wu, J., Duverdin, S. A. & Gidon, A. The guide to dendritic spikes of the mammalian cortex in vitro and in vivo. *Neuroscience* **489**, 15–33 (2022).
- Poirazi, P., Brannon, T. & Mel, B. W. Pyramidal neuron as two-layer neural network. *Neuron* **37**, 989–999 (2003).
- Tzivilaki, A., Kastellakis, G. & Poirazi, P. Challenging the point neuron dogma: FS basket cells as 2-stage nonlinear integrators. *Nat. Commun.* **10**, 3664 (2019).
- Beniaguev, D., Segev, I. & London, M. Single cortical neurons as deep artificial neural networks. *Neuron* **109**, 2727–2739.e3 (2021).
- Häusser, M. & Mel, B. Dendrites: bug or feature? *Curr. Opin. Neurobiol.* **13**, 372–383 (2003).
- Poirazi, P. & Papoutsis, A. Illuminating dendritic function with computational models. *Nat. Rev. Neurosci.* **21**, 303–321 (2020).

28. Larkum, M. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends Neurosci.* **36**, 141–151 (2013).
29. Gidon, A. et al. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science* **367**, 83–87 (2020).
30. Shepherd, G. M. & Brayton, R. K. Logic operations are properties of computer-simulated interactions between excitable dendritic spines. *Neuroscience* **21**, 151–165 (1987).
31. Harnett, M. T., Makara, J. K., Spruston, N., Kath, W. L. & Magee, J. C. Synaptic amplification by dendritic spines enhances input cooperativity. *Nature* **491**, 599–602 (2012).
32. Schachter, M. J., Oesch, N., Smith, R. G. & Taylor, W. R. Dendritic spikes amplify the synaptic signal to enhance detection of motion in a simulation of the direction-selective ganglion cell. *PLoS Comput. Biol.* **6**, e1000899 (2010).
33. Ariav, G., Polsky, A. & Schiller, J. Submillisecond precision of the input-output transformation function mediated by fast sodium dendritic spikes in basal dendrites of CA1 pyramidal neurons. *J. Neurosci.* **23**, 7750–7758 (2003).
34. Softky, W. Sub-millisecond coincidence detection in active dendritic trees. *Neuroscience* **58**, 13–41 (1994).
35. Roome, C. J. & Kuhn, B. Dendritic coincidence detection in Purkinje neurons of awake mice. *eLife* **9**, e59619 (2020).
36. Stuart, G. J. & Häusser, M. Dendritic coincidence detection of EPSPs and action potentials. *Nat. Neurosci.* **4**, 63–71 (2001).
37. Naud, R. & Sprekeler, H. Sparse bursts optimize information transmission in a multiplexed neural code. *Proc. Natl. Acad. Sci.* **115**, E6329–E6338 (2018).
38. Anandan, E. S., Husain, R. & Selvakumaran, K. Auditory attentional filter in the absence of masking noise. *Atten. Percept. Psychophys.* **83**, 1737–1751 (2021).
39. Benezra, S. E., Patel, K. B., Campos, C. P., Hillman, E. M. C. & Bruno, R. M. Learning enhances behaviorally relevant representations in apical dendrites. *eLife* **13**, RP98349 (2024).
40. Takahashi, N., Oertner, T. G., Hegemann, P. & Larkum, M. E. Active cortical dendrites modulate perception. *Science* **354**, 1587–1590 (2016).
41. Takahashi, N. et al. Active dendritic currents gate descending cortical outputs in perception. *Nat. Neurosci.* **23**, 1277–1285 (2020).
42. Xu, T. et al. Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature* **462**, 915–919 (2009).
43. Otor, Y. et al. Dynamic compartmental computations in tuft dendrites of layer 5 neurons during motor behavior. *Science* **376**, 267–275 (2022).
44. Lai, C. S. W., Adler, A. & Gan, W.-B. Fear extinction reverses dendritic spine formation induced by fear conditioning in the mouse auditory cortex. *Proc. Natl. Acad. Sci.* **115**, 9306–9311 (2018).
45. Xu, Z. et al. Fear conditioning and extinction induce opposing changes in dendritic spine remodeling and somatic activity of layer 5 pyramidal neurons in the mouse motor cortex. *Sci. Rep.* **9**, 4619 (2019).
46. Letzkus, J. J. et al. A disinhibitory microcircuit for associative fear learning in the auditory cortex. *Nature* **480**, 331–335 (2011).
