A DENDRITIC-INSPIRED NETWORK SCIENCE GENERATIVE MODEL FOR TOPOLOGICAL INITIALIZATION OF CONNECTIVITY IN SPARSE ARTIFICIAL NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Artificial neural networks (ANNs) achieve remarkable performance but at the unsustainable cost of extreme parameter density. In contrast, biological networks operate with ultra-sparse, highly organized structures, where dendrites play a central role in shaping information integration. Here we introduce the Dendritic Network Model (DNM), a generative framework that bridges this gap by embedding dendritic-inspired connectivity principles into sparse artificial networks. Unlike conventional random initialization, DNM defines connectivity through parametric distributions of dendrites, receptive fields, and synapses, enabling precise control of modularity, hierarchy, and degree heterogeneity. This parametric flexibility allows DNM to generate a wide spectrum of network topologies, from clustered modular architectures to scale-free hierarchies, whose geometry can be characterized and optimized with network-science metrics. Across image classification benchmarks (MNIST, Fashion-MNIST, EMNIST, CIFAR-10), DNM consistently outperforms classical sparse initializations at extreme sparsity (99%), in both static and dynamic sparse training regimes. Moreover, when integrated into state-of-the-art dynamic sparse training frameworks and applied to Transformer architectures for machine translation, DNM enhances accuracy while preserving efficiency. By aligning neural network initialization with dendritic design principles, DNM demonstrates that sparse bio-inspired network science modelling is a structural advantage in deep learning, offering a principled initialization framework to train scalable and energy-efficient machine intelligence.

1 INTRODUCTION

Artificial neural networks (ANNs) have demonstrated remarkable potential in various fields; however, their size, often comprising billions of parameters, poses challenges for both economic viability and environmental sustainability. Biological neural networks, in contrast, can efficiently process information using ultra-sparse structures (Drachman, 2005; Walsh, 2013).This efficiency arises from the brain's highly structured and evolutionarily optimized network topology. A central component of this architecture is the dendritic organization, the primary receptive surface of the neuron (Cuntz et al., 2010). Conventional ANNs omit a crucial component of the brain's efficiency since they traditionally depict neurons as simple point-like integrators, mainly ignoring the computing power inherent in the intricate structure of dendrites.

Research has revealed that dendrites are not passive conductors but active computational units capable of performing sophisticated, nonlinear operations and integration (London & Häusser, 2005; Larkum, 2013). This insight has motivated theoretical frameworks that model a single neuron as a multi-layer network, where dendritic branches act as nonlinear subunits that feed a final integrator at the cell body (Lauditi et al., 2025). As clarified by recent works on dendritic artificial neural networks (Li et al., 2020; Chavlis & Poirazi, 2025), the dendritic tree's ability to sample and nonlinearly integrate restricted parts of the input space can be used in neuromorphic physical networks (Li et al., 2020) or in artificial convolutional layers (Chavlis & Poirazi, 2025). In this paradigm, distinct dendritic branches process specific, localized receptive fields without sharing weights, allow-

ing for precise, location-specific feature integration. Additional efforts to translate these principles into neuromorphic systems have confirmed that also dendritic morphology has a significant impact on spatio-temporal processing and performance(Baek et al., 2024; Jones & Kording, 2021; 2022). These approaches, however, are often limited to fixed structures that mimic the computational non-linearity or direct morphology of biological neurons, overlooking the broader rules of connectivity, which also include non-uniform random network dynamic rewiring by synaptic turnover (Zhang et al., 2025; Frank et al., 2018).

Our understanding of these topological constraints has been revolutionized by the advent of large-scale functional connectomics (mic, 2025). These studies reveal that biological neural networks are not uniformly randomly wired; rather, they exhibit specific, non-uniform random connectivity patterns characterized by "like-to-like" wiring rules and distinct structural motifs across cortical layers. Translating these high-level connectomic principles, such as modularity, hierarchy, and non-random receptive field organization, into scalable, topological generative frameworks for designing the sparse structure of artificial networks remains an open challenge.

To address the gap for a flexible, principled framework for generating and testing dendritic topologies, we introduce a dendritic-inspired network science generative model for sparse topology design of bipartite layers in deep neural networks: the Dendritic Network Model (DNM). The novelty of DNM lies in the elaborated mechanism to model the non-uniform topological organization of the receptive fields on the input layers. The model's parametric approach enables the systematic exploration of the relationship between network structure and computational function. The DNM provides a principled method for generating sparse network initializations that can be integrated into modern deep learning frameworks. We demonstrate that this approach can improve performance over standard sparse initialization techniques and offers a powerful platform for exploring how structural constraints, inspired by biology, can lead to more efficient and capable artificial neural networks.

Our approach can be contrasted with other dendritic-inspired methodologies in the field. For instance, Li et al. (2020) experimentally demonstrated a fully integrated hardware network using memristor devices, where artificial dendrites provided non-linear integration and filtering to achieve highly efficient physical networks. Subsequently, Malakasis et al. (2023) utilized bio-realistic spiking neural networks to show how active dendrites combined with uniformly random synaptic turnover can optimize learning in binary classification scenarios. These works pave the way for recent advancements like the work of Chavlis & Poirazi (2025), which presents a dendritic-emulating model that reproduces the nonlinear integrative functions of dendrites. In their framework, a dendrite is mapped to a node within a tree-like subnetwork, creating a nonlinear computational component for larger networks. Distinct from these bio-mimetic approaches, the work of Kepner & Robinett (2019) is rooted in algebraic graph theory. They generate deterministic topologies (RadiX-Nets) using mixed-radix numeral systems and Kronecker products of adjacency submatrices. Their primary design goal is to ensure graph-theoretic properties such as constant expansion, path-connectedness, and symmetry. In contrast, DNM is a generative framework inspired by biological morphology. Rather than relying on algebraic products, DNM constructs connectivity via parametric distributions (spatial/non-spatial) of specific biological components: dendritic branches, localized receptive fields, and synapses. Our focus is on modeling a dendritic-inspired sparse network topology that allows for the emergence of modular and hierarchical structures rather than enforcing deterministic symmetry (Figure 1). Thus, while (Li et al., 2020; Chavlis & Poirazi, 2025) focus on the emulation of nonlinear processing in tree-like dendritic subnetwork structures, our work investigates topological dendritic-inspired principles that allows modelling the initial sparse connectivity organization from a network science perspective.

In this article, we describe the Dendritic Network Model in detail and analyse its topology and geometric characterization. We evaluate its effectiveness with extensive experiments across multiple architectures and tasks. To assess its basic functionality, we use it to initialize several static and dynamic sparse training (DST) methods on MLPs for image classification on the MNIST (LeCun et al., 2002), EMNIST (Cohen et al., 2017), Fashion MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky, 2009) datasets. The results show that DNM clearly outperforms other sparse initialization methods over all training models tested at 99% sparsity. Next, we extend the tests on Transformers (Vaswani et al., 2017a) for Machine Translation on the Multi30k en-de (Elliott et al., 2016), IWSLT14 en-de (Cettolo et al., 2014), and WMT17 en-de (Bojar et al., 2017) benchmarks. On this architecture, DNM outperforms all topological initialization methods at high sparsity levels. These findings underscore the potential of DNM in enabling highly efficient and effective network

initialization for large-scale sparse neural network training. By analyzing the best-performing DNM topologies, we can also gain insights into the relationship between network geometry, data structure, and model performance.

2 Related Works

2.1 Sparse Topological Initialization Methods

Dynamic sparse training (DST) trains a neural network with a sparse topology that evolves throughout the learning process. The initial arrangement of the connections is a critical aspect of this framework. This starting structure determines the initial pathways for information flow and acts as the foundational scaffold upon which the network learns and evolves. A well-designed initial topology can significantly improve a model's final performance and training efficiency, whereas a poor starting point can severely hinder its ability to learn effectively. The principal topological initialization approaches for dynamic sparse training are grounded in network science theory, where three basic generative models for monopartite sparse artificial complex networks are the Erdős-Rényi (ER) model (Erdős & Rényi, 1960), the Watts-Strogatz (WS) model (Watts & Strogatz, 1998), and the Barabási-Albert (BA) model (Barabási & Albert, 1999). Since the standard WS and BA models are not directly designed for bipartite networks, they were recently extended into their bipartite counterparts and termed as Bipartite Small-World (BSW) and Bipartite Scale-Free (BSF) (Zhang et al., 2024b), respectively. BSW generally outperforms BSF for dynamic sparse training (Zhang et al., 2024b). The Correlated Sparse Topological Initialization (CSTI) (Zhang et al., 2024a) is a feature-informed topological initialization method that considers the links with the strongest Pearson correlations between nodes and features in the input layer. SNIP (Lee et al., 2018) is a data-informed pruning method that identifies important connections based on their saliency scores, calculated using the gradients of the loss function with respect to the weights. Ramanujan graphs (Kalra et al.) are a class of sparse graphs that exhibit optimal spectral properties, making them suitable for initializing neural networks with desirable connectivity patterns. The Bipartite Receptive Field (BRF) network model (Zhang et al., 2025) generates networks with brain-like receptive field connectivity. This is the first attempt to mimic the structure of brain connections in a sparse network initialization model. Radix-Nets (Kepner & Robinett, 2019) offer a deterministic approach to "de novo" sparsity, utilizing mixed-radix numeral systems and the Kronecker product to construct topologies that ensure path-connectedness and symmetry while facilitating asymptotic sparsity. Finally, dendritic Artificial Neural Networks (dANNs) (Chavlis & Poirazi, 2025) introduce a bio-inspired architecture that mimics the structured connectivity and restricted input sampling of biological dendrites (e.g., using Local Receptive Fields). Unlike traditional approaches that strive for class specificity, this architecture fosters mixed-selective neuronal responses.

While the methods discussed above primarily focus on initializing layered or bipartite structures, a parallel line of research investigates Artificial Neural Networks (ANNs) with general complex topologies, unconstrained by multipartite restrictions. Monteiro et al. (2016) demonstrated that hybrid topologies combining scale-free and small-world properties, inspired by the C. elegans connectome, can significantly improve learning curves. Moving beyond manual architecture design, Xie et al. (2019) utilize random graph models (ER, BA, WS) to generate "randomly wired" networks that achieve competitive performance in image recognition. To facilitate the translation between arbitrary graph structures and neural models, Stier & Granitzer (2022) introduced the deepstruct framework. More recently, Boccato et al. (2024) provided a systematic comparison of these architectures, revealing that complex, non-layered topologies can outperform traditional Multilayer Perceptrons (MLPs) in high-difficulty tasks by potentially exploiting compositional sparsity.

3 The Dendritic Network Model

3.1 Biological Inspiration and Principles

The architecture of the Dendritic Network Model (DNM) is inspired by the structure of biological neurons. In the nervous system, neurons process information through complex, branching extensions called dendrites, which act as the primary receivers of synaptic signals. Inspired by this phenomenon, the DNM imposes a structured, dendrite-like organization on how output neurons connect

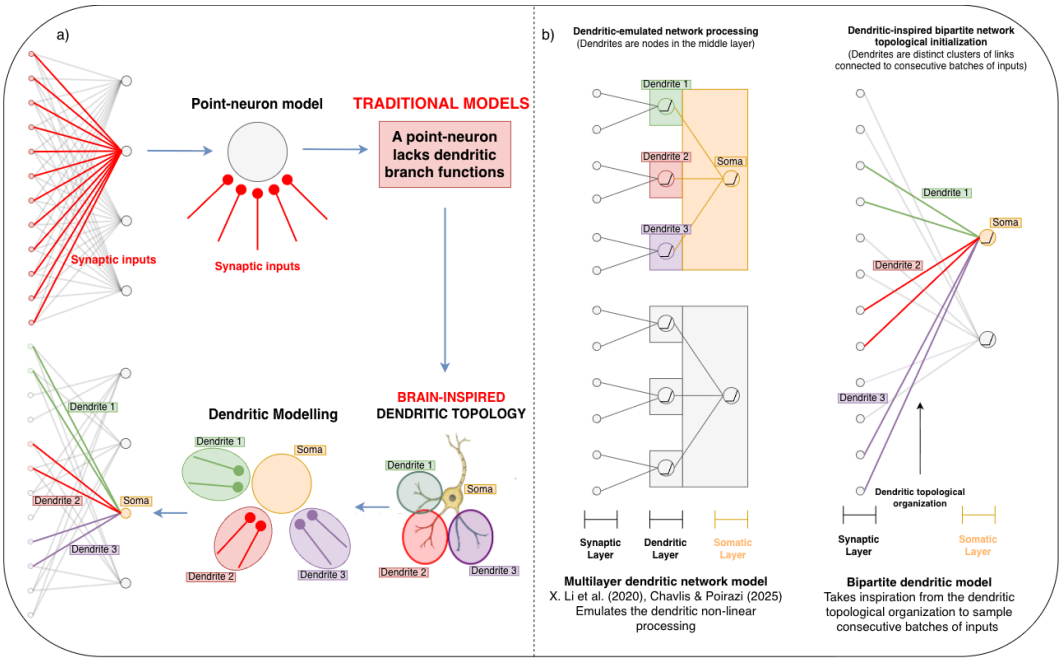


Figure 1: **Comparison of the Dendritic Network Model with traditional point-neurons and existing dendritic architectures. (a) From point-neurons to dendritic topology.** Traditional artificial neuron models (top) function as point-integrators, summing all synaptic inputs globally without spatial differentiation. In contrast, the DNM (bottom) introduces a brain-inspired topology where synaptic inputs are organized into distinct dendritic branches. This structure allows the output neuron to process inputs as clustered groups. **(b) Comparison with existing dendritic network models.** The left panel illustrates "Dendritic-emulated network processing" as seen in works like Li et al. (2020) and Chavlis & Poirazi (2025). In these architectures, dendrites are often modeled as explicit computational nodes forming an intermediate layer between inputs and the soma (a multilayer approach). The right panel illustrates the proposed DNM (a dendritic-inspired bipartite network topological initialization). Unlike previous dendritic-inspired models that have a tree-like multilayer structure, DNM embeds dendritic properties directly into the bipartite network topology. It treats dendrites as distinct clusters of links within a bipartite graph, connecting the soma to consecutive batches of inputs. This allows the network to inherit dendritic structural advantages through topological initialization rather than architectural expansion.

to the preceding layer's inputs (Figure 1). In this work, we refer to the set of connections between two adjacent layers as a bipartite sandwich layer, a term already used in prior literature to denote the bipartite subnetwork of edges that lies between one layer of neurons and the next. We retain this terminology because it captures the specific object our model generates, the pattern of connections, rather than the neurons themselves. By contrast, the term hidden layer refers to the neurons in intermediate layers. Since the DNM defines how neurons connect, but does not generate or modify the neurons themselves, "hidden layer" would not accurately describe the structural entity under consideration. Within each sandwich layer, each output neuron forms multiple dendritic branches, where each branch connects the neuron to a contiguous block of input neurons. These blocks are separated by inactive spaces, segments of the input layer to which the neuron does not connect. All branches belonging to a given output neuron must lie within a predefined local receptive window, resulting in a structured, compartmentalized connectivity pattern. This design moves away from unstructured random sparsity and instead emulates the localized, clustered organization characteristic of biological dendrites.

