Neuro-inspired Ensemble-to-Ensemble Communication Primitives for Sparse and Efficient ANNs

Orestis Konstantaropoulos*, Stelios Manolis Smyrnakis†, Maria Papadopouli*‡§

*Archimedes, Athena Research Center, Athens, Greece
†Department of Neurology, Brigham and Women's Hospital, Harvard Medical School, Boston, USA
‡Department of Computer Science, University of Crete, Greece
§Institute of Computer Science, Foundation for Research and Technology-Hellas, Greece

Abstract—The structure of biological neural circuits—modular, hierarchical, and sparsely interconnected—reflects an efficient trade-off between wiring cost, functional specialization, and robustness. These principles offer valuable insights for artificial neural network (ANN) design, especially as networks grow in depth and scale. Sparsity, in particular, has been widely explored for reducing memory and computation, improving speed, and enhancing generalization. Motivated by systems neuroscience findings, we explore how patterns of functional connectivity in the mouse visual cortex—specifically, ensemble-to-ensemble communication—can inform ANN design. We introduce G2GNet, a novel architecture that imposes sparse, modular connectivity across feedforward layers. Despite having significantly fewer parameters than fully connected models, G2GNet achieves superior accuracy on standard vision benchmarks. To our knowledge, this is the first architecture to incorporate biologically observed functional connectivity patterns as a structural bias in ANN design. We complement this static bias with a dynamic sparse training (DST) mechanism that prunes and regrows edges during training. We also propose a Hebbian-inspired rewiring rule based on activation correlations, drawing on principles of biological plasticity. G2GNet achieves up to 75% sparsity while improving accuracy by up to 4.3% on benchmarks, including Fashion-MNIST, CIFAR-10, and CIFAR-100, outperforming dense baselines with far fewer computations.

Index Terms—Neuronal Ensembles, Sparsity in Deep Learning, Functional Connectivity, Computer Vision

I. INTRODUCTION

Neuroscience has been a significant source of inspiration in the design of deep learning, for addressing challenges, such as energy cost, ability for generalization and continual learning. Concepts such as spiking neural networks and neuromorphic hardware take cues from the event-driven, sparse firing patterns observed in the brain to enable low-power, efficient computation [1]–[3]. Biologically plausible learning rules, motivated by Hebbian learning and spike-timing-dependent plasticity (STDP), offer promising alternatives to backpropagation for unsupervised or local learning [4]–[8]. Biologically-inspired components, such as precalibrated Gabor filter banks, lateral inhibition, and winner-take-all dynamics, have been incorporated in ANNs to improve robustness against adversarial perturbations and enhance interpretability [9], [10].

The structure of biological neural circuits, e.g., its modular, hiearachical, sparse inter-connectivity–thought to reflect an efficient trade-off between wiring cost, functional specialization, and robustness–offers further insight for the design of ANNs. The influence of such connectivity principles on ANNs has been increasingly explored [11]–[21]. Sparsity, in particular, is both a biological necessity and a desirable property in artificial systems. As artificial neural networks (ANNs) continue to expand in depth and size [22], sparsity has been widely studied as a means to reduce memory footprint, computational cost, and often training and inference time [23]–[26]. Moreover, sparsity may not only improve the efficiency of artificial networks but also their ability to generalize by promoting more compact internal representations and improving robustness to noise.

While the properties of individual biological neurons are well characterized, our understanding of how cortical neuron networks coordinate to process information remains limited. *Neuronal ensembles* that exhibit *synchronous firing* are thought to transmit shared information to downstream targets more efficiently [27] and have been highlighted in several seminal studies examining both spontaneous and stimulus-evoked activity patterns [28]–[40]. Putative communication primitives that mediate neuronal activity and information transmission, across cortical layers, further highlight the functional role of sparsity in information processing and communication [41]. For example, a recent study [42] found that biological neurons, along with their functionally-connected partners (i.e., other neurons with which they exhibit *statistically-significant pairwise synchronous firing*), form elementary *multi-neuronal ensembles (modules)* that act as core information-processing units both within and across granular and supragranular layers in mouse primary visual cortex. Layer 4 is considered as the primary input layer of the cortex, while supragranular layers Layer 2 and Layer 3, receiving primarily input from L4, integrate and relay information to other cortical areas. These neural ensembles exhibit strong cooperativity, with *aggregate cofiring* activity—rather than *individual neuron identity*—driving the L2/3 neuron responses in a ReLU-

like manner. In particular, across cortical laminae, the firing probability of Layer 2/3 pyramidal neurons exhibits a *ReLU-like activation* profile as a function of the number of its L4 functionally-connected neurons. Notably, the firing probability of a Layer 2/3 neuron increases *sharply* when at least approximately 13% of its Layer 4 partners co-fire, a relatively rare event, reflecting a *nonlinear integration mechanism* that promotes both *reliable supra-threshold transmission* and *sparse activity*. Moreover, ensemble-to-ensemble information transfer from Layer 4 to Layer 2/3 demonstrates enhanced specificity, sensitivity, accuracy, and precision. We speculate that such ensembles (modules), being *flexible* and *multiplexing*, may support task-specific processing, adaptation, and learning through dynamic reconfiguration of their interactions.