47. Kastellakis, G., Silva, A. J. & Poirazi, P. Linking memories across time via neuronal and dendritic overlaps in model neurons with active dendrites. *Cell Rep.* **17**, 1491–1504 (2016).
48. Malakasis, N., Chavlis, S. & Poirazi, P. Synaptic turnover promotes efficient learning in bio-realistic spiking neural networks. In *57th Asilomar Conference on Signals, Systems, and Computers* (IEEE, 2023).
49. Poirazi, P. & Mel, B. W. Impact of active dendrites and structural plasticity on the memory capacity of neural tissue. *Neuron* **29**, 779–796 (2001).
50. Kaifosh, P. & Losonczy, A. Mnemonic functions for nonlinear dendritic integration in hippocampal pyramidal circuits. *Neuron* **90**, 622–634 (2016).
51. Makarov, R., Pagkalos, M. & Poirazi, P. Dendrites and efficiency: optimizing performance and resource utilization. *Curr. Opin. Neurobiol.* **83**, 102812 (2023).
52. Pagkalos, M., Makarov, R. & Poirazi, P. Leveraging dendritic properties to advance machine learning and neuro-inspired computing. *Curr. Opin. Neurobiol.* **85**, 102853 (2024).
53. Chavlis, S. & Poirazi, P. Drawing inspiration from biological dendrites to empower artificial neural networks. *Curr. Opin. Neurobiol.* **70**, 1–10 (2021).
54. Acharya, J. et al. Dendritic computing: branching deeper into machine learning. *Neuroscience* **489**, 275–289 (2022).
55. Guerguiev, J., Lillicrap, T. P. & Richards, B. A. Towards deep learning with segregated dendrites. *eLife* **6**, e22901 (2017).
56. Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A. & Naud, R. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nat. Neurosci.* **24**, 1010–1019 (2021).
57. Illing, B., Ventura, J., Bellec, G. & Gerstner, W. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Adv. Neural Inf. Proc. Sys.* **34**, 30365–30379 (2021).
58. Jones, I. S. & Kording, K. P. Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree? *Neural Comput.* **33**, 1554–1571 (2021).
59. Kording, K. P. & König, P. Supervised and unsupervised learning with two sites of synaptic integration. *J. Comput. Neurosci.* **11**, 207–215 (2001).
60. Kirkpatrick, J. et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**, 3521–3526 (2017).
61. Siegel, M., Kording, K. P. & König, P. Integrating top-down and bottom-up sensory processing by somato-dendritic interactions. *J. Comput. Neurosci.* **8**, 161–173 (2000).
62. Tang, C. et al. Dendritic neural network: a novel extension of dendritic neuron model. *IEEE Trans. Emerg. Top. Comput. Intell.* 1–12 (2024).
63. Iyer, A. et al. Avoiding catastrophe: active dendrites enable multi-task learning in dynamic environments. *Front. Neurobotics* **16**, 846219 (2022).
64. Wu, X., Liu, X., Li, W. & Wu, Q. Improved expressivity through dendritic neural networks. *Adv. Neural Inf. Proc. Sys.* **31**, 8068–8079 (2018).
65. Wybo, W. A. M. et al. NMDA-driven dendritic modulation enables multitask representation learning in hierarchical sensory processing pathways. *Proc. Natl. Acad. Sci.* **120**, e2300558120 (2023).
66. Ringach, D. L. Mapping receptive fields in primary visual cortex. *J. Physiol.* **558**, 717–728 (2004).
67. Bair, W. Visual receptive field organization. *Curr. Opin. Neurobiol.* **15**, 459–464 (2005).
68. Chen, Y. et al. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *Interspeech 2015* 1136–1140 (ISCA, 2015).
69. Morgan, A. T., Petro, L. S. & Muckli, L. Scene representations conveyed by cortical feedback to early visual cortex can be described by line drawings. *J. Neurosci.* **39**, 9410–9423 (2019).
70. Nusrat, I. & Jang, S. B. A comparison of regularization techniques in deep neural networks. *Symmetry* **10**, 648 (2018).
71. Bartunov, S. et al. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Adv. Neural Inf. Proc. Sys.* **31**, 9390–9400 (2018).
72. Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. & Peste, A. Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.* **22**, 1–124 (2021).
73. Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

74. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Adv. Neural Inf. Proc. Sys.* **31**, 8735–8746 (2018).
75. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. Dendritic error backpropagation in deep cortical microcircuits. Preprint at <http://arxiv.org/abs/1801.00062> (2017).
76. Holley, Z. L., Bland, K. M., Casey, Z. O., Handwerk, C. J. & Vidal, G. S. Cross-regional gradient of dendritic morphology in isochronically-sourced mouse supragranular pyramidal neurons. *Front. Neuroanat.* **12**, 103 (2018).
77. Gaucher, Q. et al. Complexity of frequency receptive fields predicts tonotopic variability across species. *eLife* **9**, e53462 (2020).
78. De Vries, A. The growing energy footprint of artificial intelligence. *Joule* **7**, 2191–2194 (2023).
79. Sarker, I. H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2**, 420 (2021).
80. Wilson, D. E., Whitney, D. E., Scholl, B. & Fitzpatrick, D. Orientation selectivity and the functional clustering of synaptic inputs in primary visual cortex. *Nat. Neurosci.* **19**, 1003–1009 (2016).
81. Lavzin, M., Rapoport, S., Polsky, A., Garion, L. & Schiller, J. Non-linear dendritic processing determines angular tuning of barrel cortex neurons in vivo. *Nature* **490**, 397–401 (2012).
82. Cichon, J. & Gan, W.-B. Branch-specific dendritic Ca(2+) spikes cause persistent synaptic plasticity. *Nature* **520**, 180–185 (2015).
83. Fu, M., Yu, X., Lu, J. & Zuo, Y. Repetitive motor learning induces coordinated formation of clustered dendritic spines in vivo. *Nature* **483**, 92–95 (2012).
84. O'Hare, J. K., Wang, J., Shala, M. D., Polleux, F. & Losonczy, A. Variable recruitment of distal tuft dendrites shapes new hippocampal place fields. Preprint at <https://doi.org/10.1101/2024.02.26.582144> (2024).
85. Beaulieu-Laroche, L., Toloza, E. H. S., Brown, N. J. & Harnett, M. T. Widespread and highly correlated somato-dendritic activity in cortical layer 5 neurons. *Neuron* **103**, 235–241.e4 (2019).
86. Kastellakis, G., Cai, D. J., Mednick, S. C., Silva, A. J. & Poirazi, P. Synaptic clustering within dendrites: an emerging theory of memory formation. *Prog. Neurobiol.* **126**, 19–35 (2015).
87. Kastellakis, G. & Poirazi, P. Synaptic clustering and memory formation. *Front. Mol. Neurosci.* **12**, 300 (2019).
88. Kastellakis, G., Tasciotti, S., Pandi, I. & Poirazi, P. The dendritic engram. *Front. Behav. Neurosci.* **17**, 1212139 (2023).
89. Parthasarathy, A. et al. Mixed selectivity morphs population codes in prefrontal cortex. *Nat. Neurosci.* **20**, 1770–1779 (2017).
90. Fusi, S., Miller, E. K. & Rigotti, M. Why neurons mix: high dimensionality for higher cognition. *Curr. Opin. Neurobiol.* **37**, 66–74 (2016).
91. Barak, O., Rigotti, M. & Fusi, S. The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off. *J. Neurosci. Off. J. Soc. Neurosci.* **33**, 3844–3856 (2013).
92. Rigotti, M. et al. The importance of mixed selectivity in complex cognitive tasks. *Nature* **497**, 585–590 (2013).
93. Wu, X. et al. Mitigating communication costs in neural networks: the role of dendritic nonlinearity. Preprint at <http://arxiv.org/abs/2306.11950> (2023).
94. Zhao, L. & Zhang, Z. A improved pooling method for convolutional neural networks. *Sci. Rep.* **14**, 1589 (2024).
95. Ji, J., Gao, S., Cheng, J., Tang, Z. & Todo, Y. An approximate logic neuron model with a dendritic structure. *Neurocomputing* **173**, 1775–1783 (2016).
96. Todo, Y., Tamura, H., Yamashita, K. & Tang, Z. Unsupervised learnable neuron model with nonlinear interaction on dendrites. *Neural Netw.* **60**, 96–103 (2014).
97. Zhang, Y. et al. A lightweight multi-dendritic pyramidal neuron model with neural plasticity on image recognition. *IEEE Trans. Artif. Intell.* 1–13 (2024).
98. Bird, A. D., Jedlicka, P. & Cuntz, H. Dendritic normalisation improves learning in sparsely connected artificial neural networks. *PLOS Comput. Biol.* **17**, e1009202 (2021).
99. Hodassman, S., Vardi, R., Tugendhaft, Y., Goldental, A. & Kanter, I. Efficient dendritic learning as an alternative to synaptic plasticity hypothesis. *Sci. Rep.* **12**, 6571 (2022).
100. Mengiste, S. A., Aertsen, A. & Kumar, A. Effect of edge pruning on structural controllability and observability of complex networks. *Sci. Rep.* **5**, 18145 (2015).
101. Han, S., Pool, J., Tran, J. & Dally, W. Learning both weights and connections for efficient neural network. *Adv. Neural Inf. Proc. Sys.* **28**, 1135–1143 (2015).
102. Liu, Z. et al. Learning efficient convolutional networks through network slimming. *IEEE International Conference on Computer Vision (ICCV)* 2755–2763 (IEEE, 2017).
103. Mocanu, D. C. et al. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nat. Commun.* **9**, 2383 (2018).
104. Louizos, C., Welling, M. & Kingma, D. P. Learning sparse neural networks through L_0 regularization. Preprint at <https://arxiv.org/abs/1712.01312> (2017).
105. Scardapane, S., Comminiello, D., Hussain, A. & Uncini, A. Group sparse regularization for deep neural networks. *Neurocomputing* **241**, 81–89 (2017).
106. Patil, S. M. & Dovrolis, C. PHEW: Constructing sparse networks that learn fast and generalize well without training data. In *Proceedings of the 38th International Conference on Machine Learning* 8432–8442 (PMLR, 2021).
107. Tanaka, H., Kunin, D., Yamins, D. L. & Ganguli, S. Pruning neural networks without any data by iteratively conserving synaptic flow. *Adv. Neural Inf. Proc. Sys.* **33**, 6377–6389 (2020).
108. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
109. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Preprint at <http://arxiv.org/abs/1708.07747> (2017).
110. Clanuwat, T. et al. Deep Learning for Classical Japanese Literature. Preprint at <https://arxiv.org/abs/1801.00062> (2017).
111. Cohen, G., Afshar, S., Tapson, J. & van Schaik, A. EMNIST: an extension of MNIST to handwritten letters. Preprint at <https://arxiv.org/abs/1702.05373> (2017).
112. Krizhevsky, A. Learning multiple layers of features from tiny images. <https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf> (2019).
113. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
114. Paulovich, F. V., Nonato, L. G., Minghim, R. & Levkowitz, H. Least square projection: a fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Trans. Vis. Comput. Graph.* **14**, 564–575 (2008).
115. Rauber, P. E., Fadel, S. G., Falcao, A. X. & Telea, A. C. Visualizing the hidden activity of artificial neural networks. *IEEE Trans. Vis. Comput. Graph.* **23**, 101–110 (2017).
116. Maaten, L. Learning a parametric embedding by preserving local structure. In *Proc. Twelfth International Conference on Artificial Intelligence and Statistics* 384–391 (PMLR, 2009).
117. Chollet, F. & others. Keras. (2015). <https://keras.io>
118. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. In *OSDI'16: Proc. 12th USENIX Conf. Operating Systems*