3.2 The DNM Generative Algorithm

To translate these biological principles into a computational structure, the DNM produces the sparse connectivity matrix of bipartite sandwich layers via a generative algorithm. The process builds connections iteratively for each output neuron j through the following steps: (1) **Degree determi-**

nation: first, determine the total degree for the output neuron based on a specific degree distribution strategy; (2) **Receptive field definition:** define the receptive field for the output neuron by topologically mapping the output neuron's position to a central point on the input layer and establishing a receptive window around this center; (3) **Dendritic allocation**: determine the number of dendritic branches used to connect the output neuron to the input layer; (4) **Dendritic Placement:** place evenly spread-out dendritic brances within the neuron's receptive window defined in step 2; (5) **Synaptic distribution:** distribute the output neuron's total degree across the dendrites. Appendix F describes the algorithm in depth.

3.3 Parametric Specification

By parametrizing the core biological features of DNM, like the number of dendrites for each output neuron, the size of the receptive windows, the distribution of synapses across dendrites, and the degree distribution across output neurons, the DNM provides a flexible framework for generating network topologies that are sparse, structured, and biologically plausible. Appendix C shows how the connectivity of an MLP is shaped by the DNM. To apply biological spatial principles to non-spatial MLPs, we index neurons $i \in \{1, \ldots, N\}$, and their physical location x_i is defined linearly such that the distance between adjacent indices is minimized. This allows us to define "spatial" distributions where connectivity probabilities depend on the relative distance $|x_i - x_j|$ between neurons in adjacent layers.

Sparsity (s) The sparsity parameter (s) defines the percentage of potential connections between the input and output layers that are absent, controlling the trade-off between computational cost and representational power.

Dendritic Distribution The dendritic distribution governs the number of branches that connect each output neuron to the input layer, which can be seen as the number of distinct input regions a neuron integrates information from. The central parameter for this is M, which defines the mean number of dendrites per output neuron. This distribution can be implemented in one of three ways. The simplest is a fixed distribution, where every output neuron has exactly M dendrites. Alternatively, a non-spatial distribution introduces stochasticity by sampling the number of dendrites for each output neuron from a probability distribution (e.g., Gaussian or uniform) with a mean of M. Finally, a spatial distribution makes the number of dendrites for each neuron dependent on its position within the layer. Using a Gaussian or inverted Gaussian profile, this configuration implies that some neurons integrate signals from many distinct input regions (a high dendrite count), while others connect to fewer, more focused regions (a low dendrite count).

Receptive Field Width Distribution The receptive field of an output neuron j is defined as the contiguous subset of input neurons to which j *potentially* connects. We define a mapping function $\phi(j)$ that projects the index of output neuron j to a center coordinate on the input layer. The receptive field is then the interval $[\phi(j) - \frac{W_j}{2}, \phi(j) + \frac{W_j}{2}]$, where W_j is the receptive field width determined by the parameter α. The receptive field mirrors the concept of receptive fields in biology. This process is governed by a mean parameter, α, which specifies the average percentage of consecutive input neurons on the input layer from which an output neuron can sample connections. This distribution can be configured in several ways: a fixed distribution assigns an identical window size α to all output neurons; a non-spatial distribution introduces variability by drawing each neuron's window size from a probability distribution (e.g., Gaussian or uniform) centered on α; and a spatial distribution links the window size to the neuron's position in its layer, allowing for configurations where receptive windows are, for instance, wider at the center and narrower at the edges.

Degree Distribution The degree distribution samples the number of incoming connections for each output neuron. This can be configured using a fixed distribution, where every output neuron is allocated the same degree. To introduce heterogeneity, a non-spatial distribution can be used to sample the degree for each neuron from a probability distribution. Finally, a spatial distribution allows the degree to vary based on the neuron's position, for instance, by creating highly connected, hub-like neurons at the center of the layer. The mean degree is set by the layer size and target sparsity.

Synaptic Distribution Once an output neuron's total degree is determined, the synaptic distribution allocates these connections among its various dendritic branches. The allocation can be fixed, where each dendrite receives an equal number of synapses. Alternatively, a non-spatial distribution can introduce random variability in synapse counts per dendrite. A spatial distribution can also be applied, making the number of synapses dependent on a dendrite's topological location, for example by assigning more connections to central branches versus outer ones. This distribution has a mean of $\frac{N_{in} \cdot (1-s)}{M}$, where N_{in} is the size of the input layer. While the degree distribution determines the total connectivity k_j of an output neuron j, the synaptic distribution governs the partition of k_j across the neuron's M_j dendritic branches. Formally, if $s_{j,b}$ is the number of synapses on the b-th dendritic branch of neuron j, the distribution ensures $\sum_{b=1}^{M_j} s_{j,b} = k_j$. This allocation can be uniform (fixed), stochastic (non-spatial), or topology-dependent (spatial), allowing specific branches, such as those in the center of the receptive field, to be more densely connected than distal branches.

Layer Border Wiring Pattern The DNM includes a setting to control how connections are handled at the boundaries of the input layer. The default behavior is a *wrap-around* topology, where the input layer is treated as a ring. This means a receptive window for a neuron near one edge can wrap around to connect to neurons on the opposite edge, ensuring all neurons have a similarly structured receptive field. Alternatively, a *bounded* pattern can be enforced. In this mode, receptive windows are strictly confined within the layer's physical boundaries. If a receptive field extends beyond the first or last input neuron, it is clamped to the edge. This enforces a more stringent locality, which we analyze further in Appendix I.

3.4 Network Topology and Geometric Characterization

A central hypothesis of this work is that specific topological features, such as modularity and hierarchy, confer distinct inductive biases that facilitate learning in ANNs. To test this hypothesis, it is essential to demonstrate that the DNM is not limited to a single structural configuration but rather functions as a flexible generative framework capable of accessing a diverse landscape of topologies. In this section, we systematically vary the hyperparameters defined in Section 3.3 to characterize this landscape. Our goal is to show that by tuning the DNM's parameters, we can controllably transition the network architecture across three distinct regimes: from unstructured uniformly connected random graphs, to input-order-dependent highly modular networks, and finally to input-order-dependent hierarchical, scale-free (Barabási & Albert, 1999) topologies. [1].

Figure 2 illustrates this topological diversity by comparing a baseline random network with several DNM configurations in a 3-layered MLP with 90% sparsity. Each panel displays the network's coalescent embedding (Cacciola et al., 2017) in hyperbolic space, its adjacency matrix, and network science metrics: characteristic path length (L), modularity (Q), structural consistency (σ_c), and the power-law exponent of the degree distribution (γ).

The coalescent embedding maps nodes onto a 2D hyperbolic disk, where the nodes' angular coordinates are computed via non-linear dimensionality reduction to cluster structurally similar nodes, while radial coordinates are derived from node popularity (degree). This visualization reveals **hierarchy** (nodes near the center act as hubs) and **modularity** (angular grouping indicates community structure). The full algorithmic details are provided in Appendix A.

The baseline random network (Figure 2a) lacks structure ($Q = 0.14$, $\sigma_c = 0.04$). Figure 2b represents a DNM network with $M = 3$, $\alpha = 1$, and all distributions fixed, yielding high modularity ($Q = 0.64$) and structural consistency ($\sigma_c = 0.76$). A key finding is that by setting a spatial Gaussian degree distribution (Figure 2c), the DNM generates a hierarchical topology that exhibits scale-free (Barabási & Albert, 1999) properties. Specifically, the resulting degree distribution follows a power law $P(k) \sim k^{-\gamma}$ with an exponent $\gamma = 2.30$. Since typical scale-free networks exhibit $2 < \gamma < 3$ (Barabási & Albert, 1999), this confirms that DNM can synthesize architectures with hub-like characteristics and hierarchical organization purely through parametric initialization. Similar measures are found when setting a spatial Gaussian synaptic distribution (Figure 2d, $Q = 0.54$, $\sigma_c = 0.74$),

[1] To facilitate an intuitive exploration of this landscape, we have developed an interactive web application where readers can adjust the model's parameters and visualize the resulting network structures. The application is available at: https: `https://dendritic-network-model.streamlit.app/`

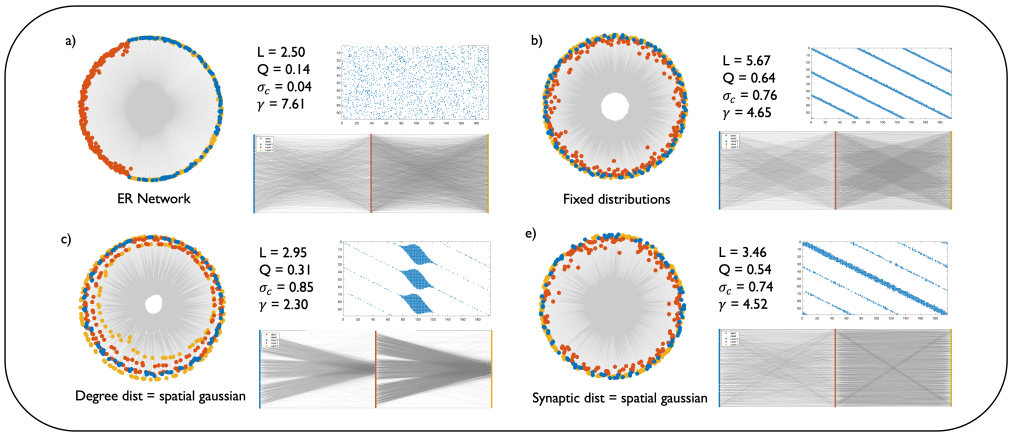


Figure 2: **Geometric and topological characterization of the Dendritic Network Model.** The figure compares a baseline random network (a) with various DNM configurations (b-d) for a 3-layered MLP of size $98 \times 196 \times 196$ with 90% sparsity. Each panel shows a coalescent embedding in hyperbolic space (left), the first layer's adjacency matrix (top right), a bipartite graph representation (bottom right), and key network science metrics: characteristic path length L, modularity (Q), structural consistency (σ_c), and the power law exponent of the degree distribution (γ). The network in (b) is a standard DNM model, generated using fixed distributions for all parameters, $M = 3$, and $\alpha = 1$. Panels (c-d) modify this standard configuration by switching a single parameter's distribution to spatial Gaussian: (c) degree distribution, (d) synaptic distribution.

because this configuration does not alter the structure of the network much, as highlighted by the adjacency matrix depicted.

This analysis shows that DNM is a highly flexible framework that can produce a wide spectrum of network architectures. This ability to controllably generate diverse network geometries is fundamental for analyzing the relationship between network structure and computational function in ANNs.

4 Experiments

4.1 Experimental Setup

We conduct experiments over two regimes: Static Sparse Training and Dynamic Sparse Training (DST).

Static Sparse Training We first evaluate DNM on Multilayer Perceptrons (MLPs) for image classification tasks. In this regime, the topology remains fixed after initialization to isolate the performance of initial sparse network.

Dynamic Sparse Training (DST) To validate the robustness of DNM as an initialization strategy for evolving topologies, we integrate it into state-of-the art DST frameworks. We select three DST methods that represent different landscapes of topological evolution mechanisms: SET (Mocanu et al., 2018) utilises random link regrowth; RigL (Evci et al., 2020) adopts gradient-based link regrowth; CHTs (Zhang et al., 2025) uses a network science-based Hebbian-inspired gradient-free link regrowth. A detailed description of these models is provided in Appendix H.2. By testing DNM on these fundamentally different regrowth strategies, we aim to prove that the benefits of the initialization are not limited to a specfic training paradigm. Finally, we apply DNM to Transformer (Vaswani et al., 2017b) models for machine translation tasks.

Implementation details For MLP training, we sparsify all layers except the final layer to prevent disconnected output neurons, noting that the final layer has relatively fewer connections compared to previous layers. Comprehensive parameter settings are detailed in Appendix D, and sensitivity tests on the DNM hyperparameters are provided in Appendix E.

Baseline methods We compare the performance of the DNM initialization against baseline topologies found in the literature. On static sparse training, we compare DNM with a randomly initialized network, the Bipartite Small World (BSW) (Zhang et al., 2024b), the Bipartite Receptive Field (BRF) (Zhang et al., 2025), the Ramanujan graph (Kalra et al.) initialization techniques, the RadiX-Nets (Kepner & Robinett, 2019) and dANN-R (Chavlis & Poirazi, 2025) [2], which proved to be the best-performing dANN variant over our tests. We also include CSTI (Zhang et al., 2024a) and SNIP (Lee et al., 2018) as the baseline models, noting that their comparison is inherently unfair due to their data-informed nature. For dynamic sparse training, we compare DNM with a random initialization, BRF, BSW, Ramanujan graph, RadiX-Nets, dANN, SNIP and CSTI. Finally, for tests on Transformer models, we compare DNM with the BRF initialization, which was proven to be the state-of-the-art sparse initialization method in previous studies (Zhang et al., 2025).

4.2 MLP FOR IMAGE CLASSIFICATION

Static sparse training As an initial evaluation of DNM's performance, we compare it to other topological initialization methods for static sparse training for image classification tasks. On all benchmarks, DNM outperforms the baseline models, as shown in Table 1. Analyzing the best-performing DNM networks is crucial to understanding the relationship between network topology and task performance. This aspect is assessed in Section 5.

Table 1: Image classification accuracy of statically trained, 99% sparse MLPs with different initial network topologies, compared to the fully-connected (FC) model. The scores are averaged over 3 seeds ± their standard errors. Bold values denote the best performance amongst initialization methods different from CSTI and SNIP.

	Static Sparse Training			
	MNIST	**Fashion MNIST**	**EMNIST**	**CIFAR10**
FC	98.80±0.00	90.87±0.02	87.08±0.04	62.35±0.13
CSTI	98.11±0.03	88.55±0.18	84.74±0.06	52.60±0.25
SNIP	98.03±0.03	88.65±0.07	85.19±0.04	61.89±0.48
Random	95.58±0.03	86.76±0.05	78.42±0.26	54.75±0.15
BSW	97.27±0.05	87.87±0.10	82.92±0.05	56.26±0.04
BRF	97.28±0.03	87.78±0.14	82.88±0.02	54.86±0.08
Ramanujan	96.39±0.10	86.44±0.14	81.78±0.08	54.61±0.32
RadiX-Nets	97.06±0.12	88.02±0.05	82.65±0.11	50.90±0.23
dANN-R	96.10±0.11	86.52±0.01	80.64±0.11	51.57±0.23
DNM	**98.07±0.09**	**88.86±0.21**	**85.63±0.10**	**58.71±0.28**

Dynamic sparse training We first test DNM on the baseline dynamic sparse training methods, SET and RigL. The results are shown and discussed in Appendix B, proving that DNM outperforms the other sparse initialization methods of MLPs (99% sparsity) over the datasets tested.Table 2 shows the result of the same tests on the state-of-the-art DST method, CHTs. Not only does DNM exhibit high performance for this task, but it can also surpass the input-informed CSTI method.