Inspired by this work [42], we aim to examine the role of these "pathways", i.e., ensemble-to-ensemble communication primitives, by imposing a similar structure on the feedforward layers of ANNs. The proposed architecture, **G2GNet**, introduces sparse, modular connectivity across layers. This approach allows us to analyze the role of functional connectivity structures observed in mice visual cortex while simultaneously proposing an efficient deep learning architecture. Despite having significantly fewer parameters and computational load compared to a fully-connected one, G2GNet achieves superior performance on computer vision benchmarks. To our knowledge, this is the *first* study that integrates *functional connectivity* patterns and organization principles identified in the visual cortex as a *structural bias into ANNs* to assess their effect on the performance of artificial architecture. Specifically, each layer in G2GNet is partitioned into groups that *preferentially connect* to corresponding groups in adjacent layers. We complement this static structural bias with a dynamic sparse training (DST) mechanism that prunes *and* regrows edges *during training*. Furthermore, we introduce a Hebbian-based criterion for edge updates, guided by activation correlations among neurons—inspired by the plasticity observed in biological neural networks.

In summary, this work makes the following key contributions: **1)** Develops an ANN architecture that incorporates neuronal grouping strategies inspired by the functional connectivity patterns of biological neural networks as a *structural bias*; **2)** Integrates a Hebbian-inspired criterion for *dynamic addition* and *removal of connections* in the proposed architecture; **3)** Analyzes the impact of different layer-wise grouping strategies on network accuracy; **4)** Demonstrates that the proposed structural bias achieves up to 75% sparsity in feedforward layers while improving accuracy by up to 4.3% on computer vision benchmarks, such as Fashion-MNIST, CIFAR-10 and CIFAR-100, compared to fully-connected architectures.

II. BACKGROUND: SPARSITY IN DEEP LEARNING

One of the most common approaches to **compress** neural networks is through weight pruning after training [43]–[47]. A key insight from this line of work is provided by the Lottery Ticket Hypothesis [48], [49], which suggests the existence of **sparse subnetworks**, "winning tickets", that when *trained* in

isolation, can match the performance of their dense counterparts. That is, a "winning ticket" is a subnetwork that performs as well as the full network. To bypass the need for training large, dense models to uncover such subnetworks, alternative methods have been proposed that initialize and train sparse networks directly [50], [51]. These approaches fall under the umbrella of *static sparse training*, as the set of active edges remains *fixed* throughout the training process.

In contrast, **dynamic sparse training** (DST) explores methods for updating the sparse connectivity of a network during training [52]–[56]. DST *dynamically* prunes and regrows connections based on specific criteria, allowing the network structure to evolve, while maintaining a constant number of edges. For example, the Sparse Evolutionary Training (SET) method [52] removes weights with the smallest magnitudes, while other approaches [55], [57] incorporate gradient information to guide the update process, often at the expense of increased memory and computational cost. Inspired by neuroplasticity, DST offers a framework for topological flexibility, enabling the model to adapt its structure. This process mirrors the synaptic regeneration and rewiring in the brain as well as the dynamic "reorganization" of the functional-connectivity observed in visual cortex [42].

Apart from the distinction between *dynamic* and *static sparsity*, sparsity can also be categorized as either **structured** or **unstructured**. In unstructured sparsity, individual weights (connections) are pruned *independently* based on a given criterion, commonly magnitude-based pruning. While this typically leads to irregular sparsity patterns (i.e., individual elements in the weight matrix are zeroed out), the overall structure and dimensions of the layers remain unchanged. This form of sparsity offers high theoretical compression but is difficult to exploit for actual speedups on general-purpose hardware (e.g., CPUs or GPUs) without specialized sparse libraries or hardware support [47].

In contrast, structured sparsity applies *pruning at higher levels of abstraction*, such as channels (feature maps), filters (entire convolution kernels), neurons (rows in fully connected layers), or even entire layers (e.g., residual blocks in ResNets) [44], [58]. This results in more *regular and hardware-friendly sparsity patterns* that can be efficiently accelerated using standard hardware. However, achieving high accuracy under such constraints is often more difficult [59].

Among structured sparsity techniques, $N{:}M$ sparsity has received particular attention due to its compatibility with modern accelerators [60]. In this scheme, every block of M elements in the weight matrix contains exactly N *non-zero values*, a property also shared by the connectivity patterns in G2GNet. Recent work has also explored how to dynamically train $N{:}M$ sparse networks using DST [61].