- Design and Implementation (chairs Keeton, K. & Roscoe, T.) 265–283 (USENIX Association, 2016).
119. Harris et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
120. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
121. McKinney, W. Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference 51–56 (2010).
122. Hunter, J. D. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007).
123. Waskom, M. L. Seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).
124. Jariwala, S. seaborn-image: image data visualization. GitHub. <https://github.com/SarthakJariwala/seaborn-image> (2020).
125. Vallat, R. Pingouin: statistics in Python. *J. Open Source Softw.* **3**, 1026 (2018).
126. Chavlis, S., Poirazi, P. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. Poirazi-Lab/dendritic_anns. <https://doi.org/10.5281/zenodo.14360989> (2024).
127. Park, J. et al. Contribution of apical and basal dendrites to orientation encoding in mouse V1 L2/3 pyramidal neurons. *Nat. Commun.* **10**, 5372 (2019).

Acknowledgements

The authors thank Dr. Michalis Pagkalos, Zinovia Stefanidi, Dr. Athanasia Papoutsis, Prof. Blake Richards, Ioannis-Rafail Tzonevrakis, Dr. Eirini Troulinou, and Prof. Grigorios Tsagkatakis for their valuable and constructive feedback. This work was supported by the NIH (GA: 1R01MH124867-04, to P.P.) and the European Commission (H2020-FETOPEN-2018-2019-2020-01, FET-Open Challenging Current Thinking, project NEUREKA, GA-863245 to P.P.).

Author contributions

S.C. and P.P. conceived the project, designed the experiments, and wrote the manuscript. S.C. implemented the models, performed the experiments, analyzed the data, and created the figures. P.P. supervised and funded the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-56297-9>.

Correspondence and requests for materials should be addressed to Panayiota Poirazi.

Peer review information *Nature Communications* thanks Joel Zylberberg and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025