4.3 TRANSFORMER FOR MACHINE TRANSLATION

We assess the Transformer's performance on a machine translation task across three datasets. We take the best performance of the model on the validation set and report the BLEU score on the test set. Beam search, with a beam size of 2, is employed to optimize the evaluation process. On the Multi30k and IWSLT datasets, we conduct a thorough hyperparameter search to find the best settings for our DNM model. For the WMT dataset, in contrast, we simply use the best settings found in

[2]For fairness, to perform this comparison, we substituted each of the bipartite sandwich layers in our network with Chavlis and Poirazi's three-layered subnetwork of sizes x, $2x$, and x respectively, where x is the size of the input. Then, to compensate for the size difference between the two models, we initialized the dANNs in a way such that the number of connections between networks is the same, rather than their sparsities. However, we also provide tests on the original model published by Chavlis & Poirazi (2025) in Appendix J.

Table 2: Image classification on MNIST, Fashion MNIST, EMNIST, and CIFAR10 of the CHTs model on MLPs with 99% sparsity over various topological initialization methods, compared to the fully-connected (FC) model. The scores indicate the accuracy of the models, averaged over 3 seeds ± their standard errors. Bold values denote the best performance amongst initialization methods different from CSTI.

	CHTs			
	MNIST	**Fashion MNIST**	**EMNIST**	**CIFAR10**
FC	98.80±0.00	90.87±0.02	87.08±0.04	62.35±0.13
CSTI	98.70±0.04	90.56±0.09	87.47±0.04	69.59±0.20
Random	98.46±0.08	90.02±0.14	87.04±0.09	64.62±0.08
BSW	98.45±0.03	90.22±0.07	**87.14±0.03**	67.16±0.03
BRF	98.52±0.08	90.55±0.08	87.09±0.10	66.72±0.96
Ramanujan	98.37±0.04	89.78±0.12	86.82±0.09	64.57±0.10
RadiX-Nets	98.44±0.05	90.10±0.18	86.85±0.06	64.92±0.11
dANN-R	–±–	–±–	–±–	–±–
DNM	**98.59±0.03**	**90.57±0.10**	**87.14±0.09**	**68.52±0.03**

the previous tests. This approach helps to verify that DNM performs well even without extensive, dataset-specific tuning. DNM markedly improves the performance of the CHTs algorithm (Table 3, 4).

Table 3: Performance comparison of BRF and DNM initialization on Transformer models trained with CHTs on Multi30k en-de and IWSLT en-de translation tasks with varying sparsity levels (95% and 90%). BLEU scores (higher is better) are averaged over 3 seeds ± standard error. Bold indicates best performance for given sparsity and initialization.

	CHTs			
Initialization	Multi30k		IWSLT	
	0.95	0.90	0.95	0.90
FC	31.38±0.38		24.48±0.30	
BRF	28.94±0.57	29.81±0.37	21.15±0.10	21.92±0.17
DNM	**30.54±0.42**	**31.45±0.35**	**22.09±0.14**	**23.52±0.24**

Table 4: Performance comparison of BRF and DNM initialization on Transformer models trained with CHTs on machine translation tasks across the WMT en-de dataset with varying final sparsity levels (95% and 90%). Contrary to the BRF model, the DNM model's parameters were transferred from the best-performing combinations of previous tests, avoiding any parameter search. Entries are BLEU scores (higher is better), averaged over 3 seeds ± standard error. Bold values denote the best performance for a given sparsity and initialization.

	CHTs	
Initialization	WMT	
	0.95	0.90
FC	25.52	
BRF	20.94±0.63	22.40±0.06
DNM	**21.34±0.20**	**22.56±0.14**

5 Results Analysis

To understand which network structures are inherently best suited for specific tasks, we analyze the topologies of the top-performing models from our static sparse training experiments. Static sparse training is ideal for this analysis because its fixed topology allows us to link network structure to task

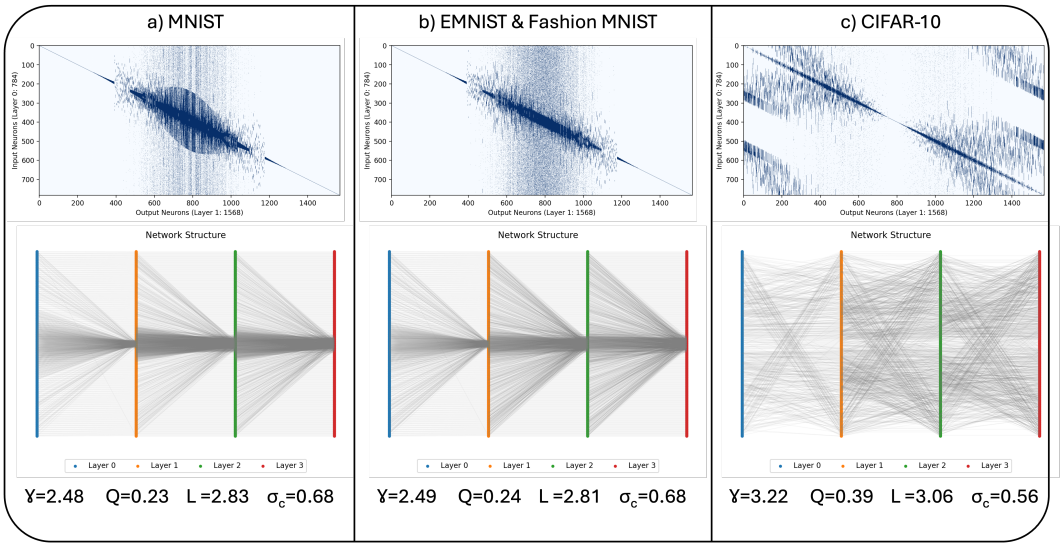


Figure 3: **Representation of the best performing DNM models on image classification.** The figure compares the best performing DNM architectures on MNIST (a), Fashion MNIST and EMNIST (b), and CIFAR10 (c). Each panel shows the network's adjacency matrix (top) and the network's layerwise representation (bottom). Furthermore, each panel exhibits the network's topological measures: characteristic path length L, modularity (Q), structural consistency (σ_c), and the power law exponent of the degree distribution (γ).

performance directly, isolating it as a variable in a way that is impossible with dynamic methods. Figure 3 shows the adjacency matrices of these models, their direct bipartite graph representations, and their key metrics in network science. For image classification on Fashion MNIST and EMNIST, the optimal network's topology is identical, and very similar to that on MNIST. These networks are scale-free ($\gamma \leq 3$) (Barabási & Albert, 1999) and exhibit a small characteristic path. Finally, we obtain contrasting results when assessing the network adopted for CIFAR-10 classification. Its higher γ parameter indicates that this network lacks hub nodes, possibly hinting that for more complex datasets like CIFAR-10, a more distributed and less hierarchical connectivity pattern is advantageous. Such a topology might promote the parallel processing of localized features across the input space, which is critical for natural image recognition, where object location and context vary significantly. In Appendix N, we expand our analysis by examining the topologies of the best-performing and worst-performing models on each of the dataset, and Appendix E gives a more detailed analysis of the best parameter combinations for each of the tests performed.

Overall, this analysis reveals a compelling relationship between task complexity and optimal network topology. While simpler, more structured datasets like MNIST and EMNIST benefit from scale-free, hierarchical architectures that can efficiently integrate global features through hub neurons, the more complex CIFAR-10 dataset favors a flatter, more distributed architecture. This underscores the potential of the DNM: its parametric flexibility allows it to generate these distinct, task-optimized topologies, moving beyond a one-size-fits-all approach to sparse initialization.

6 Conclusion

In this work, we introduced the Dendritic Network Model (DNM), a novel generative framework for initializing sparse neural networks inspired by the structure of biological dendrites. We have shown that the DNM is a highly flexible tool capable of producing a wide spectrum of network architectures, from modular to hierarchical and scale-free, by systematically adjusting its core parameters.

Our extensive experiments across multiple architectures demonstrate the effectiveness of our approach. At extreme sparsity levels, DNM consistently outperforms alternative topological initialization methods in both static and dynamic sparse training regimes, sometimes exceeding the performance of the data-informed CSTI and SNIP.

REFERENCES

Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, 2025.

Eunhye Baek, Sen Song, Chang-Ki Baek, Zhao Rong, Luping Shi, and Carlo Vittorio Cannistraci. Neuromorphic dendritic network computation with silent synapses for visual motion perception. *Nature Electronics*, 7(6):454–465, 2024.

Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. *arXiv preprint arXiv:1711.05136*, 2017.

Tommaso Boccato, Matteo Ferrante, Andrea Duggento, and Nicola Toschi. Beyond multilayer perceptrons: Investigating complex topologies in neural networks. *Neural Networks*, 171:215–228, 2024.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. Findings of the 2017 conference on machine translation (wmt17). Association for Computational Linguistics, 2017.

Alberto Cacciola, Alessandro Muscoloni, Vaibhav Narula, Alessandro Calamuneri, Salvatore Nigro, Emeran A Mayer, Jennifer S Labus, Giuseppe Anastasi, Aldo Quattrone, Angelo Quartarone, et al. Coalescent embedding in the hyperbolic space unsupervisedly discloses the hidden geometry of the brain. *arXiv preprint arXiv:1705.04192*, 2017.

Carlo Vittorio Cannistraci and Alessandro Muscoloni. Geometrical congruence, greedy navigability and myopic transfer in complex networks and brain connectomes. *Nature Communications*, 13 (1):7308, 2022.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th IWSLT evaluation campaign. In Marcello Federico, Sebastian Stüker, and François Yvon (eds.), *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pp. 2–17, Lake Tahoe, California, December 4-5 2014. URL https://aclanthology.org/2014.iwslt-evaluation.1.

Spyridon Chavlis and Panayiota Poirazi. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. *Nature communications*, 16(1):943, 2025.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Hermann Cuntz, Friedrich Forstner, Alexander Borst, and Michael Häusser. One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS computational biology*, 6(8):e1000877, 2010.

David A Drachman. Do we have brain to spare?, 2005.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pp. 70–74. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-3210. URL http://www.aclweb.org/anthology/W16-3210.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–60, 1960.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International conference on machine learning*, pp. 2943–2952. PMLR, 2020.

AC Frank, S Huang, M Zhou, A Gdalyahu, G Kastellakis, TK Silva, E Lu, X Wen, P Poirazi, JT Trachtenberg, et al. Hotspots of dendritic spine turnover facilitate clustered spine addition and learning and memory. nat commun 9: 422, 2018.

Ole-Christoffer Granmo. The tsetlin machine–a game theoretic bandit driven approach to optimal pattern recognition with propositional logic. *arXiv preprint arXiv:1804.01508*, 2018.

Siddhant Jayakumar, Razvan Pascanu, Jack Rae, Simon Osindero, and Erich Elsen. Top-kast: Top-k always sparse training. *Advances in Neural Information Processing Systems*, 33:20744–20754, 2020.

Ilenna Simone Jones and Konrad Paul Kording. Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree? *Neural Computation*, 33(6):1554–1571, 2021.

Ilenna Simone Jones and Konrad Paul Kording. Do biological constraints impair dendritic computation? *Neuroscience*, 489:262–274, 2022.

Suryam Arnav Kalra, Arindam Biswas, Pabitra Mitra, and Biswajit Basu. Sparse neural architectures via deterministic ramanujan graphs. *Transactions on Machine Learning Research*.

Jeremy Kepner and Ryan Robinett. Radix-net: Structured sparse matrices for deep neural networks. In *2019 IEEE international parallel and distributed processing symposium workshops (IPDPSW)*, pp. 268–274. IEEE, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

Mike Lasby, Anna Golubeva, Utku Evci, Mihai Nica, and Yani Ioannou. Dynamic sparse training with structured sparsity. *arXiv preprint arXiv:2305.02299*, 2023.

Clarissa Lauditi, Enrico M Malatesta, Fabrizio Pittorino, Carlo Baldassi, Nicolas Brunel, and Riccardo Zecchina. Impact of dendritic nonlinearities on the computational capabilities of neurons. *PRX Life*, 3(3):033003, 2025.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Xinyi Li, Jianshi Tang, Qingtian Zhang, Bin Gao, J Joshua Yang, Sen Song, Wei Wu, Wenqiang Zhang, Peng Yao, Ning Deng, et al. Power-efficient neural network with artificial dendrites. *Nature Nanotechnology*, 15(9):776–782, 2020.

Sean Lie. Harnessing the power of sparsity for large gpt ai models. *Cerebras Systems Blog*, 2022.

Michael London and Michael Häusser. Dendritic computation. *Annu. Rev. Neurosci.*, 28(1):503–532, 2005.

Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H Eugene Stanley. Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences*, 112(8): 2325–2330, 2015.

Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3): 261–277, 1988.

Nikos Malakasis, Spyridon Chavlis, and Panayiota Poirazi. Synaptic turnover promotes efficient learning in bio-realistic spiking neural networks. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 942–949. IEEE, 2023.

Adam Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. In *2013 IEEE 54th Annual Symposium on Foundations of computer science*, pp. 529–537. IEEE, 2013.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Roberto LS Monteiro, Tereza Kelly G Carneiro, José Roberto A Fontoura, Valéria L da Silva, Marcelo A Moret, and Hernane Borges de Barros Pereira. A model for improving the learning curves of artificial neural networks. *PloS one*, 11(2):e0149874, 2016.

Alessandro Muscoloni, Josephine Maria Thomas, Sara Ciucci, Ginestra Bianconi, and Carlo Vittorio Cannistraci. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature communications*, 8(1):1615, 2017.

Alessandro Muscoloni, Ilyes Abdelhamid, and Carlo Vittorio Cannistraci. Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more. *BioRxiv*, pp. 346916, 2018.

Alessandro Muscoloni, Umberto Michieli, Yingtao Zhang, and Carlo Vittorio Cannistraci. Adaptive network automata modelling of complex networks. 2022.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

Julian Stier and Michael Granitzer. Deepstruct–linking deep learning and graph theory. *Software Impacts*, 11:100193, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017a.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017b.

Christopher A Walsh. Peter huttenlocher (1931–2013). *Nature*, 502(7470):172–172, 2013.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 (6684):440–442, 1998.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1284–1293, 2019.

Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34:20838–20850, 2021.

Yingtao Zhang, Jialin Zhao, Wenjing Wu, and Alessandro 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*, 2024a.

Yingtao Zhang, Jialin Zhao, Wenjing Wu, Alessandro Muscoloni, and Carlo Vittorio Cannistraci. 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*, 2024b. URL https://openreview.net/forum?id=iayEcORsGd.

Yingtao Zhang, Diego Cerretti, Jialin Zhao, Wenjing Wu, Ziheng Liao, Umberto Michieli, and Carlo Vittorio Cannistraci. Brain network science modelling of sparse neural networks enables transformers and llms to perform as fully connected. *arXiv preprint arXiv:2501.19107*, 2025.