Our work builds on the growing body of research of brain-inspired deep learning and sparsity in ANNs by incorporating functional connectivity patterns from biological networks as a *structural bias in ANN architectures*. Our sparse architecture builds on *neuronal ensembles, i.e., groups* (used interchangeably in the paper) *within each layer* and their *inter-*
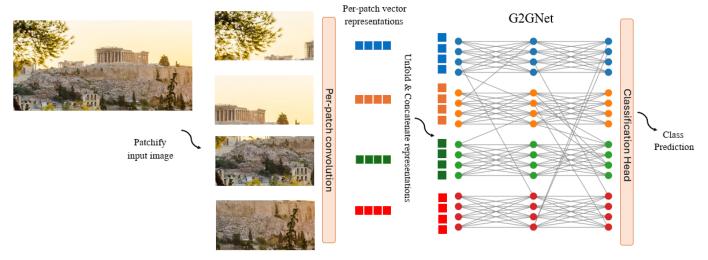
Fig. 1: Overview of the proposed G2GNet architecture. The input image is first divided into patches (four in this example), and a convolutional layer is applied *independently* to each patch to produce vector representations. These vectors are concatenated and passed through a feed-forward network with a predefined *sparse structure*, inspired by *group-to-group* (neural ensemble to ensemble) communication patterns observed in biological neural circuits. We refer to this architecture as **G2GNet**. A final linear classifier predicts the class of the input image. Here, for clarity, we demonstrate a simplified instance of G2GNet where each layer has 16 nodes divided in 4 groups.

layer communication. This introduces the notion of **pathways**: active groups that transmit information between layers, bringing *modularity* and *structured sparsity* into the design. In addition, we explore the integration of *dynamic* sparse training techniques within this biologically informed framework. We introduce a *Hebbian-like criterion* for pruning and growing connections inspired by biological synaptic plasticity.

III. METHODOLOGY

We will examine the role of the ensemble-to-ensemble information flow observed in biological networks by imposing a structural bias on the feedforward layers of ANNs. A key challenge is how the functional grouping observed in biological neurons can be translated into *structural grouping in ANNs*. We propose that **grouping** can be achieved by ensuring that *neurons within the same group share highly similar connectivity patterns*, that is, neurons in a specific group (say i) connect to the *same neurons* in the next layer with the *same probability*. Additionally, each group is *strongly coupled with its corresponding group in the next layer*, while maintaining *only sparse connections with other groups* in that layer. This design results in a highly sparse network with small-world-like properties across its layers. In the following paragraphs, we present the details of our architecture.

A. Sparse layer structure inspired by primary visual cortex (V1)

Inspired by the modular organization and group-specific connectivity observed in the primary visual cortex (V1), we propose a sparse architecture that enforces structured modularity in feedforward layers. Conceptually, our **G2GNet** architecture defines a *probabilistic connectivity mask* between consecutive layers that activates *only* a subset of the possible

connections, favoring *within strongly coupled groups communication*.

Let the number of neurons in layer i be N_i, and in layer $i+1$ be N_{i+1}. We formally define the construction of a layer's connectivity as follows:

1) Partition the N_i neurons of layer i into G_i groups, each containing N_i/G_i neurons.
2) Partition the neurons of layer $i+1$ into G_{i+1} groups.
3) For each neuron in group k of layer i, connect it to neurons in group k of layer $i+1$ with probability **p**, and to neurons in *all other groups* of layer $i+1$ with a *much lower probability* $\mathbf{p}' \ll \mathbf{p}$.

This construction results in a connectivity matrix between layers i and $i+1$ that is approximately *block diagonal*: the *diagonal* blocks, corresponding to same-index groups, are *dense* with probability **p**, while the *off-diagonal blocks* are much *sparser* with probability p'. Figure 1 schematically depicts the G2GNet architecture and Figure 2 visualizes the resulting connectivity mask given a specific instance. Both Figures depict a simplified case where $G_i = 4, N_i = 16$ for each layer i. The hyperparameters used in our experiments are noted in Section IV.

As information flows across layers, it primarily travels along pathways formed by densely-connected group pairs that are connected with a high probability p. In parallel, information also propagates through significantly sparser connections between the groups, i.e., connected with the lower probability $p' << p$, allowing for secondary information leakage across pathways. For clarity, throughout the rest of the paper, we refer to p as *intra-pathway probability* and p' as *inter-pathway probability*.

The proposed pattern can be highly sparse. Instead of maintaining all $N_i \times N_{i+1}$ parameters, the number of active
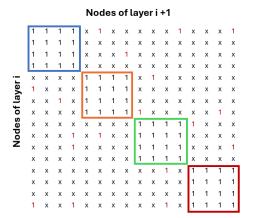
Nodes of layer i +1

Fig. 2: Adjacency matrix between two consecutive layers in G2GNet. Assume layers i and $i+1$ each contain 16 neurons, partitioned into 4 groups. Neurons in group k of layer i are connected to neurons in group k of layer $i+1$ with high intra-pathway probability ($p = 1$), and to neurons in other groups with low inter-pathway probability ($p' = 0.05$). A value of 1 in the adjacency matrix indicates an *active connection*; otherwise, the entry is marked as x. The resulting matrix exhibits an approximately block-diagonal structure, reflecting the *group-to-group* connectivity pattern.

connections becomes

$$N_i \times N_{i+1} \times \left[\frac{p}{G_{i+1}} + \frac{p'(G_{i+1} - 1)}{G_{i+1}} \right].$$

For example, with $p = 1$, $p' = 0.05$, and $G_i = G_{i+1} = 8$, the architecture achieves 85% sparsity, utilizing only 15% of the potential parameters.

