Brain network science modelling of sparse neural networks enables Transformers and LLMs to perform as fully connected

Yingtao Zhang[1,2], **Diego Cerretti**[1,2], **Jialin Zhao**[1,2], **Ziheng Liao**[1,2], **Wenjing Wu**[1,2]
Umberto Michieli[4] **& Carlo Vittorio Cannistraci**[1,2,3*]

[1]Center for Complex Network Intelligence (CCNI)[†]
[2]Dept. of Computer Science, [3]Dept. of Biomedical Engineering, Tsinghua University
[4]University of Padova

Abstract

This study aims to enlarge our current knowledge on the application of brain-inspired network science principles for training artificial neural networks (ANNs) with sparse connectivity. Dynamic sparse training (DST) emulates the synaptic turnover of real brain networks, reducing the computational demands of training and inference in ANNs. However, existing DST methods face difficulties in maintaining peak performance at high connectivity sparsity levels. The Cannistraci-Hebb training (CHT) is a brain-inspired method that is used in DST for growing synaptic connectivity in sparse neural networks. CHT leverages a gradient-free, topology-driven link regrowth mechanism, which has been shown to achieve ultra-sparse (1% connectivity or lower) advantage across various tasks compared to fully connected networks. Yet, CHT suffers two main drawbacks: (i) its time complexity is $\mathcal{O}(N \cdot d^3)$- N node network size, d node degree - hence it can be efficiently applied only to ultra-sparse networks. (ii) it rigidly selects top link prediction scores, which is inappropriate for the early training epochs, when the network topology presents many unreliable connections. Here, we design the first brain-inspired network model - termed bipartite receptive field (BRF) - to initialize the connectivity of sparse artificial neural networks. Then, we propose a matrix multiplication GPU-friendly approximation of the CH link predictor, which reduces the computational complexity to $\mathcal{O}(N^3)$, enabling a fast implementation of link prediction in large-scale models. Moreover, we introduce the Cannistraci-Hebb training soft rule (CHTs), which adopts a flexible strategy for sampling connections in both link removal and regrowth, balancing the exploration and exploitation of network topology. Additionally, we propose a sigmoid-based gradual density decay strategy, leading to an advanced framework referred to as CHTss. Empirical results show that BRF offers performance advantages over previous network science models. Using 1% of connections, CHTs outperforms fully connected networks in MLP architectures on visual classification tasks, compressing some networks to less than 30% of the nodes. Using 5% of the connections, CHTss outperforms fully connected networks in two Transformer-based machine translation tasks. Finally, using 30% of the connections, CHTs and CHTss achieve superior performance compared to other dynamic sparse training methods in language modeling across different sparsity levels, and it surpasses the fully connected counterpart in zero-shot evaluations.

[*] Corresponding author: kalokagathos.agon@gmail.com

[†] Research Center in Tsinghua Laboratory of Brain and Intelligence (THBI), Department of Psychological and Cognitive Sciences.

Preprint. Under review.

1 Introduction

Artificial neural networks (ANNs) have led to significant advances in various fields such as natural language processing, computer vision, and deep reinforcement learning. The most common ANNs consist of several fully connected (FC) layers, which account for a large portion of the total parameters in recent large language models [1, 2]. This dense connectivity poses major challenges during the model training and deployment phases. In contrast, neural networks in the brain inherently exhibit sparse connectivity [3, 4]. This natural design in the brain exploits sparsity, suggesting a model in which the number of connections does not scale quadratically with the number of neurons. This could alleviate computational constraints, enabling more scalable network architectures.

Dynamic sparse training (DST) [5, 6, 7, 8, 9] has emerged as a promising approach to reduce computational and memory overhead of training deep neural networks while maintaining or even improving model performance. DST is also biologically inspired: it draws an analogy to synaptic turnover [10] in the brain, a fundamental neurobiological process in which synapses are continuously formed, strengthened, weakened, and eliminated over time. This dynamic rewiring enables the brain to adapt, learn, and store memories efficiently while preserving the overall stability of the network. Similarly, in DST, connections are dynamically pruned and regrown throughout training, allowing the network to adapt its connectivity structure in response to learning signals while maintaining a fixed sparsity level.