A GLOSSARY OF NETWORK SCIENCE

In this section, we introduce the basic notions of network science mentioned in this article.

Scale-Free Network A Scale-Free Network (Barabási & Albert, 1999) is characterized by a highly uneven distribution of degrees amongst the nodes. A small number of nodes, called hubs, have a very high degree, and a large number of nodes have very few connections. The degree distribution of scale-free networks follows a power law trend $P(k) \sim k^{\gamma}$, where γ is a constant smaller than 3. In contrast, nodes in random networks are distributed following a Binomial distribution.

Watts-Strogatz Model and Small-World Network A Small-World Network (Watts & Strogatz, 1998) is characterized by a small average path length. This property implies that any two nodes can communicate through a short chain of connections. The Watts-Strogatz (Watts & Strogatz, 1998) model is well known for its high clustering and short path lengths. This network is modelled by a parameter β between 0 and 1 that can determine its level of clustering. When β takes low values ($\beta \approx 0$), the WS network is a highly clustered lattice. On the other hand, when β approaches 1, the network becomes a random small-world graph. Intermediate values of β can generate a clustered network that maintains small-world connectivity.

Formally, a network is small-world when the path of length L between two randomly chosen nodes grows proportionally to the logarithm of the number of nodes (N) in the network, that is:

$$L \propto logN. \tag{1}$$

Structural Consistency Structural consistency (Lü et al., 2015) is an index based on the first-order matrix perturbation of the adjacency matrix, which represents the predictability of the network structure. A perturbation set ΔE is randomly sampled from the original link set E. Identifying as E^L the links ranked as the top L according to the structural perturbation method (Lü et al., 2015), with $L = |\Delta E|$, the structural consistency σ_c is calculated as:

$$\sigma_c = \frac{|E^L \cap \Delta E|}{\Delta E}. \tag{2}$$

Modularity Modularity (Newman, 2006) quantifies the tendency of nodes in the network to form distinct communities (or clusters). This measure ranges from -1 to 1. A high modularity score (close to 1) hints at the presence of dense connections between nodes within communities, but sparse connections between nodes belonging to different communities. A modularity score close to 0, in contrast, suggests that the network lacks any community organization and the interaction between nodes is essentially uniform. When modularity approaches -1, the network exhibits an anticommunity structure. This means that nodes are strongly connected across the network, and there is little differentiation into separate groups. In other words, a negative modularity represents a cohesive network. The formula to compute the modularity (Q) is:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{3}$$

where A represents the network's adjacency matrix, and k_i and k_j are the degrees of nodes i and j, respectively. $\delta(c_i, c_j)$ is the Kronecker delta function, which equals one if i and j are in the same community, else it equals 0.

Characteristic path length The characteristic path length is computed as the average node-pairs length in the network; it is a measure associated with the network's small-worldness (Cannistraci & Muscoloni, 2022). The characteristic path length (L) is derived by:

$$L = \frac{1}{n(n-1)} \cdot \sum_{i,j} d(i,j), \tag{4}$$

where n is the number of nodes in the network, and d(i,j) is the shortest path length between node i and node j.

Coalescent Embedding Coalescent embedding (Muscoloni et al., 2017) is a class of machine learning algorithms used for unsupervised dimensionality reduction and embedding complex networks in a geometric space, often hyperbolic. This method maps high-dimensional information on a low-dimensional embedding while maintaining the essential topological features of the network. This embedding reveals latent structures of the system, like hierarchical and scale-free structures. In this article, coalescent embedding maps the networks that have latent hyperbolic geometry onto the two-dimensional hyperbolic space. The approach involves 4 steps: 1) links are pre-weighted with topological rules that approximate the underlying network geometry (Cannistraci & Muscoloni, 2022); 2) non-linear dimensionality reduction; 3) generation of angular coordinates; 4) generation of radial coordinates.

This process is illustrated in Figure 2, which showcases the results of applying a specific coalescent embedding pipeline to four different synthetic networks. The embeddings shown were generated without any initial link pre-weighting (step 1). For the non-linear dimensionality reduction (step 2), the Isomap (Balasubramanian & Schwartz, 2002) algorithm was used. Finally, the angular coordinates (step 3) were determined using Equidistant Adjustment (EA), a process that preserves the relative order of the nodes while arranging them at perfectly uniform angular intervals.

B Experiments on baseline DST methods

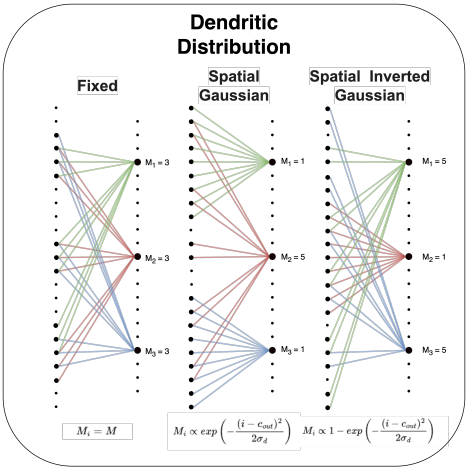
In this section, we provide the results of our experiments on the baseline dynamic sparse training methods, SET and RigL. The results are shown in Tables 5 and 6, proving that DNM outperforms the other sparse initialization methods of MLPs (99% sparsity) over the datasets tested. The model's performance is comparable to the input-informed CSTI and SNIP, highlighting that DNM's high degree of freedom can match a topology induced by data features.

Table 5: Image classification on MNIST, Fashion MNIST, EMNIST, and CIFAR-10 of the SET model on MLPs with 99% sparsity over various topological initialization methods, compared to the fully-connected (FC) model. The scores indicate the accuracy of the models, averaged over 3 seeds ± their standard errors. Bold values denote the best performance amongst initialization methods different from CSTI and SNIP.

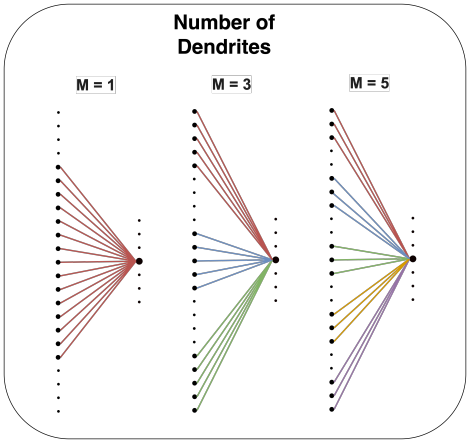
	SET			
	MNIST	**Fashion MNIST**	**EMNIST**	**CIFAR10**
FC	98.80±0.00	90.87±0.02	87.08±0.04	62.35±0.13
CSTI	98.40±0.06	89.96±0.07	86.70±0.10	65.31±0.16
SNIP	98.66±0.02	90.43±0.08	87.13±0.02	63.45±0.14
Random	98.16±0.06	89.17±0.15	86.03±0.12	62.80±0.24
BSW	98.22±0.03	89.28±0.09	86.21±0.02	64.13±0.11
BRF	98.56±0.03	89.58±0.11	86.21±0.11	64.40±0.25
Ramanujan	98.08±0.03	88.72±0.11	85.89±0.04	62.28±0.15
RadiX-Nets	98.37±0.08	89.33±0.08	86.15±0.09	55.91±0.13
dANN-R	97.95±0.10	88.91±0.02	85.47±0.09	57.44±0.09
DNM	**98.67±0.04**	**89.66±0.05**	**87.32±0.11**	**64.47±0.17**

C Impact of DNM parameters on Network Science

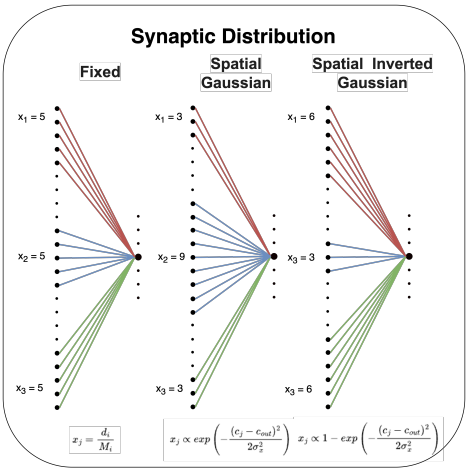
In this section, we provide a visual breakdown of how the topological structure of the Dendritic Neural Model (DNM) adapts when individual control parameters are varied. While the mathematical definitions of these distributions are provided in the main text, visualizing the resulting connectivity patterns offers greater insight into the network's plasticity and pruning capabilities. Figure 4 illustrates the effects of varying key DNM parameters on the resulting network topology. Each subfigure isolates a single parameter change while holding all others constant, allowing for a clear view of its specific influence.



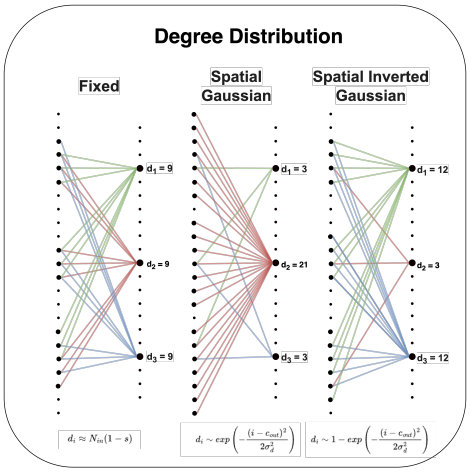
(a) Effect of the dendritic distribution on the network topology.



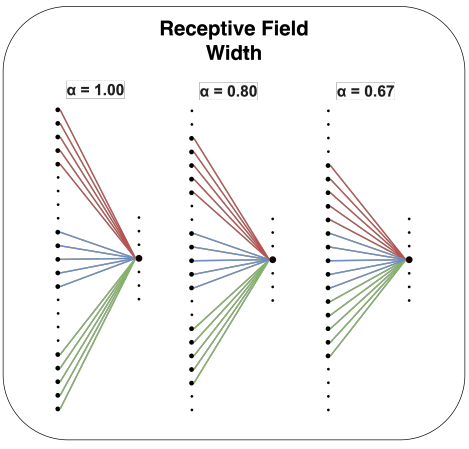
(b) Effect of the number of dendrites on the network topology.



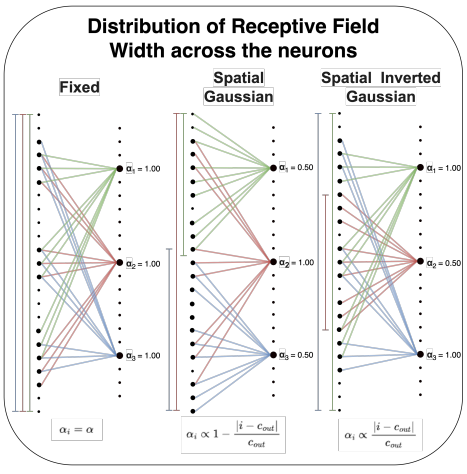
(c) Effect of the synaptic distribution on the network topology.



(d) Effect of the degree distribution on the network topology.



(e) Effect of the receptive field width on the network topology.

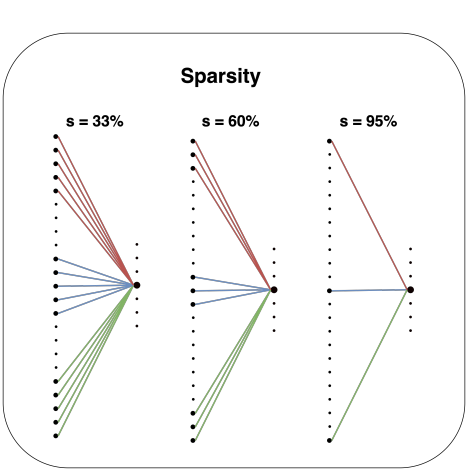


(f) Effect of the receptive field width distribution on the network topology.

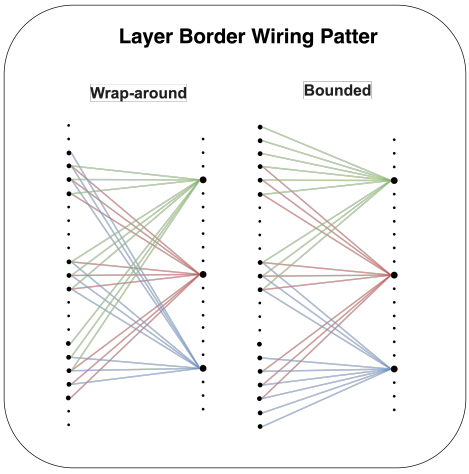
Figure 4: Representations of the network's topology obtained by varying a DNM parameter while keeping all others fixed.

Table 6: Image classification on MNIST, Fashion MNIST, EMNIST, and CIFAR-10 of the RigL model on MLPs with 99% sparsity over various topological initialization methods, compared to the fully-connected (FC) model. The scores indicate the accuracy of the models, averaged over 3 seeds ± their standard errors. Bold values denote the best performance amongst initialization methods different from CSTI and SNIP.

RigL				
	MNIST	**Fashion MNIST**	**EMNIST**	**CIFAR10**
FC	98.80±0.00	90.87±0.02	87.08±0.04	62.35±0.13
SNIP	98.76±0.05	90.50±0.06	87.30±0.04	63.31±0.25
CSTI	98.77±0.02	90.19±0.03	87.28±0.06	60.59±0.46
Random	98.66±0.27	89.88±0.04	87.18±0.07	64.13±0.11
BSW	**98.74±0.03**	90.12±0.06	87.28±0.10	65.19±0.23
BRF	98.18±0.03	89.79±0.02	87.05±0.14	63.55±0.78
Ramanujan	98.37±0.04	89.78±0.12	86.82±0.09	64.57±0.10
RadiX-Nets	98.44±0.05	90.10±0.18	86.85±0.06	64.57±0.10
dANN-R	98.54±0.05	89.44±0.05	86.81±0.04	62.03±0.06
DNM	**98.74±0.06**	**90.22±0.02**	**87.35±0.15**	**65.58±0.13**



(a) Effect of the dendritic distribution on the network topology.



(b) Effect of the layer border setting on the network topology.

Figure 5: Representations of the network's topology obtained by varying a DNM parameter while keeping all others fixed.

D Hyperparameter Settings and Implementation Details

Our experimental setup is designed to replicate the conditions in Zhang et al. (2025). Configurations are assessed on validation sets before being tested on separate test sets. All reported scores are the average of three runs using different random seeds, presented with their corresponding standard errors.