One might ask why not simply retain a random 15% of connections rather than enforcing a modular structure. However, inspired by biological observations of functional neuronal groups, our structured connectivity consistently yields higher accuracy on vision datasets, see IV. In addition, the explicit group structure enables *structured sparsity*, which is more compatible with acceleration on modern hardware.

B. Grouping strategies

The way neurons are grouped within each layer plays a crucial role in *shaping how information flows* through the network. Conceptually, grouping determines which neurons are densely connected and which interact only sparsely, thereby influencing both spatial locality and feature integration.

Several grouping strategies can be considered: neurons can be grouped by their index ("**Index-based**", Fig. 3A), preserving the ordering induced by the input. Alternatively, neurons can be assigned to groups *randomly*, disregarding any structure ("**Random**", Fig. 3B). In this work, we propose a *hybrid* method inspired by the MLPMixer architecture [62], which we call the *Mixer* strategy ("**Mixer**" Fig. 3C). The Mixer *alternates* between two grouping schemes in successive layers:

- In even-numbered layers, neurons are partitioned by index into consecutive blocks, as described previously.

- In odd-numbered layers, neurons are assigned to groups in an interleaved fashion: each neuron with index l is assigned to group k such that $k = l \mod G_i$.

Figure 3 illustrates this alternating grouping mechanism, again in the simplified case where $G_i = 4, N_i = 16$ for each layer i.

The choice of grouping affects *how spatial information from the input image is processed*. When neurons are grouped by index, *spatial proximity of the input pixels is preserved across layers*: neurons within the same group predominantly process information originating from the same spatial region of the input, while inter-pathway (sparser) connections enable *integration from more distant regions*. This index-based grouping is loosely inspired by the columnar structure in the higher mammals of neurons with similar properties, like orientation columns. In contrast, random grouping destroys this spatial locality, as neurons from different regions of the input are arbitrarily combined.
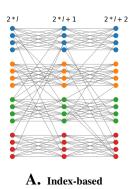
The proposed Mixer strategy strikes a balance: even-numbered layers preserve spatial locality by grouping neurons that process nearby input regions, while odd-numbered layers promote integration of information from different regions by mixing neurons from distinct spatial areas into the same group. Using MLPMixer terminology, in even layers groups correspond to *patches*, while in odd layers groups correspond to *channels*. Unlike the original MLPMixer, however, we allow communication both within and between patches and channels in every layer through the low-probability inter-pathway p' connections.
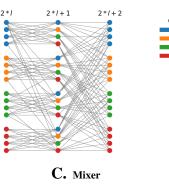
C. Dynamic Sparse Training

Having defined the G2GNet architecture and imposed its biologically inspired structural bias, we now introduce a mechanism to *dynamically update the network's sparse connectivity* during *training*, following the principles of Dynamic Sparse Training (DST). A common approach in DST is to initialize the network using the Erdős–Rényi (ER) method, which generates a random binary mask for each layer to enforce a predefined sparsity level. *During training*, the sparse connectivity is *periodically updated* every Δt training iterations. A fraction of the currently *active* edges is removed according to a pruning criterion C_P, rendering them inactive and excluding them from the forward and backward passes. An *equal number* of connections is then reintroduced (added) from the pool of inactive edges based on a growth criterion C_G, maintaining a constant overall sparsity level. Below we describe the pruning and growth (addition) criteria.

D. Rewiring Criteria during Training

We explore several combinations of pruning and growth criteria to apply to G2GNet. For pruning, we consider: **(i)** a backpropagation (BP) magnitude-based criterion that removes edges with the smallest (BP) weight, as in [52], [63], **(ii)** random selection, and **(iii)** a novel Hebbian-inspired criterion C_H, described below. For growth, we examine the random addition and the Hebbian-based criterion C_H. All pruning

Fig. 3: Comparison of neuron grouping strategies across three consecutive layers of the G2GNet. Each subfigure illustrates three consecutive layers, with neurons color-coded by group membership. *Note that groups are defined such that all neurons within a group share the same connection probabilities to neurons in the next layer.* We demonstrate a simplified instance of G2GNet where $G_i = 4, N_i = 16$ for each layer i. **(a) Index-based grouping:** neurons are divided into contiguous blocks by index, preserving spatial locality from the input image across layers. **(b) Random grouping:** neurons are assigned to groups without regard to position, disrupting spatial structure and coherence. **(c) Mixer strategy (proposed):** groups alternate between index-based (even-numbered layers) and interleaved assignments (odd-numbered layers, where group k contains all neurons l such that $k = l \mod G_i$). This approach maintains spatial locality in some layers while promoting integration across spatial regions in others—balancing modularity and feature mixing.

and regrowth steps are applied every $\Delta t = 1000$ iterations, dropping and adding 2.5% of the edges in each layer. A detailed sensitivity analysis, varying these hyperparameters, is presented in Section IV.