Apart from some detailed distinctions, the primary innovation in this field centers on the development of the regrowth criterion. A notable advancement is the gradient-free regrowth method introduced by Cannistraci-Hebb training (CHT) [9]. This method is inspired by epitopological learning—literally meaning ‘new topology’—and is rooted in brain-inspired network science theory [11, 12, 13, 14, 15]. Epitopological learning explores how learning can be implemented on complex networks by changing the shape of their connectivity structure (epitopological plasticity). CHT has demonstrated remarkable advantages in training ultra-sparse ANNs with connectivity levels of 1% or lower, often outperforming fully connected networks in various tasks. However, despite these advances, CHT encounters two major challenges: 1) During dynamic sparse training, its rigid link selection mechanism can lead to *epitopological local minima* where the sets of removed links and regrown links exhibit significant overlap, severely impedes the exploration of optimal network topologies. 2) The time complexity of the CHT regrowth method, Cannistraci-Hebb 3 on Length 3 paths (CH3-L3p), is $\mathcal{O}(N \cdot d^3)$, where N represents the number of nodes in the network and d is the average degree. A length 3 path is a walk of three consecutive links. As the network becomes denser, the time complexity approaches $\mathcal{O}(N^4)$, rendering it impractical for large-scale and higher-density models.

In this article, we present the **C**annistraci-**H**ebb **T**raining **s**oft rule (CHTs), which introduces several key innovations: 1) To address the issue of epitopological local minima, CHTs employs a multinomial distribution to sample link scores from removal and regrowth metrics, enabling more flexible and effective exploration of network topologies. 2) CHTs incorporates novel substitution node-based link prediction mechanisms, reducing the computational time complexity to $\mathcal{O}(N^3)$. This improvement makes CHTs scalable to large-scale models with higher network density. 3) CHTs initializes the sparse topologies for bipartite networks with the brain-like receptive field, demonstrating superior performance compared to the traditional Erdős–Rényi [5], bipartite small world, and bipartite scale-free model [9]. Additionally, we propose a sigmoid gradual density decay strategy, which, when integrated with CHTs, forms an enhanced framework termed CHTss. This combined approach further optimizes the training process for sparse neural networks.

To evaluate the effectiveness of the **C**annistraci-**H**ebb **T**raining **s**oft rule with **s**igmoid density decay (CHTss), we conduct extensive experiments across multiple architectures and tasks. Firstly, to assess the basic concept of CHTs, we employ MLPs on benchmark datasets, including MNIST [16], EMNIST [17], and Fashion MNIST [18]. The results show that CHTs performs better than fully connected networks with only 1% of the connections (99% sparsity) in MLPs. Further, to evaluate the end-to-end approach CHTss, we utilize Transformers [19] on machine translation datasets such as Multi30k en-de [20], IWSLT14 en-de [21], and WMT17 en-de [22]. Additionally, we test CHTs and CHTss also on the LLaMA-130M model with language modeling tasks and zero-shot evaluation tasks. From the results of the above experiments, CHTss outperforms fully connected Transformers with only 5% of the links on Multi30k and IWSLT and achieves performance comparable to the fully connected LLaMA-130M in language modeling tasks on OpenWebText. Moreover, CHTs and CHTss outperform the fully connected LLaMA-130M on zero-shot evaluation tasks on GLUE [23]