D.1 MLP for Image Classification

Models are trained for 100 epochs using Stochastic Gradient Descent (SGD) with a learning rate of 0.025, a batch size of 32, and a weight decay of 5×10^{-4} 7. All sparse models are trained at a 99% sparsity level. For dynamic methods, we used SET, RigL, and CHTs. The regrowth strategy for CHTs is CH2_L3n (Muscoloni et al., 2018). For our DNM, we conduct a grid search over its key hyperparameters. We tested a mean dendrite count (M) of 3. For the dendritic, degree, receptive field width, and synaptic distributions, we searched across fixed, spatial Gaussian, and

Table 7: Hyperparameters of MLP on Image Classification Tasks

Hyper-parameter	**MLP**
Hidden Dimension	1568 (3072 for CIFAR10)
# Hidden layers	3
Batch Size	32
Training Epochs	100
LR Decay Method	Linear
Start Learning Rate	0.025
End Learning Rate	$2.5e^{-4}$
ζ (fraction of removal)	0.3
Update interval (for DST)	1 epoch
Momentum	0.9
Weight decay	$5e^{-4}$

spatial inverted Gaussian options. The mean receptive field width (α) was fixed at 1.0. For the BSW baseline, the rewiring probability is searched in the set $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For the BRF baseline, we searched the randomness parameter r over the same set of values, and also tested both fixed and uniform degree distributions.

D.2 Transformer for Machine Translation

We use a standard 6-layer Transformer architecture with 8 attention heads and a model dimension of 512. The dimension of the feed-forward network is set to 1024 for Multi30k and 2048 for IWSLT14 and WMT17. All models are trained using the Adam optimizer (Kingma & Ba, 2014) with the noam learning rate schedule. Dataset-specific training parameters are as follows:

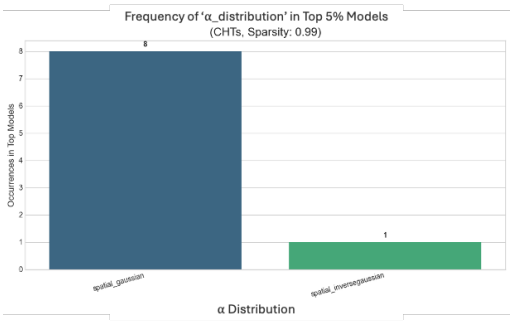
- Multi30k: Trained for 5,000 steps with a learning rate of 0.25, 1000 warmup steps, and a batch size of 1024.
- IWSLT14: Trained for 20,000 steps with a learning rate of 2.0, 6000 warmup steps, and a batch size of 10240.
- WMT17: Trained for 80,000 steps with a learning rate of 2.0, 8000 warmup steps, and a batch size of 12000.

We evaluated models at final sparsity levels of 90% and 95%. For Multi30k and IWSLT14, we performed a comprehensive hyperparameter search. The search for IWSLT14 included a mean dendrite count $M \in \{3, 7, 21\}$ and various combinations of fixed and spatial distributions for all DNM parameters. For WMT17, to assess generalization, we did not perform a new search. Instead, we directly applied the best-performing DNM configuration identified from the IWSLT14 experiments. This configuration used M=7 with a fixed dendritic distribution, alongside spatial Gaussian or inverse-Gaussian distributions for degree, receptive field width, and synapses.

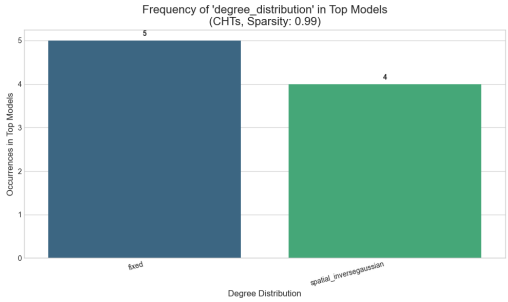
E Sensitivity Tests

We provide sensitivity tests for DNM hyperparameters. First, we focus on the analysis of CHTs on MLPs for image classification at 99% sparsity. Next, we study the parametric configurations for the CHTs model on the Multi30k translation task at 90% sparsity. For each task, we vary one parameter at a time, keeping the others fixed to a specific configuration. We calculate the coefficient of variation (CV) of the scores to quantify the sensitivity of the model to each parameter. A low CV indicates that the model's performance is relatively stable across different settings of that parameter, suggesting low sensitivity. Conversely, a high CV suggests that the model's performance is more variable and sensitive to changes in that parameter. We average each parameter's CV across various parametric configurations to obtain a robust measure of sensitivity. Next, we analyse the top 5% best-performing configurations for each task to understand the commonalities in the optimal settings. This method not only helps us understand which parameters are most influential but also guides future configurations for similar tasks.

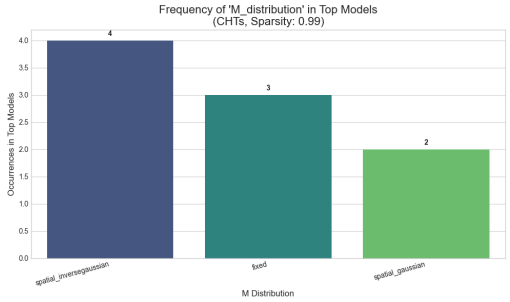
Dynamic Sparse Training for Image Classification We analyse the sensitivity of DNM parameters for the initialization of CHTs on MLPs for image classification at 99% sparsity. The analysis is performed over MNIST, Fashion MNIST, EMNIST, and CIFAR-10. The sensitivity analysis, summarized in Table 8, evaluates the impact of DNM initialization parameters for CHTs at 99% sparsity. Across all benchmarks, the Degree Distribution consistently emerges as the most critical parameter, highlighting the paramount importance of initial network connectivity. Following in descending order of influence are the Receptive Field Width, Dendritic, and Synaptic distributions. Analyzing the top 5% best-performing configurations, we observe similar trends across various datasets (Figure 6). The most relevant findings is that a spatial gaussian receptive field width distribution is constantly preferred.



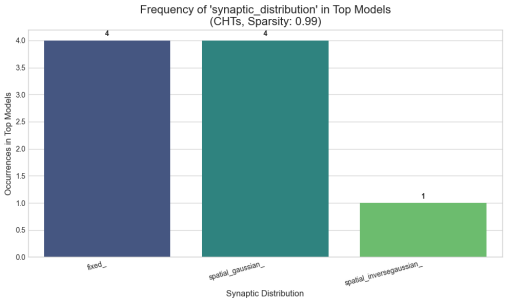
(a) Sensitivity analysis of the receptive field width distribution.



(b) Sensitivity analysis of the degree distribution.



(c) Sensitivity analysis of the dendritic distribution.



(d) Sensitivity analysis of the synaptic distribution.

Figure 6: Sensitivity analysis of the DNM parameters for CHTs initialization over MLPs for image classification on CIFAR-10 at 99% sparsity.

Table 8: Sensitivity Analysis Results for CHTs on MNIST, EMNIST, Fashion MNIST, CIFAR-10 for Image Classification. The table presents the average coefficient of variation (CV) for each DNM parameter across different configurations at 99% sparsity level. A higher CV indicates greater sensitivity of the model's performance to changes in that parameter.

	MNIST	**EMNIST**	**Fashion MNIST**	**CIFAR-10**
Degree Dist	0.001538	0.001419	0.003379	0.017639
Rec Field Width Dist	0.000506	0.001585	0.001205	0.005381
Dendritic Dist	0.000525	0.001265	0.001136	0.004983
Synaptic Dist	0.000464	0.000989	0.000964	0.004306

Dynamic Sparse Training for Machine Translation We focus on the analysis of DNM for the initialization of the CHTs model on Multi-30k for machine translation. Both at 90% and 95% sparsity, the calculated coefficients of variation do not surpass 0.01. This indicates that the model's performance is relatively stable across different settings of each parameter, suggesting low sensitivity (Table 9). At both sparsity levels, the dendritic distribution appears to be the most sensitive,

whereas M and α are the most stable. Analyzing the top 5% best-performing configurations, we observe that spatial distributions generally outperform fixed distributions (Figure 7).

Table 9: Sensitivity Analysis Results for CHTs on Multi30k for Machine Translation. The table presents the average coefficient of variation (CV) for each DNM parameter across different configurations at 90% and 95% sparsity levels. A higher CV indicates greater sensitivity of the model's performance to changes in that parameter.

Parameter	**Average Coefficient of Variation (CV)**	
	90%	**95%**
Dendritic distribution	0.008665	0.010520
Degree distribution	0.010910	0.014196
Receptive Field Width Distribution	0.008511	0.009607
Synaptic distribution	0.009444	0.011526
M	0.008821	0.009938
α	0.007576	0.009607

Note: Higher CV indicates greater impact on performance.

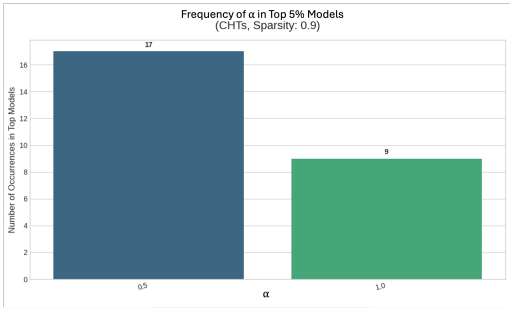
F Modelling Dendritic Networks

The Dendritic Network Model (DNM) generates the sparse connectivity matrix of bipartite sandwich layers by iteratively building connections for each output neuron based on the principles of dendritic branching and localized receptive fields. The generation process can be broken down into the following steps:

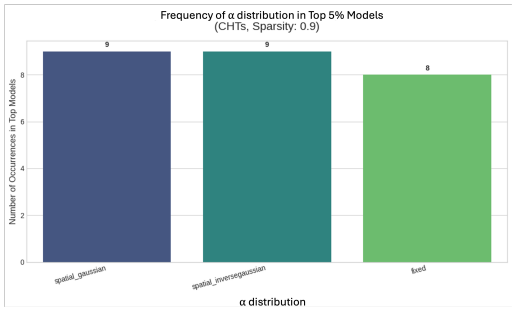
1. Determine the degree of each output neuron in the layer based on one of the three distribution strategies (fixed, non-spatial, spatial). A probabilistic rounding and adjustment mechanism ensures that no output neuron is disconnected and the sampled degrees sum precisely to the target total number of connections of the layer.
2. Next, for each output neuron j, define a receptive field. This is done by topologically mapping the output neuron's position at a central point in the input layer and establishing a receptive window around this center. The size of this window is controlled by the parameter $\alpha_j \in [0, 1]$, which determines the fraction of the input layer that the neuron can connect to. α_j itself can be fixed or sampled from a spatial or non-spatial distribution.
3. For each output neuron, determine the number of dendritic branches, M_j to be used to connect it to the input layer. Again, M_j is determined based on one of the three configurations (fixed for all neurons, or sampled from a distribution that could depend on the neuron's position in the layer).
4. Place the M_j dendrites as dendritic centers within the neuron's receptive window, spacing them evenly across the window.
5. The neuron's total degree, obtained from step 1, is distributed across its M_j dendrites according to a synaptic distribution (fixed, spatial, or non-spatial). For each dendrite, connections are formed with the input neurons that are spatially closest to its center.
6. Finally, the process ensures connection uniqueness and adherence to the precise degree constraints.

G Dynamic Sparse Training (DST)

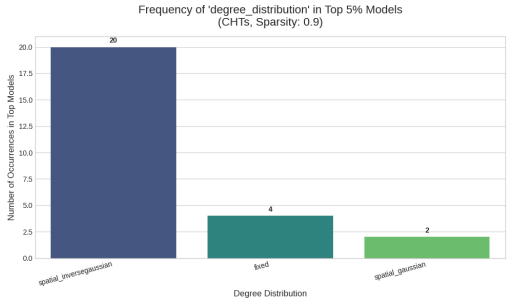
Dynamic sparse training (DST) is a subset of sparse training methodologies that allows for the evolution of the network's topology during training. Sparse Evolutionary Training (SET) (Mocanu et al., 2018) is the pioneering method in this field, which iteratively removes links based on the absolute magnitude of their weights and regrows new connections randomly. Subsequent developments have expanded upon this method by refining the pruning and regrowth steps. One such advancement was proposed by Deep R (Bellec et al., 2017), a method that evolves the network's topology based on stochastic gradient updates combined with a Bayesian-inspired update rule. RigL (Evci et al., 2020)



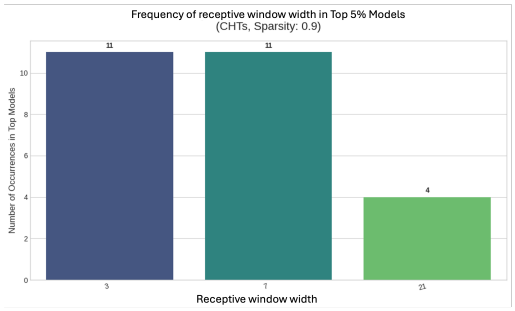
(a) Sensitivity analysis of α.



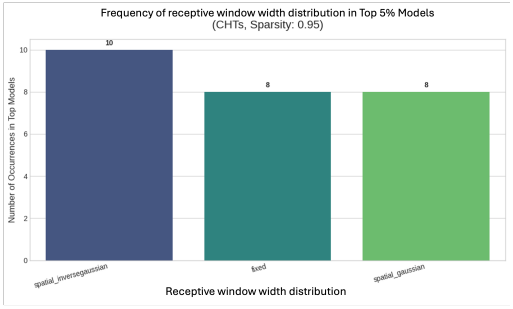
(b) Sensitivity analysis of the receptive field width distribution.



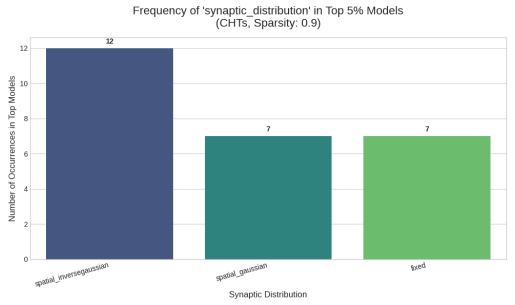
(c) Sensitivity analysis of the degree distribution.



(d) the receptive window width.



(e) Sensitivity analysis of the dendritic distribution.



(f) Sensitivity analysis of the synaptic distribution.

Figure 7: Sensitivity analysis of the DNM parameters for CHTs initialization over Transformer models for machine translation on Multi30k at 90% sparsity.

advanced the field further by leveraging the gradient information of non-existing links to guide the regrowth of new connections. MEST (Yuan et al., 2021) is a method that exploits information on the gradient and the weight magnitude to selectively remove and randomly regrow new links, similarly to SET. MEST introduces the EM&S technique that gradually decreases the density of the network until it reaches the desired sparsity level. Top-KAST (Jayakumar et al., 2020) maintains a constant sparsity level through training, iteratively selecting the top K weights based on their magnitude and applying gradients to a broader subset of parameters. To avoid the model being stuck in a suboptimal sparse subset, Top-KAST introduces an auxiliary exploration loss that encourages ongoing adaptation of the mask. A newer version of RigL, sRigL (Lasby et al., 2023), adapts the principles of the original model to semi-structured sparsity, speeding up the training from scratch of vision models. CHT (Zhang et al., 2024b) is the state-of-the-art (SOTA) dynamic sparse training framework that adopts a gradient-free regrowth strategy that relies solely on topological information (network shape intelligence). This model suffers from two main drawbacks: it has time complexity $\mathcal{O}(N \cdot d^3)$ (N node network size, d node degree), and it rigidly selects top link prediction scores, which causes suboptimal link removal and regrowth during the early stages of training. For this reason, this model

was evolved into CHTs (Zhang et al., 2025), which adopts a flexible strategy to sample connections to remove and regrow, and reduces the time complexity to $\mathcal{O}(N^3)$. The same authors propose a sigmoid-based gradual density decay strategy, namely CHTss (Zhang et al., 2025), which proves to be the state-of-the-art dynamic sparse training method over multiple tasks. CHTs and CHTss can surpass fully connected MLP, Transformer, and LLM models over various tasks using only a small fraction of the networks' connections.