We propose a *Hebbian-inspired update mechanism* tailored to the modular structure of G2GNet. Specifically, we introduce a criterion C_H that adds or removes connections based on the functional connectivity of intergroup neurons. The idea is to favor connections between neurons whose activations are more strongly correlated, in line with Hebbian plasticity principles.

Let $x_i, x_j \in \mathbb{R}^B$ denote the activation vectors of neurons i and j over a batch of size B. We define the pairwise **functional connectivity of neurons i and j** during a period as the cosine similarity between their activation vectors during that period:

$$C_H(i,j) = \frac{x_i x_j^\top}{\|x_i\|_2 \cdot \|x_j\|_2} \quad (1)$$

where $\| \cdot \|_2$ denotes the L_2 norm. When used as a pruning criterion, edges with the lowest functional connectivity scores are removed. Conversely, when used as a growth criterion, *new edges are added* by selecting those with the highest functional connectivity scores from the set of *inactive* connections.

IV. EXPERIMENTS

A. Architectural Details

To evaluate the effectiveness of the proposed G2GNet architecture, we conduct experiments on three standard image classification benchmarks: Fashion-MNIST and CIFAR-10 each contain 10 classes, while CIFAR-100 includes 100 classes. Fashion-MNIST consists of grayscale images of size 28×28, whereas CIFAR-10 and CIFAR-100 contain RGB images of size 32×32. These datasets are widely used for evaluating the performance of novel neural network architectures under varying conditions. Although G2GNet is applied

in a purely feedforward setting, and thus results on CIFAR-100 are naturally lower, they still provide informative insights into the effectiveness of the approach.

Unless otherwise stated, we use the following default hyperparameters throughout our experiments: $p = 1$, $p' = 0.15$, and $G_i = 8$ for all layers. The G2GNet consists of three hidden layers, each with 1024 neurons with ReLU activations. The input image is divided into 16 patches, and a convolutional layer with 32 output channels is applied to each patch independently. The resulting patch embeddings are concatenated and passed through the G2GNet. Given this configuration, each group in the first hidden layer corresponds to two input patches. A final linear classifier produces the output logits.

The model is trained end-to-end for 20 epochs using backpropagation with cross-entropy loss. Optimization is performed using the Adam optimizer with a learning rate of 1×10^{-3}.

Is the G2GNet structure bias effective? To assess the benefit of our proposed structure, we compare it to several alternative architectures:

- *Fully-connected baseline:* We replace our G2GNet sparse network with fully-connected ones, either keeping the same width for each layer or adjusting the width to match the total number of parameters. Our proposed structure achieves higher accuracy when having equal number of parameters and similar or better accuracy when having 25% of the parameters across our benchmarks.
- *Random sparsity (Erdős–Rényi):* We construct uniformly random sparse layers, dropping edges with probability $\tilde{p}$ such that the total number of parameters matches ours. Again, our modular structure, which favors local connections, outperforms this random baseline, see Fig. 4A.

These results are summarized in Table 1.

TABLE I: Test accuracies (%) of different connectivity patterns on FashionMNIST (F-MNIST), CIFAR-10 and CIFAR-100. We compare fully-connected architectures with the same number of parameters (v1) or same width (v2) as ours, as well as ER random sparse networks and our V1-inspired topologies. e

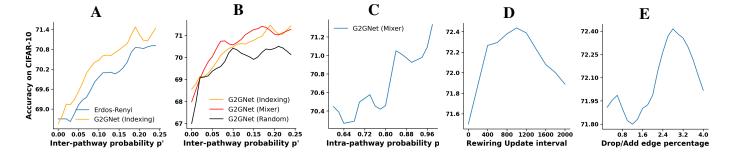| Connectivity Pattern	F-MNIST Accuracy (%)	CIFAR-10 Accuracy (%)	CIFAR-100 Accuracy (%)	Num. Parameters
Fully-Connected v1	90.2	68.4	38.7	806,400
Fully-Connected v2	90.9	71.5	40.8	*3,150,000*
ER Random Graph	89.2	69.8	40.7	806,100
G2GNet (Proposed)	**91.14**	**71.5**	**42.2**	806,100

Fig. 4: Experimental results on CIFAR-10 (a) As the inter-pathway probability p' of connecting sparsely connected groups, and thus the number of edges, increases, accuracy improves and plateaus around 70.8% for $p' \in [0.08, 0.15]$, corresponding to approximately 18% to 25% sparsity. Compared to Erdős-Rényi (ER) graphs of equal number of edges, our structure bias consistently yields higher accuracy. (b) The Mixer technique outperforms grouping by indexing across a wide range of p' values, while random grouping significantly underperforms. (c) Increasing the intra-pathway probability p of connecting densely connected groups leads to higher accuracy; we therefore set $p = 1$ in all experiments. (d) Effect of the rewiring update interval: accuracy is evaluated as a function of the number of training iterations between edge removal and addition steps. The best performance is observed when the update occurs every 1000 iterations. (e) Effect of the % of drop/add edges: accuracy is measured for different proportions of weights pruned and regrown, with the update interval fixed at 1000 iterations. The optimal performance is achieved around a 2.5% update rate. Reported values are mean accuracies over five runs with identical hyperparameters. For grouping strategy comparisons, the connectivity matrix initialization is kept fixed.