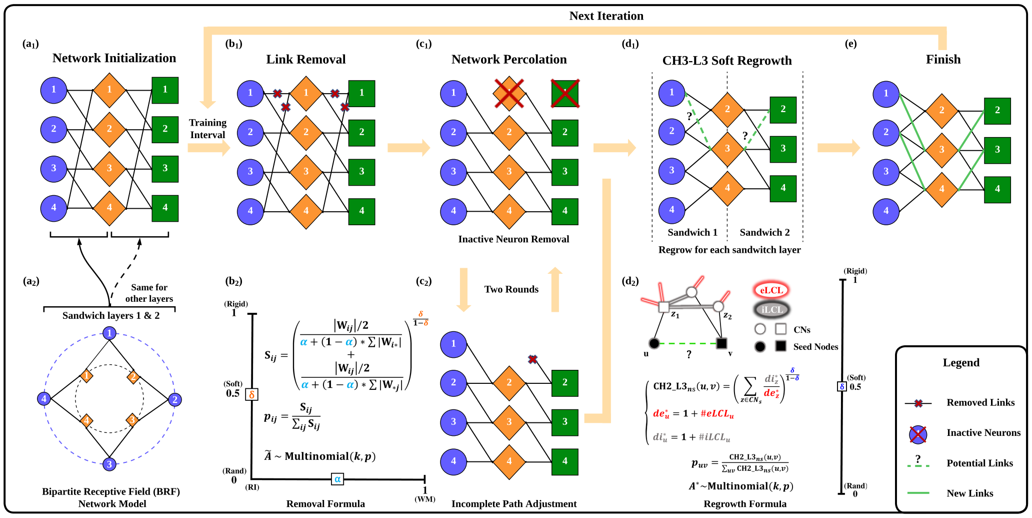
Next Iteration
(a₁)
Network Initialization
Training Interval
(b₁)
Link Removal
(c₁)
Network Percolation
Inactive Neuron Removal
(d₁)
CH3-L3 Soft Regrowth
?
?
Sandwich 1
Sandwich 2
Regrow for each sandwitch layer
(e)
Finish
(a₂)
Same for other layers
Sandwich layers 1 & 2
Bipartite Receptive Field (BRF) Network Model
(b₂)
(Rigid) 1
(Soft) 0.5
δ
(Rand) 0
(RI)
α
1 (WM)
$S_{ij} = \left(\frac{|W_{ij}|/2}{\alpha + (1-\alpha) * \sum |W_{i*}|} + \frac{|W_{ij}|/2}{\alpha + (1-\alpha) * \sum |W_{*j}|} \right)^{\frac{\delta}{1-\delta}}$
$p_{ij} = \frac{S_{ij}}{\sum_{ij} S_{ij}}$
$\tilde{A} \sim \text{Multinomial}(k, p)$
Removal Formula
(c₂)
Two Rounds
Incomplete Path Adjustment
(d₂)
z_1
z_2
u
?
v
eLCL
iLCL
CNs
Seed Nodes
$\text{CH2\_L3}_{ns}(u,v) = \left(\sum_{z \in CN_s} \frac{di_z^*}{de_z^*} \right)^{\frac{\delta}{1-\delta}}$
$de_u^* = 1 + \#eLCL_u$
$di_u^* = 1 + \#iLCL_u$
$p_{uv} = \frac{\text{CH2\_L3}_{ns}(u,v)}{\sum_{uv} \text{CH2\_L3}_{ns}(u,v)}$
$A^* \sim \text{Multinomial}(k, p)$
Regrowth Formula
(Rigid) 1
δ (Soft) 0.5
(Rand) 0
Legend
Removed Links
Inactive Neurons
?
Potential Links
New Links

Figure 1: **Illustration of the CHTs process.** One training iteration follows the steps of (a1) → (b1) → (c1) → (c2) → (d1) → (e). (a1) Network initialization with each of the sandwich layers (bipartite networks connecting layers' input nodes to their output nodes) being a bipartite receptive field (BRF) network. (a2) BRF network representation with $r = 0$. (b1) Link removal process. (b2) Formula for determining which links to remove. (c1) Removal of inactive neurons caused by link removal. (c2) Adjust and remove incomplete links caused by inactive neuron removal. (d1) Regrowth of links according to the CH2-L3 node-based soft rule. (d2) Detailed illustration of the CH2-L3 node-based soft rule. (e) Finished state of the network after one iteration. The next iteration repeats the steps (b1) - (e) from this finished state. $\tilde{A}$ indicates the removal set of the iteration and A^* is the regrown set.

and SuperGLUE [24] with only 30% connections. These findings underscore the potential of CHTs and CHTss in enabling highly efficient and effective large-scale sparse neural network training.