H Baseline Methods

We describe in detail the models compared in our experiments.

H.1 Sparse Network Initialization

Bipartite Scale-Free (BSF) The Bipartite Scale-Free network (Zhang et al., 2024b) is an extension of the Barabási-Albert (BA) model (Barabási & Albert, 1999) to bipartite networks. We detail the steps to generate the network. 1) Generate a BA monopartite network consisting of $m+n$ nodes, where m and n are the numbers of nodes of the first and second layer of the bipartite network, respectively. 2) Randomly select m and n nodes to assign to the two layers. 3) Count the number of connections between nodes within the same layer (frustrations). If the two layers have an equal number of frustrations, match each node in layer 1 with a frustration to a node in layer 2 with a frustration, randomly. Apply a single rewiring step using the Maslov-Sneppen randomization (MS) procedure for every matched pair. If the first layer counts more frustrations, randomly sample a subset of layer 1 with the same number of frustrations, and repeat step 1. For each remaining frustration in layer 1, sequentially rewire the connections to the opposite layer using the preferential attachment method from step 1. If the second layer has more frustrations than the first, apply the opposite procedure. The resulting network will be bipartite and exhibit a power-law distribution with exponent $\gamma = 2.76$.

Bipartite Small-World (BSW) The Bipartite Small-World network (Zhang et al., 2024b) is an extension of the Watts-Strogatz model to bipartite networks. It is modelled as follows: 1) Build a regular ring lattice with a number of nodes $N = \#L_1 + \#L_2$, with $L_1 > L_2$, where L_1 and L_2 represent the nodes in the first and second layers of the bipartite network, respectively. 2) Label the N nodes in a way such that for every L_1 node positioned in the network, $\#L_1/\#L_2$ nodes from L_2 are placed at each step. Then, at each step, establish a connection between an L_1 node and the $K/2$ closest L_2 neighbours in the ring lattice. 3) For every node, take every edge connecting it to its $K/2$ rightmost neighbors, and rewire it with a probability β, avoiding self-loops and link duplication. When $\beta = 1$, the generated network corresponds to a random graph.

Correlated Sparse Topological Initialization (CSTI) The Correlated Sparse Topological Initialization (CSTI) (Zhang et al., 2024b) initializes the topology of the layers that interact directly with the input features. The construction of CSTI follows four steps. 1) **Vectorization:** Denoting as n the number of randomly sampled input data from the training set and as M the number of valid features with variance different from zero among these samples, we build an $n \times M$ matrix. 2) **Feature selection:** We perform feature selection by calculating the Pearson Correlation for each feature. Hence, we construct a correlation matrix. 3) **Connectivity selection:** Next, we construct a sparse adjacency matrix, with entries "1" corresponding to the top-k% values from the correlation matrix (where the value of k depends on the desired sparsity level). A scaling factor $\times$ determines the dimension of the hidden layer. 4) **Assembling topologically hubbed network blocks:** Finally, the selected adjacency matrix masks the network to form the initialized topology for each bipartite sandwich layer.

SNIP SNIP (Lee et al., 2018) is a static sparse initialization method that prunes connections based on their sensitivity to the loss function. The sensitivity of a connection is defined as the absolute value of the product of its weight and the gradient of the loss with respect to that weight, evaluated on a small batch of training data. Connections with the lowest sensitivity are pruned until the desired sparsity level is reached. This method allows for the identification of important connections before training begins, enabling the training of sparse networks from scratch.

Ramanujan Graphs Ramanujan Graphs (Lubotzky et al., 1988) are a class of optimal expander graphs that exhibit excellent connectivity properties. They are characterized by their spectral gap, which is the difference between the largest and second-largest eigenvalues of their adjacency matrix. A larger spectral gap indicates better expansion properties, meaning that the graph is highly connected and has a small diameter. Ramanujan graphs are constructed using deep mathematical principles from number theory and algebraic geometry. They are known for their optimal expansion properties, making them ideal for applications in network design, error-correcting codes, and computer science. In this article, we use bipartite Ramanujan graphs as a sparse initialization method for neural networks.

In our experiments, we built bipartite Ramanujan graphs as a theoretically-grounded initialization method, following the core principles outlined by Kalra et al.. Drawing inspiration from the findings of Marcus et al. (2013), which prove the existence of bipartite Ramanujan graphs for all degrees and sizes, we constructed these graphs as follows:

- Generate d random permutation matrices, where d is the desired degree of the graph. Each permutation matrix represents a perfect matching, which is a set of edges that connects each node in one layer to exactly one node in the other layer without any overlaps.
- Iteratively combine these matchings. In each step, deterministically decide whether to add or subtract the successive matching to the current adjacency matrix. This decision is made by minimizing a *barrier function* that ensures that the eigenvalues remain within the Ramanujan bounds.

Bipartite Receptive Field (BRF) The Bipartite Receptive Field (BRF) Zhang et al. (2025) is the first sparse topological initialization model that generates brain-network-like receptive field connectivity. The BRF directly generates sparse adjacency matrices with a customized level of spatial-dependent randomness according to a parameter $r \in [0, 1]$. A low value of r leads to less clustered topologies. As r increases towards 1, the connectivity patterns tend to be generated uniformly at random. Specifically, when r tends to 0, BRF builds adjacency matrices with links near the diagonal (adjacent nodes from the two layers are linked), whereas when r increases 1, this structure tends to break.

Mathematically, consider an $N \times M$ bipartite adjacency matrix $M_{i,j_{i=1,\ldots,M,\ j=1,\ldots,N}}$, where M represents the input size and n represents the output size. Each entry $m_{i,j}$ of the matrix is set to 1 if node i from the input layer connects to node j of the output layer, and 0 otherwise. Define the scoring function

$$S_{i,j} = d_{ij}^{\frac{1}{1-r}}, \tag{5}$$

where

$$d_{ij} = min\{|i - j|, |(i - M) - j|, |i - (j - N)|\} \tag{6}$$

is the distance between the input and output neurons. S_{ij} represents the distance of an entry of the adjacency matrix from the diagonal, raised to the power of $\frac{1-r}{r}$. When the parameter r tends to zero, the scoring function becomes more deterministic; when it tends to 1, all scores $S_{i,j}$ become more uniform, leading to a more random adjacency matrix.

The model is enriched by the introduction of the degree distribution parameter. The Bipartite Receptive Field with fixed sampling (BRFf) sets the degree of all output neurons to be fixed to a constant value. The Bipartite Receptive Field with uniform sampling, on the other hand, samples the degrees of output nodes from a uniform distribution.

RadiX-Nets RadiX-Nets (Kepner & Robinett, 2019) represent a family of sparse deep neural network topologies initialized "de novo," meaning they are constructed with a sparse structure from the outset rather than being pruned from a dense parent network. This approach addresses the limitations of hardware capacity by deterministically generating topologies that are much more diverse than previous sparse implementations (such as X-Nets) while preserving critical graph-theoretic properties. The construction relies on mixed-radix numeral systems and the Kronecker product to achieve connectivity.Formally, a mixed-radix topology is defined by an ordered set of integers

$$\mathcal{N} = (N_1, N_2, \ldots, N_L), \tag{7}$$

which implicitly defines a numeral system representing integers in the range 0 to $N' - 1$, where $N' = \prod_{i=1}^{L} N_i$. he final RadiX-Net topology is constructed via the Kronecker product of adjacency submatrices from these mixed-radix systems and dense adjacency submatrices. This construction guarantees two essential properties:

- Path-connectedness: There exists at least one path between any input and output node, ensuring that information can flow through the network.
- Symmetry: Each input node has the same number of paths to each output node, promoting uniformity in information distribution.

For a RadiX-Net defined by a mean radi μ and d radices (with $d \sim log_{\mu}(N')$), the density of the graph scales asymptotically as $\frac{1}{\mu^{d-1}}$, allowing for significant sparsity while maintaining expressive power.

dANNs Dendritic Artificial Neural Networks (dANNs) (Chavlis & Poirazi, 2025) incorporate the structural connectivity and restricted sampling properties of biological dendrites to improve parameter efficiency and robustness9. Unlike the point-neuron model of standard ANNs, a dANN splits the computational unit into two layers: a dendritic layer and a somatic layer. Input data is first fed into the dendritic layer via sparse connections (synaptic weights), where each dendrite performs a linear weighted sum followed by a nonlinearity. These dendritic activations are then weighted (cable weights) and summed at the soma before passing through a second nonlinearity.A defining feature of dANNs is their restricted input sampling, inspired by the receptive fields (RFs) of the visual cortex12. The authors propose three specific sampling strategies:

- Random Sampling (dANN-R): Input features are selected randomly for each dendrite.
- Local Receptive Fields (dANN-LRF): Inputs are sampled from a spatially restricted neighborhood (e.g., a 4×4 pixel grid), preserving spatial locality.
- Global Receptive Fields (dANN-GRF): Sampling is restricted per soma, but dendrites belonging to that soma sample from distributed locations around a central point.

The dANN models exhibit mixed selectivity, where nodes respond to multiple classes. This property results in networks that are more robust to noise and overfitting, matching or outperforming dense networks while using orders of magnitude fewer trainable parameters.

H.2 Dynamic Sparse Training (DST)

SET (Mocanu et al., 2018) At each training step, SET removes connections based on weight magnitude and randomly regrows new links.

RigL (Evci et al., 2020) At each training step, RigL removes connections based on weight magnitude and regrows new links based on gradient information.

CHTs (Zhang et al., 2025) Cannistraci-Hebb Training (CHT)(Muscoloni et al., 2022) is a brain-inspired gradient-free link regrowth method. It predicts the existence and the likelihood of each nonobserved link in a network. The rationale is that in complex networks that have a local-community structure, nodes within the same community tend to activate simultaneously ("fire together"). This co-activation encourages them to form new connections among themselves ("wire together") because they are topologically isolated. This isolation, caused by minimizing links to outside the community, creates a barrier that reinforces internal signaling. This strengthened signaling, in turn, promotes the creation of new internal links, facilitating learning and plasticity within the community. CHTs enhances this gradient-free regrowth method by incorporating a soft sampling rule and a node-based link-prediction mechanism.

I Analysis of the Bounded Layer Border Wiring Pattern

To investigate the impact of the layer border wiring pattern setting, we compare the default wrap-around topology with the bounded topology. We tested the bounded configuration on the same

image classification tasks as the default, at 99% sparsity using static sparse training (Table 10), SET (Table 11), and CHTs (Table 12). Overall, the performance of the bounded model is comparable to that of the default wrap-around model, with some variations across datasets and training methods. The bounded model outperforms the default on Fashion MNIST and CIFAR10 when using static sparse training and SET, while the default has a slight edge on MNIST and EMNIST. For CHTs, the results are mixed, with each model excelling in different datasets. These findings suggest that while strict locality can be beneficial in certain contexts, the flexibility of wrap-around connections may provide advantages in others. The choice of wiring pattern should thus be informed by the specific characteristics of the task and dataset at hand.

Table 10: Comparison of wrap-around and bounded topology performances for static sparse training of MLPs. The entries represent the accuracy for image classification over different datasets at 99% sparsity, averaged over 3 seeds ± their standard error.

Static Sparse Training				
	MNIST	**EMNIST**	**Fashion_MNIST**	**CIFAR10**
Bounded	97.64±0.10	84.00±0.06	**89.19±0.01**	**61.63±0.18**
Wrap-around	**97.82±0.03**	**84.76±0.13**	88.47±0.03	59.04±0.17

Table 11: Comparison of wrap-around and bounded topology performances for SET initialization on MLPs. The entries represent the accuracy for image classification over different datasets at 99% sparsity, averaged over 3 seeds ± their standard error.

SET				
	MNIST	**EMNIST**	**Fashion_MNIST**	**CIFAR10**
Bounded	**98.40±0.02**	**86.52±0.02**	**89.78±0.09**	**65.67±0.18**
Wrap-around	98.36±0.05	86.50±0.06	89.75±0.04	64.81±0.01

Table 12: Comparison of wrap-around and bounded topology performances for CHTs initialization on MLPs. The entries represent the accuracy for image classification over different datasets at 99% sparsity, averaged over 3 seeds ± their standard error.

CHTs				
	MNIST	**EMNIST**	**Fashion_MNIST**	**CIFAR10**
Bounded	**98.66±0.03**	87.35±0.00	**90.68±0.09**	68.03±0.14
Wrap-around	98.62±0.01	**87.40±0.04**	90.62±0.16	**68.76±0.11**

J Comparison of DNM with other dendritic networks

Previous work, like that of Chavlis & Poirazi (2025), has explored the integration of dendritic structures into artificial neural networks, demonstrating improvements in parameter efficiency and robustness. Our Dendritic Network Model (DNM) distinguishes itself through its comprehensive approach to modeling dendritic connectivity. While Chavlis & Poirazi (2025) primarily focuses on the functional aspects of dendrites within a tree-like multilayered structure (dendritic and somatic layers), DNM embeds dendritic properties directly into the bipartite network topology. DNM treats dendrites as distinct clusters of links within a bipartite graph, connecting the soma to consecutive batches of inputs. This allows the network to inherit dendritic structural advantages through topological initialization rather than architectural expansion 1.

Section 4.2 presents a direct comparison between DNM and the best performing dANN model. The results indicate that DNM consistently outperforms dANNs across various image classification tasks at extreme sparsity levels (99%). These tests were performed by substituting each bipartite

sandwich layer in our original network with dANN's three-layered subnetwork, ensuring that the total number of trainable parameters remained constant for a fair comparison. We attribute our model's superior performance to a structural limitation in the dANN framework: the inclusion of dendritic integration layers increases network depth, which in turn forces each bipartite sandwich layer to be significantly sparser. For completeness, we perform additional tests on the network originally proposed by Chavlis & Poirazi (2025). In particular, we take a network instantiated with three sequential dendritic blocks, following the structure:

$$Input \rightarrow (Dendrite \rightarrow Soma) \times 3 \rightarrow Output.$$

We set the width of the dendritic layers to twice the size of the input layer ($N_{dendrite} = 2 \times N_{in}$). Conversely, the somatic integration layers are fixed at a width $N_{soma} = 256$, and the synapses parameter is set to 128. We compare the three models introduced by Chavlis & Poirazi (2025), dANN-R, dANN-LRF, and dANN-GRF, against a modified dANN in which we replace the connectivity to the dendritic layers with a DNM topology that maintains the same sparsity levels. The parameters chosen for the DNM topology were extracted from the best performing configuration found in Section 4.2 for EMNSIT at 99% sparsity. The results, summarized in Table 13, indicate that the DNM-initialized model outperforms dANN, which is the model that we found to perform the best out of those proposed by Chavlis & Poirazi (2025). This further underscores the effectiveness of DNM's topological approach in enhancing network performance.