Understanding hyperparameters We now examine the impact of the key hyperparameters of our architecture on accuracy, using the CIFAR-10 dataset. Figure 4C shows that increasing the intra-pathway probability p consistently improves accuracy. For this reason, we adopt $p = 1$ as the default setting. Figure 4A illustrates the effect of the sparse inter-pathway probability p'. Interestingly, accuracy exhibits a convex relationship with p': at very low p', where ensembles are largely disconnected, accuracy drops. Conversely, at higher p', where the network comes closer to the fully-connected one, accuracy saturates and slightly decreases due to the loss of the structural inductive bias. The best performance - sparsity trade-off is achieved for $p' \in [0.08, 0.15]$, where the network remains sparse but inter-ensemble communication is preserved.

Comparing grouping techniques In Figure 4B, we also report the impact of the different grouping techniques described in Section III-B. We found that the Mixer technique performs best, as it enables dense modules to process information from diverse regions of the input image. Interestingly, when grouping is random, the information flow lacks any structure related to the spatial proximity of the input, and the results deteriorate to levels like the Erdős–Rényi connectivity pattern.

B. Sparse Training Evaluation

Table II shows the performance of various dynamic sparse training (DST) strategies applied to G2GNet on the CIFAR-10 and CIFAR-100 datasets. We evaluated all combinations of pruning (edge removal) criteria and the two growth (edge addition) methods described in Section III. Specifically, edges are pruned using one of the following strategies: (i) random selection, (ii) magnitude-based pruning (following the SET criterion [52]), or (iii) our proposed Hebbian-based criterion. Edges are then regrown either *randomly* or using the *Hebbian criterion*, which prioritizes connections between neurons with highly correlated activation patterns.

We did not consider DST methods that rely on dense gradient storage [55], [56], as they require maintaining gradient information for inactive connections, which undermines the computational benefits of sparsity.

Empirically, we found that DST strategies consistently improve the performance of G2GNet on CIFAR-10, and on CIFAR-100 when the Hebbian growth criterion is employed. Among the various combinations, the Hebbian criterion generally performs best for edge addition, consistently outperforming random growth. The most effective configuration involves pruning by magnitude (SET) and growing with the Hebbian criterion on CIFAR-10, while on CIFAR-100, the best results are achieved by using the Hebbian criterion for both pruning and growth.

We then fixed the DST configuration to use Hebbian-based criteria for both pruning and growth, and conducted a sensitivity analysis on two key hyperparameters: the *weight update interval* and the *proportion of connections modified per update* (see Figures 4D, 4E). In line with prior work [63], we found that performance is optimized when updates occur every 1000 training iterations, with an update rate of approximately 2.5%. When the update interval is too short or the update rate too high, the network struggles to adapt and train the newly-added parameters effectively. On the other hand, if updates are too infrequent or the update rate too low, the impact of the rewiring mechanism diminishes.

TABLE II: Test accuracies (%) on CIFAR-10 and CIFAR-100 for combinations of edge **removal** (rows) and edge **addition** (columns) in Dynamic Sparse Training (DST). Averages over five runs with fixed initial topology are reported.

	CIFAR-10		CIFAR-100	
	Addition Strategy			
Removal Strategy	**Random**	**Hebbian**	Random	**Hebbian**
SET (Magnitude-based) [52]	71.6	**72.7**	41.8	42.3
Random	**72.5**	72.0	41.8	42.28
Hebbian (ours)	71.5	72.5	41.4	**42.7**
Static baseline (no updates)	71.4		42.2	

V. DISCUSSION AND FUTURE WORK

Inspired by ensemble-to-ensemble information transmission observed in biological circuits of the mouse visual cortex, we proposed a structural bias for the feedforward layers of artificial neural networks. We refer to this architecture as **G2GNet**: a sparse network that facilitates information flow through static, coupled group pairs across adjacent layers, while still allowing *controlled information leakage* between uncoupled group pairs. This design introduces structured pathways through the depth of the network, which can be columnar, randomly shaped, or mixed, depending on the grouping strategy.

The initial connectivity pattern imposed by G2GNet may not contain the "winning ticket", a subnetwork that performs as well as the full network, as described by the Lottery Ticket Hypothesis. To address this, we incorporate a dynamic edge rewiring mechanism that maintains a fixed number of pruned and newly-activated edges while adapting connectivity during training. Central to this mechanism is our proposed **Hebbian-inspired criterion**, which adds or prunes edges based on the activation correlation between connected neurons. Our approach yields strong performance on computer vision benchmarks and enables **structured sparsity**, a form of sparsity that is more compatible with current hardware acceleration.

We demonstrate that: **i)** G2GNet is an effective method for matrix sparsification; **ii)** allowing communication between distinct pathways improves performance; and **iii)** functional connectivity patterns, and more specifically, pairwise correlations of the activation of neurons can guide the edge rewiring in ANNs, enhancing the efficiency.