Table 13: Comparison of the dANN models (Chavlis & Poirazi, 2025) with a modified version where DNM replaces the connectivity patterns in the dendritic layer (dANN-DNM).

Model	MNIST	EMNIST	Fashion MNIST
dANN-R	98.49±0.05	85.96±0.09	89.77±0.08
dANN-LRF	98.51±0.00	85.60±0.15	89.39±0.07
dANN-GRF	98.53±0.06	86.26±0.05	89.77±0.08
dANN-DNM	**98.70±0.05**	**86.64±0.07**	**90.09±0.04**

K Transferability of Optimal Topologies

A critical question for our generative model of network topology is whether its principles are generalizable across different tasks. To investigate this, we conducted a comprehensive transfer learning experiment to assess if an optimal topology discovered on one task could be effectively applied to others. This tests the hypothesis that the DNM can capture fundamental structural priors beneficial for a class of problems, thereby reducing the need for extensive hyperparameter searches on every new dataset.

Experimental Design We identified the best-performing DNM hyperparameter configuration from the static sparse training experiments for each of the four datasets: MNIST, Fashion MNIST, EMNIST, and CIFAR-10. We then performed a cross-transfer analysis where the optimal configuration for a source dataset was used to initialize MLP models for the other three target datasets. We compared these transferred topologies against baseline initialization methods and against the DNM models specifically tuned for the target task ("DNM (Fine-tuned)").

Results The results, summarized in Table 14, reveal two distinct trends in topological transferability. First, we observe high transferability among the simpler grayscale datasets. Regardless of whether the parameters are transferred from MNIST, EMNIST, or Fashion MNIST, the resulting DNM models consistently outperform the baseline initialization methods on the other grayscale targets. In these cases, the transferred performance is often comparable to the task-specific fine-tuned models.

However, a significant performance gap emerges when transferring topologies from simpler tasks to the more complex CIFAR-10 dataset. As shown in Table 14, configurations optimized for MNIST, EMNIST, or Fashion MNIST yield poor performance when applied to CIFAR-10 (approximately 47-48% accuracy compared to 58.71% for the fine-tuned model). This suggests that the structural

priors learned from simpler inputs are insufficient for the complexity of natural images. Consequently, for practical scenarios involving image classification with MLPs, we recommend utilizing the parametric configuration derived from CIFAR-10. This configuration acts as a more robust baseline for complex tasks. The specific parameters for this recommended configuration are detailed in Table 15.

Table 14: Performance of transferred DNM topologies on static sparse training at 99% sparsity. We evaluate the transferability of the single best hyperparameter configuration found on each source dataset applied to the other targets. Scores represent accuracy averaged over 3 seeds ± standard error. Bold values denote the best performance among transfer and baseline methods.

	Static Sparse Training			
	MNIST	**Fashion MNIST**	**EMNIST**	**CIFAR10**
FC	98.80 ± 0.00	90.87 ± 0.02	87.08 ± 0.04	62.35 ± 0.13
CSTI	98.11 ± 0.03	88.55 ± 0.18	84.74 ± 0.06	52.60 ± 0.25
SNIP	98.03 ± 0.03	88.65 ± 0.07	85.19 ± 0.04	61.89 ± 0.48
Random	95.58 ± 0.03	86.76 ± 0.05	78.42 ± 0.26	54.75 ± 0.15
BSW	97.27 ± 0.05	87.87 ± 0.10	82.92 ± 0.05	56.26 ± 0.04
BRF	97.28 ± 0.03	87.78 ± 0.14	82.88 ± 0.02	54.86 ± 0.08
Ramanujan	96.39 ± 0.10	86.44 ± 0.14	81.78 ± 0.08	54.61 ± 0.32
RadiX-Nets	97.06 ± 0.12	88.02 ± 0.05	82.65 ± 0.11	50.90 ± 0.23
dANN-R	96.10 ± 0.11	86.52 ± 0.01	80.64 ± 0.11	51.57 ± 0.23
DNM (Fine-tuned)	98.07 ± 0.09	88.86 ± 0.21	85.63 ± 0.10	58.71 ± 0.28
Transferred from MNIST	98.07 ± 0.09	88.92 ± 0.09	85.80 ± 0.04	48.24 ± 0.25
Transferred from Fashion MNIST	98.05 ± 0.09	88.86 ± 0.21	85.63 ± 0.10	47.52 ± 0.39
Transferred from EMNIST	98.05 ± 0.09	88.86 ± 0.21	85.63 ± 0.10	47.52 ± 0.39
Transferred from CIFAR-10	97.48 ± 0.38	83.48 ± 0.02	88.62 ± 0.11	58.71 ± 0.28

Discussion This experiment strongly suggests that the structural principles identified by DNM as optimal for MNIST serve as a powerful and generalizable prior for other image classification tasks. The ability to transfer a high-performing topology with minimal performance loss has significant practical implications, as it can drastically reduce the computational cost associated with architecture search for new applications. This finding reinforces the idea that bio-inspired, structured initialization is not merely a task-specific trick but a robust strategy for building efficient sparse networks.

L LIMITATIONS

While the Dendritic Network Model demonstrates significant improvements in accuracyat extreme sparsity levels, we acknowledge a practical limitation regarding current hardware acceleration. Most contemporary GPUs are highly optimized for dense matrix operations; consequently, unstructured sparsity does not always translate into wall-clock training speedups without specialized software support. However, the hardware landscape is rapidly evolving to address this bottleneck. Emerging technologies are specifically designed to efficiently support unstructured sparsity (Lie, 2022). Our method is designed to be future-proof, positioned to fully leverage dynamic sparse training and inference as this specialized hardware becomes widely accessible.

M PRACTICAL GUIDELINES FOR DNM INITIALIZATION

To facilitate the adoption of the Dendritic Network Model (DNM) without the need for extensive hyperparameter search, we provide recommended baseline configurations for MLPs and Transformers. These settings are derived from our most robust experimental results: the CIFAR-10 configuration for image classification and the IWSLT14-to-WMT17 transferred configuration for machine translation.

M.1 MLP FOR IMAGE CLASSIFICATION

For Multi-Layer Perceptrons (MLPs) applied to image classification tasks, particularly those involving complex visual features, we recommend the configuration detailed in Table 15. This setup was found to be the most effective for the CIFAR-10 dataset, and proved to stably outperform other initialization baselines on diverse datasets (K).

Table 15: Recommended DNM parameter configuration for practical image classification scenarios. This configuration corresponds to the optimal settings identified for CIFAR-10.

Parameter	**Value / Distribution**
Sparsity (s)	0.99
Mean Dendrites	3
Dendritic Distribution	Fixed
Receptive window width	1.0
Receptive window width distribution	Spatial Inverse-Gaussian
Degree Distribution	Spatial Inverse-Gaussian
Synaptic Distribution	Spatial Inverse-Gaussian
Layer Border Wiring Pattern	Wrap-around

M.2 TRANSFORMERS FOR MACHINE TRANSLATION

For Transformer architectures applied to machine translation, we recommend the configuration that successfully transferred from IWSLT14 to the large-scale WMT17 dataset (4.3). As shown in Table 16, this configuration uses a higher dendrite count ($M = 7$) compared to MLPs and employs spatial distributions to organize connectivity.

Table 16: Recommended DNM configuration for Transformers on Machine Translation tasks (derived from IWSLT14 to WMT17 transfer).

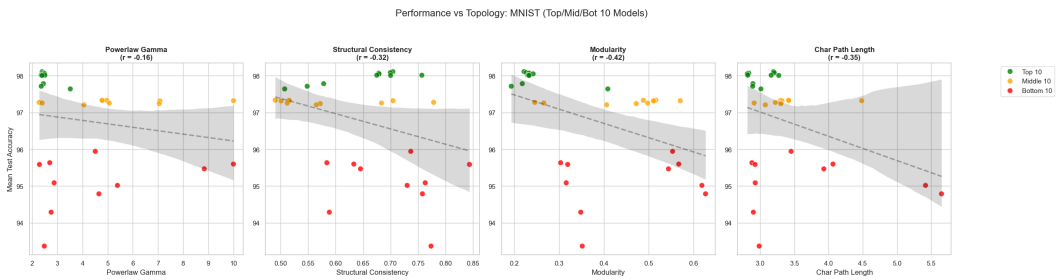
Parameter	**Value / Setting**
Sparsity (s)	0.90
Mean Dendrites	7
Dendritic Distribution	Fixed
Receptive window width	1.0
Receptive window width distribution	Spatial Gaussian
Degree Distribution	Spatial Inverse-Gaussian
Synaptic Distribution	Spatial Inverse-Gaussian
Layer Border Wiring Pattern	Wrap-around

N RELATIONSHIP BETWEEN TOPOLOGY AND PERFORMANCE

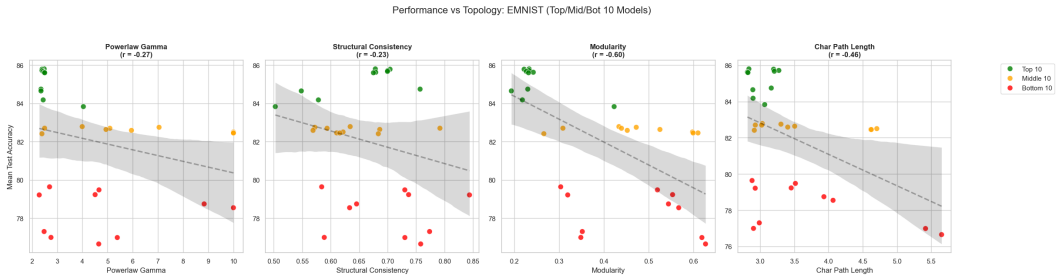
In this section, we analyze how different topological properties of DNM-initialized networks correlate with their performance on image classification tasks. We focus on four key graph-theoretic metrics: powerlaw distribution, average path length, clustering coefficient, and degree distribution. Understanding these relationships can provide insights into why certain configurations yield better results.

To investigate the structural drivers of performance in DNM-initialized networks, we analyzed the correlation between key graph-theoretic metrics and test accuracy across the MNIST, Fashion-MNIST, EMNIST, and CIFAR-10 datasets. For each dataset, we isolated the top 10, middle 10, and bottom 10 performing models from our hyperparameter search to visualize how topological variance dictates model efficacy. The results are summarized in Figure 8.

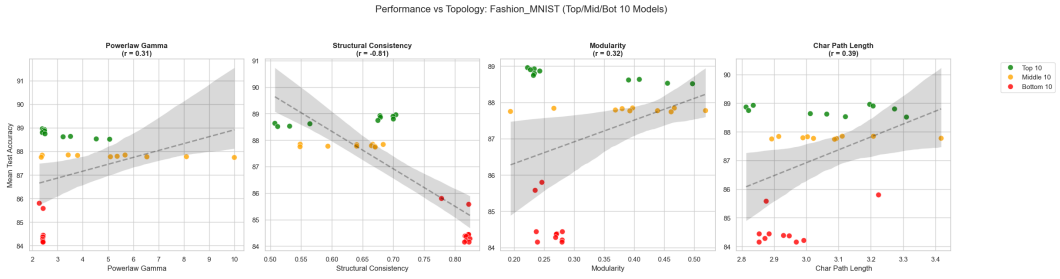
Powerlaw gamma Regarding the degree distribution exponent (γ), we see Fashion-MNIST aligning with CIFAR-10. Both datasets exhibit a positive correlation between accuracy and γ ($r = 0.31$



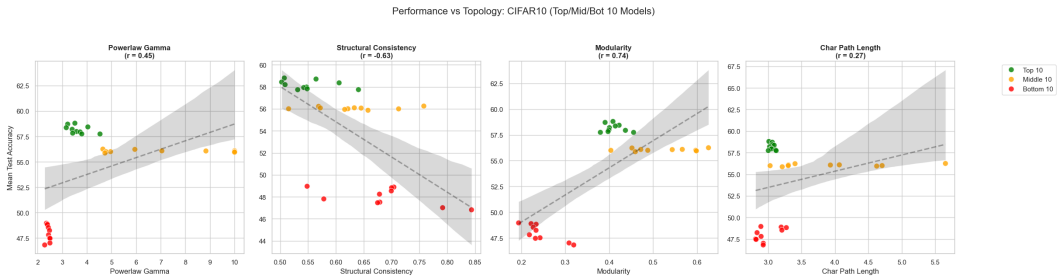
(a) MNIST



(b) EMNIST



(c) Fashion-MNIST



(d) CIFAR-10

Figure 8: **Performance vs. Topology Analysis.** Scatter plots showing the relationship between test accuracy and four topological metrics: Powerlaw Gamma (γ), Structural Consistency (σ_c), Modularity (Q), and Characteristic Path Length (L). Models are color-coded by performance tier: Top 10 (Green), Middle 10 (Orange), and Bottom 10 (Red). The trend lines and Pearson correlation coefficients (r) highlight task-specific topological preferences.

for Fashion-MNIST, $r = 0.45$ for CIFAR-10). Since a higher γ implies a steeper decay in the degree distribution (fewer extreme hubs), this indicates that complex tasks prefer a more distributed connectivity where information processing is shared among many nodes rather than concentrated in a few central hubs. Conversely, MNIST and EMNIST show negative correlations ($r = -0.16$

and $r = -0.27$), suggesting they benefit from the strong, centralized integration provided by heavy-tailed, hub-dominated topologies.

Structural Consistency Across all four datasets, we observe a consistent negative correlation between Structural Consistency (σ_c) and accuracy. Fashion-MNIST exhibits the strongest negative correlation ($r = -0.81$), reinforcing the finding that strict structural predictability is detrimental to initialization performance, regardless of the dataset complexity.

Modularity We observe that higher modularity (Q) correlates positively with performance across CIFAR-10. The best-performing topologies exhibit high modularity, suggesting that complex natural image recognition benefits from distinct, specialized communities of neurons that process local features independently before integration. Conversely, for the simpler MNIST and EMNIST datasets, lower modularity appears advantageous, indicating that a more integrated network structure suffices for less complex tasks.

Characteristic Path Like CIFAR-10, Fashion-MNIST models show a positive correlation ($r = 0.39$), indicating a benefit from longer path lengths that preserve local processing. This stands in contrast to MNIST and EMNIST ($r < 0$), which favor "small-world" architectures with short global integration paths.