While our method is *only conceptually* inspired by biological principles, such as ensemble-to-ensemble communication and information flow, it opens promising directions for future research. In particular, the notion of functional groups in biological networks could be extended to dynamic group formation in ANNs, driven by input data, internal network state, or even as a means for network training and avoiding of catastrophic forgetting. Currently, our experiments focus on feedforward layers, which limits the complexity of tasks evaluated. However, the underlying structural bias of G2GNet is applicable to convolutional layers as well as to the query, key, and value matrices of attention mechanisms in transformer-based architectures. Moreover, part of our on-going efforts includes the analysis of these ideas on SNNs and an in-depth examination of mixer-like grouping strategies. G2GNet provides an example of how the organizational principles identified in neuroscience (e.g., in [42]) may inform the design of ANNs, providing also a simulation tool for testing of neuro-computational hypotheses.

REFERENCES

[1] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, 2019.

[2] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, 2023.

[3] Z. Zhou, Y. Zhu, C. He, Y. Wang, S. YAN, Y. Tian, and L. Yuan, "Spikformer: When spiking neural network meets transformer," in *ICLR*, 2023.

[4] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in *NeurIPS*, 2016.

[5] T. Toosi and E. Issa, "Brain-like flexible visual inference by harnessing feedback feedforward alignment," in *NeurIPS*, 2023.

[6] G. Hinton, "The forward-forward algorithm: Some preliminary investigations," 2022. [Online]. Available: https://arxiv.org/abs/2212.13345

[7] G. Dellaferrera and G. Kreiman, "Error-driven input modulation: solving the credit assignment problem without a backward pass," in *International Conference on Machine Learning*. PMLR, 2022, pp. 4937–4955.

[8] A. Journé, H. G. Rodriguez, Q. Guo, and T. Moraitis, "Hebbian deep learning without feedback," 2023. [Online]. Available: https://arxiv.org/abs/2209.11883

[9] J. Dapello, T. Marques, M. Schrimpf, F. Geiger, D. Cox, and J. J. DiCarlo, "Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations," in *NeurIPS*, 2020.

[10] K. P. Panousis, S. Chatzis, and S. Theodoridis, "Stochastic local winner-takes-all networks enable profound adversarial robustness," 2021. [Online]. Available: https://arxiv.org/abs/2112.02671

[11] S. M. Naji, A. Abtahi, and F. Marvasti, "Efficient sparse artificial neural networks," 2021. [Online]. Available: https://arxiv.org/abs/2103.07674

[12] S. Herculano-Houzel, B. Mota, P. Wong, and J. H. Kaas, "Connectivity-driven white matter scaling and folding in primate cerebral cortex," *Proceedings of the National Academy of Sciences*, 2010.

[13] S. Xie, A. Kirillov, R. Girshick, and K. He, "Exploring randomly wired neural networks for image recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[14] X.-J. Zhang, J. M. Moore, T.-T. Gao, X. Zhang, and G. Yan, "Brain-inspired wiring economics for artificial neural networks," *PNAS Nexus*, 2025.

[15] Y. Zhang, J. Zhao, W. Wu, and A. Muscoloni, "Epitopological learning and cannistraci-hebb network shape intelligence brain-inspired theory for ultra-sparse advantage in deep learning," in *The Twelfth International Conference on Learning Representations*, 2024.

[16] G. Shen, D. Zhao, Y. Dong, and Y. Zeng, "Brain-inspired neural circuit evolution for spiking neural networks," *Proceedings of the National Academy of Sciences*, 2023.

[17] R. F. Betzel, A. Avena-Koenigsberger, J. Goñi, Y. He, M. A. de Reus, A. Griffa, P. E. Vértes, B. Mišic, J.-P. Thiran, P. Hagmann, M. van den Heuvel, X.-N. Zuo, E. T. Bullmore, and O. Sporns, "Generative models of the human connectome," *NeuroImage*, vol. 124, 2016.

[18] O. Sporns, *Networks of the Brain*. MIT press, 2016.

[19] ——, "Small-world connectivity, motif composition, and complexity of fractal neuronal connections," *Biosystems*, vol. 85, 2006.

[20] Barabasi, *Network Science*. Cambridge University Press, 2015. [Online]. Available: https://doi.org/10.1098/rsta.2012.0375

[21] V. Dorovatas, G. Paraskevopoulos, and A. Potamianos, "Auto-compressing networks," *arXiv preprint arXiv:2506.09714*, 2025.

[22] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020. [Online]. Available: https://arxiv.org/abs/2001.08361

[23] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1902.09574

[24] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks," *J. Mach. Learn. Res.*, Jan. 2021.

[25] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, 2022.

[27] I. Smyrnakis and S. Smirnakis, "Information Transfer Through Stochastic Transmission of a Linear Combination of Rates," *Neural Computation*, vol. 25, 2013.

[28] N. R. de Lorente, "Cerebral cortex: architecture, intracortical connections, motor projections," *Physiology of the nervous system*, 1949.

[29] M. Abeles, H. Bergman, E. Margalit, and E. Vaadia, "Spatiotemporal firing patterns in the frontal cortex of behaving monkeys," *Journal of Neurophysiology*, vol. 70, Oct. 1993.

[30] A. Grinvald, A. Arieli, M. Tsodyks, and T. Kenet, "Neuronal assemblies: Single cortical neurons are obedient members of a huge orchestra," *Biopolymers*, vol. 68, Feb. 2003.

[31] D. O. Hebb, "The first stage of perception: growth of the assembly," *The Organization of Behavior*, vol. 4, 1949.

[32] D. L. Ringach, "Spontaneous and driven cortical activity: implications for computation," *Current Opinion in Neurobiology*, vol. 19, 2009.

[33] L. Carrillo-Reid, W. Yang, Y. Bando, D. S. Peterka, and R. Yuste, "Imprinting and recalling cortical ensembles," *Science*, vol. 353, 2016.

[34] S. Panzeri, M. Moroni, H. Safaai, and C. D. Harvey, "The structures and functions of correlations in neural population codes," *Nature Reviews Neuroscience*, vol. 23, 2022.

[35] B. B. Averbeck, P. E. Latham, and A. Pouget, "Neural correlations, population coding and computation," *Nature reviews neuroscience*, 2006.

[36] R. Yuste, "From the neuron doctrine to neural networks," *Nature Reviews Neuroscience*, 2015.

[37] R. Cossart, D. Aronov, and R. Yuste, "Attractor dynamics of network up states in the neocortex," *Nature*, vol. 423, 2003.

[38] J.-e. K. Miller, I. Ayzenshtat, L. Carrillo-Reid, and R. Yuste, "Visual stimuli recruit intrinsically generated cortical ensembles," *Proceedings of the National Academy of Sciences of the USA*, 2014.

[39] E. Russo and D. Durstewitz, "Cell assemblies at multiple time scales with arbitrary lag constellations," *Elife*, vol. 6, 2017.

[40] R. Yuste, R. Cossart, and E. Yaksi, "Neuronal ensembles: Building blocks of neural circuits," *Neuron*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627323009674

[41] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, vol. 37, 1997.

[42] M. Papadopouli, I. Smyrnakis, E. Koniotakis, M.-A. Savaglio, C. Brozi, E. Psilou, G. Palagina, and S. M. Smirnakis, "Brain orchestra under spontaneous conditions: Identifying communication modules from the functional architecture of area v1," *bioRxiv*, 2024. [Online]. Available: https://www.biorxiv.org/content/early/2024/03/04/2024.02.29.582364

[43] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1989.

[44] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

[45] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[46] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, 2020.

[47] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks," *J. Mach. Learn. Res.*, 2021.

[48] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.

[49] H. Zhou, J. Lan, R. Liu, and J. Yosinski, *Deconstructing lottery tickets: zeros, signs, and the supermask*, 2019.

[50] N. Lee, T. Ajanthan, and P. Torr, "SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY," in *International Conference on Learning Representations*, 2019.

[51] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," 2020. [Online]. Available: https://arxiv.org/abs/2002.07376

[52] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature communications*, vol. 9, 2018.

[53] H. Mostafa and X. Wang, "Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization," in *International Conference on Machine Learning*. PMLR, 2019.

[54] G. Bellec, D. Kappel, W. Maass, and R. Legenstein, "Deep rewiring: Training very sparse deep networks," in *International Conference on Learning Representations*, 2018.

[55] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *International conference on machine learning*. PMLR, 2020.

[56] S. M. Jayakumar, R. Pascanu, J. W. Rae, S. Osindero, and E. Elsen, "Top-kast: top-k always sparse training," in *Neural Information Processing Systems*, 2020.

[57] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," 2020. [Online]. Available: https://openreview.net/forum?id=ByeSYa4KPS

[58] S. Narang, G. Diamos, S. Sengupta, and E. Elsen, "Exploring sparsity in recurrent neural networks," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=BylSPv9gx

[59] S. Lym, E. Choukse, S. Zangeneh, W. Wen, S. Sanghavi, and M. Erez, "Prunetrain: fast neural network training by dynamic sparse model reconfiguration," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019.

[60] A. Zhou, Y. Ma, J. Zhu, J. Liu, Z. Zhang, K. Yuan, W. Sun, and H. Li, "Learning n:m fine-grained structured sparse neural networks from scratch," 2021. [Online]. Available: https://arxiv.org/abs/2102.04010

[61] M. Lasby, A. Golubeva, U. Evci, M. Nica, and Y. Ioannou, "Dynamic sparse training with structured sparsity," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=kOBkxFRKTA

[62] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. P. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-mixer: An all-MLP architecture for vision," in *Advances in Neural Information Processing Systems*, 2021.

[63] A. I. Nowak, B. Grooten, D. C. Mocanu, and J. Tabor, "Fantastic weights and how to find them: Where to prune in dynamic sparse training," 2023. [Online]. Available: https://arxiv.org/abs/2306.12230