O Learning rate sensitivity

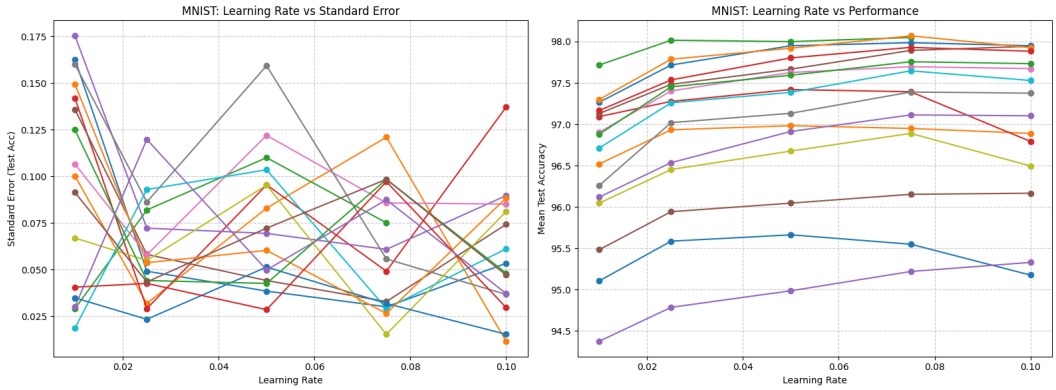
This section analyzes the impact of learning rate variations on model stability (Standard Error) and predictive performance (Mean Test Accuracy) across three distinct datasets: Fashion MNIST, EMNIST, and MNIST. The plots illustrate the trade-off between convergence speed and stability as the learning rate is increased from 0.01 to 0.1 (Figure 9). Across all three datasets, a learning rate in the range of 0.025 to 0.05 appears to be the optimal operating window. This range consistently provides the best balance of maximizing test accuracy while avoiding the instability and divergence (high standard error) associated with rates approaching 0.10.

P Comparison with Dense Equivalent Architectures

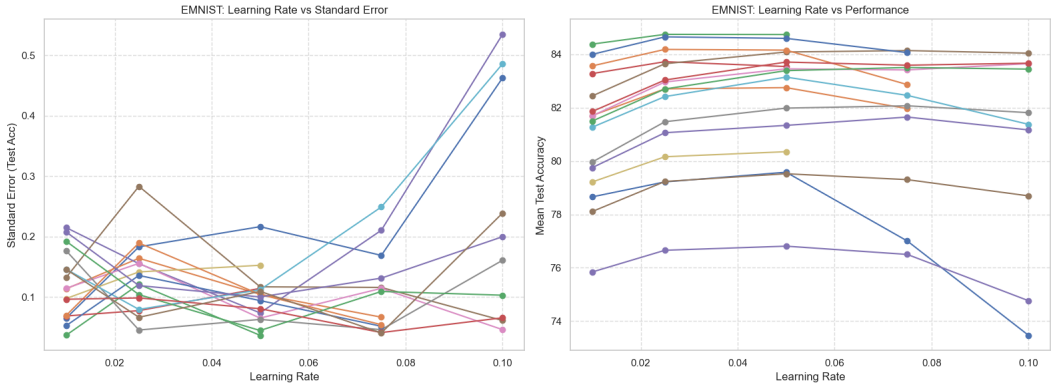
To rigorously validate the structural advantage of the Dendritic Network Model (DNM), it is necessary to decouple the benefits of network topology from the benefits of parameter efficiency. While Table 1 compares DNM against the over-parameterized Fully Connected (FC) network, a more rigorous baseline is the *Dense Equivalent* (DE) architecture.

Motivation At 99% sparsity, a sparse network retains only 1% of the parameters of the original FC model. A critical question arises: *Does the sparse topology provide an inductive bias that facilitates learning, or could a dense network with the same tiny parameter budget perform equally well?* The Dense Equivalent model is constructed by reducing the hidden dimension size H of the MLP such that the total number of trainable parameters equals the non-zero parameter count of the sparse DNM model. If DNM outperforms the DE baseline, it confirms that the *arrangement* of connections (the dendritic topology) is superior to a fully connected structure of equal capacity. Conversely, if the DE model performs comparably, it would suggest that the performance is governed solely by the parameter count, negating the need for complex sparse initialization.

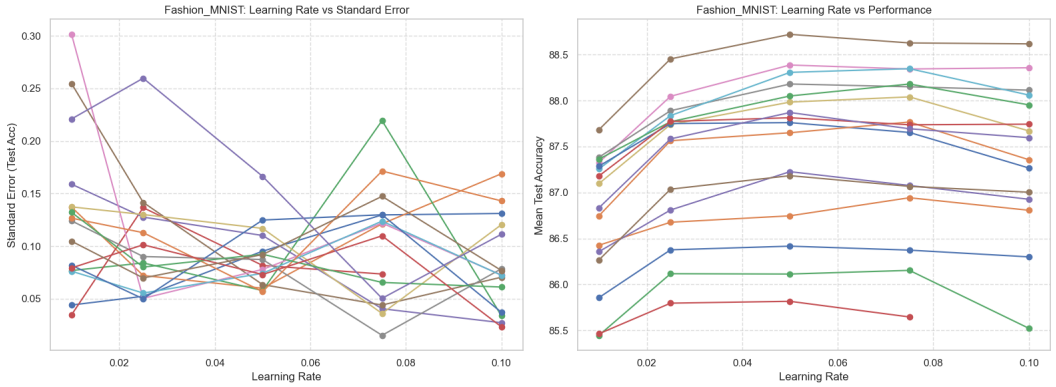
Results Table 17 presents the comparison between the Dense Equivalent baseline and the sparse initializations at 99% sparsity for image classification on the CIFAR-10 dataset. To match the extreme parameter reduction of 99% sparsity, the layer widths of the dense network must be drastically reduced. This severely limits the width of the representation that can be propagated through the network. In contrast, other sparse initialization methods maintain the original wide layer dimensions but sparsify the connections. As shown in the results, DNM significantly outperforms the Dense Equivalent baseline across all datasets.



(a) MNIST



(b) EMNIST



(c) Fashion MNIST

Figure 9: **Learning Rate Sensitivity Analysis.** Each subplot illustrates the relationship between learning rate, mean test accuracy, and standard error for three datasets: (a) MNIST, (b) EMNIST, and (c) Fashion MNIST. The blue line represents mean test accuracy, while the orange line indicates standard error across different learning rates. Error bars denote standard error.

Q GENERALIZATION ON NON-VISUAL OR NON-LANGUAGE DOMAINS.

The DNM is proposed as a general, brain-inspired topological initialization for linear layers across various architectures. While our main focus is on visual and language domains, we acknowledge the importance of testing generalization. To address this, performed two additional experiments targeting Graph-Based Learning and Reinforcement Learning scenarios. In both tests, we have

Table 17: Comparison of Static Sparse Training vs. Dense Equivalent (DE) architectures on the CIFAR-10 dataset. Models utilize the same number of trainable parameters (approx. 1% of the original FC network). The DE model uses reduced hidden dimensions to match the parameter count of the 99% sparse DNM. Results are averaged over 3 seeds ± standard error.

Static Sparse Training	
	CIFAR10
FC	62.35±0.13
CSTI	52.60±0.25
SNIP	61.89±0.48
DE	56.37±0.31
Random	54.75±0.15
BSW	56.26±0.04
BRF	54.86±0.08
Ramanujan	54.61±0.32
RadiX-Nets	50.90±0.23
dANN-R	51.57±0.23
DNM	**58.71±0.28**

fixed a parametric configuration for DNM (all distributions fixed, 3 dendrites) without performing any hyperparameter tuning.

Pattern Recognition Bandit (Granmo, 2018) We evaluated DNM on a "PatternBandit" task, where an agent must detect a specific local signal pattern ([1, 1, 1, 1]) hidden within a noisy state vector to receive a reward (Range: -1.0 to 1.0). This task tests the network's ability to maintain local feature integration capabilities under sparsity. We compared a Dense baseline against Random and DNM initializations using CHTs at 90% and 95% sparsity. Table 18 proves that DNM is capable of outperforming both the dense and the randomly initialized sparse networks, both at 90% and 95% sparsities.

Synthetic Node Classification We applied DNM to the linear transformation layers of a Graph Convolutional Network (GCN) (Kipf, 2016) on a synthetic dataset with high informative feature density (85% informative features). We tested extreme sparsity regimes (99%, 95%, and 90%) to see if DNM prevents topological collapse. The results are shown in Table 19, and show that the principles of DNM generalize effectively to non-visual and non-language domains, outperforming fully connected networks at extreme sparsity levels.

R DNM'S INTERACTION WITH TRAINING DYNAMICS

We agree that the primary focus of our work is to introduce and validate DNM as a novel sparse topological initialization framework. Our central hypothesis is that a bio-inspired, structured initial topology provides a significant structural advantage over conventional random or other sparse initializations. To test this, our primary investigation focused on the most critical dynamic related to our contribution: the dynamics of sparsity itself.To address the lack of exploration of how DNM interacts with other training dynamics beyond sparsity, we compared our model with a random sparse initialization for static sparse training on MLPs for image classification on CIFAR-10. For these tests, we selected the best performing parametric configuration for image classification on CIFAR-10. DNM demonstrates superior training dynamics compared to random initialization, as shown in Figure 10. This figure compares the accuracy, loss, and average gradient norm through training epochs of the DNM-initialized network compared to the random one.

Gradient Norms In the Random initialization, the gradient norm for deeper layers (e.g., Layer 2, represented by the blue line) remains suppressed and stagnant around 0.3-0.4. In contrast, the DNM initialization facilitates significantly healthier gradient flow, with Layer 2 norms rising steadily to about 0.6. This suggests that the dendritic topology mitigates the vanishing gradient problem in

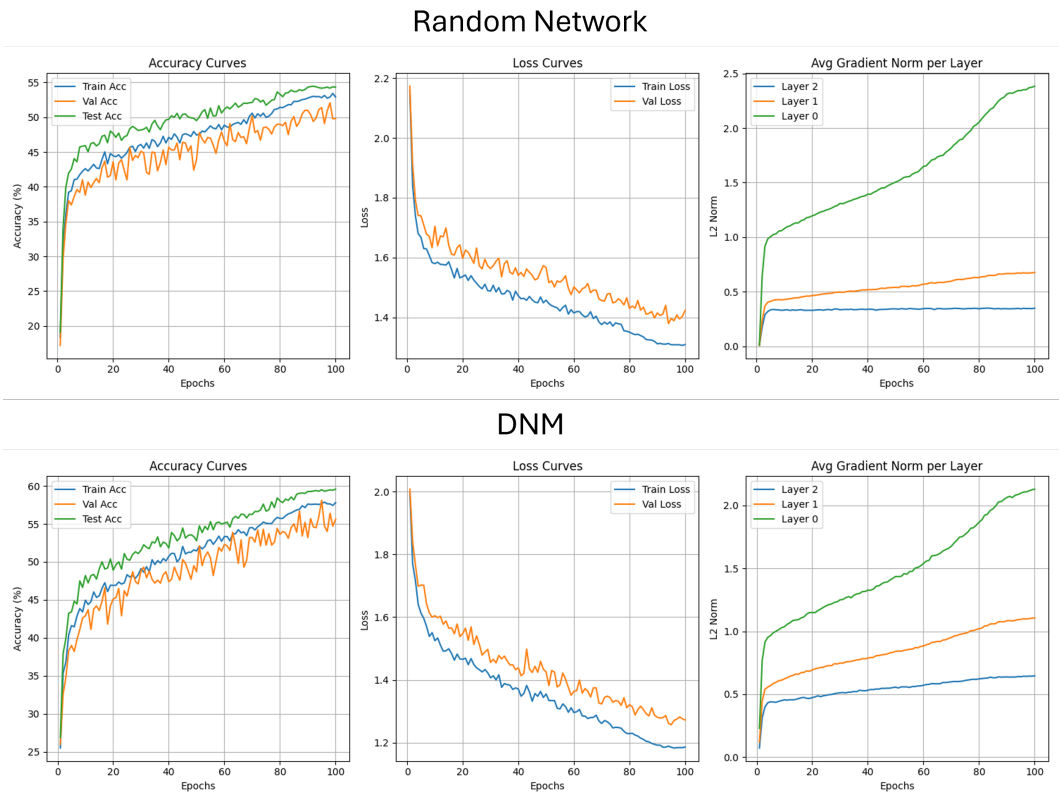


Figure 10: **Comparison of training dynamics between DNM and Random initialization.** The plots illustrate the evolution of test accuracy, training/validation loss, and gradient norms (Layer 2) over 100 epochs on CIFAR-10. DNM exhibits faster convergence, higher final accuracy, and healthier gradient flow compared to random initialization.

sparse networks better than random connections, allowing for more effective weight updates in downstream layers.

Loss and Accuracy The DNM model achieves a lower final training and validation loss compared to the Random model. Consequently, DNM reaches a higher final testing accuracy ($\approx 60\%$) compared to the Random baseline ($\approx 55\%$).

Convergence Speed Analysis To quantify the convergence speed, we utilize the Area Across Epochs (AAE) metric (**?**). AAE integrates the area between the ideal accuracy (100%) and the learning curve over the training duration E. A lower AAE integrates the area underneath the learning curve over the training epochs E. A higher AAE indicates that the model reached higher accuracy levels faster and sustained them throughout the training process. The DNM model achieves a 53.32 AAE measure, while the random model achieves 49.68. DNM yields a higher AAE by a margin of 3.64, confirming that it converges significantly faster.

S Combination of DNM with Pruning and Regularization Techniques

DNM performs well as an initialization method of DST frameworks, which inherently involves pruning links. DNM can also be combined effectively with regularization techniques, as proved by the tests on image classification, where models were trained with weight decay, which is one of the most popular forms of L2 regularization.

Table 18: Performance comparison on Mean Reward (averaged over 3 seeds). DNM outperforms both dense and random baselines at high sparsity.

Sparsity	Model Configuration	Mean Reward	vs Random Baseline
95%	Dense (Baseline)	0.71	-
	Random + CHTs	0.65	-
	DNM + CHTs	**0.72**	+10.77%
90%	Dense (Baseline)	0.71	-
	Random + CHTs	0.65	-
	DNM + CHTs	**0.75**	+15.39%

Table 19: GCN Synthetic Node Classification Accuracy. Tests performed on extreme sparsity regimes to evaluate topological collapse.

Sparsity	Model Configuration	Mean Accuracy	vs Random Baseline
99%	Dense (Baseline)	0.7544	-
	Random Sparse	0.6433	-
	DNM + CHTs	**0.7294**	+13.38%
95%	Dense (Baseline)	0.7544	-
	Random Sparse	0.7572	-
	DNM + CHTs	**0.7594**	+0.29%
90%	Dense (Baseline)	0.7544	-
	Random Sparse	0.7561	-
	DNM + CHTs	**0.7628**	+0.89%

T REPRODUCTION STATEMENT

All experiments were conducted on NVIDIA A100 80GB GPUs. MLP and Transformer models were trained using a single GPU. The code to reproduce the experiments will be made publicly available upon publication.

U CLAIM OF LLM USAGE

The authors declare that Large Language Models (LLMs) were used in the writing process of this manuscript. However, the core idea and principles of the article are entirely original and were not generated by LLMs.