2 Related Work

2.1 Dynamic sparse training

Dynamic sparse training is a subset of sparse training methodologies. Unlike static sparse training methods (also known as pruning at initialization) [25, 26, 27, 28], dynamic sparse training allows for the evolution of network topology during the training process. The pioneering method in this field is Sparse Evolutionary Training (SET) [5], which removes links based on the magnitude of their weights and regrows new links randomly. Subsequent developments have sought to refine and expand upon this concept of dynamic topological evolution. One such advancement was proposed by DeepR [29], a method that adjusts network connections based on stochastic gradient updates combined with a Bayesian-inspired update rule. Another significant contribution is RigL [7], which leverages the gradient information of non-existing links to guide the regrowth of new connections during training. MEST [8] utilizes both gradient and weight magnitude information to selectively remove and randomly regrow new links, analogously to SET. In addition, it introduces an EM&S strategy that allows the model to train at a higher density and gradually converge to the target sparsity. The Top-KAST [6] method maintains constant sparsity throughout training by selecting the top K parameters based on parameter magnitude at each training step and applying gradients to a broader subset B, where $B \supset A$. To avoid settling on a suboptimal sparse subset, Top-KAST also introduces an auxiliary exploration loss that encourages ongoing adaptation of the mask. Additionally, sRigL [30] adapts the principles of RigL to semi-structured sparsity, facilitating the training of vision models from scratch with actual speed-ups during training phases. Despite these advancements, the state-of-the-art method remains RigL-based, yet it is not fully sparse in backpropagation, necessitating the computation of gradients for non-existing links. Addressing this limitation, Zhang et al. [9] propose CHT, a dynamic sparse training methodology that adopts a gradient-free regrowth strategy that relies solely on topological information (network shape intelligence), achieving an ultra-sparse configuration that surpasses fully connected networks in some tasks.

2.2 Cannistraci-Hebb Theory and Network Shape Intelligence

As the SOTA gradient-free link regrown method, CHT [9] originates from a brain-inspired network science theory. Drawn from neurobiology, Hebbian learning was introduced in 1949 [31] and can be summarized in the axiom: “neurons that fire together wire together.” This could be interpreted in two ways: changing the synaptic weights (weight plasticity) and changing the shape of synaptic connectivity [11, 12, 13, 14, 15]. The latter is also called *epitopological plasticity* [11] because plasticity means “to change shape,” and *epitopological* means “via a new topology.” *Epitopological Learning* (EL) [12, 13, 14] is derived from this second interpretation of Hebbian learning and studies how to implement learning on networks by changing the shape of their connectivity structure. One way to implement EL is via link prediction, which predicts the existence and likelihood of each nonobserved link in a network. CH3-L3 is one of the best-performing and most robust network automata, belonging to the Cannistraci-Hebb (CH) theory [32], which can automatically evolve the network topology starting from a given structure. The rationale is that, in any complex network with local-community organization, the cohort of nodes tends to be co-activated (fire together) and to learn by forming new connections between them (wire together) because they are topologically isolated in the same local community [32]. This minimization of the external links induces a topological isolation of the local community, which is equivalent to forming a barrier around it. The external barrier is fundamental to maintaining and reinforcing the signaling in the local community, inducing the formation of new links that participate in epitopological learning and plasticity.

3 Cannistraci-Hebb training soft rule with sigmoid gradual density decay

3.1 Soft removal and regrowth.

Definition 1. Epitopological local minima. In the context of dynamic sparse training methods, we define an epitopological local minima (ELM) as a state where the sets of removed links and regrown links exhibit a significant overlap.

Let A_t be the set of existing links in the network at the training step t. Let $\tilde{A}_t$ be the set of removal links and A_t^* be the set of regrown links. The overlap set between removed and regrown links at step t can be quantified as $O_t = \tilde{A}_t \cap A_t^*$. An ELM occurs if the size of O_t at step t is significantly large compared to the size of A_t^*, indicating a high probability of the same links being removed and regrown repeatedly throughout the subsequent training steps. This can be formally represented as $\frac{|O_t|}{|A_t^*|} \geq \theta$, where θ is a predefined threshold close to 1, indicating strong overlap. This definition is essential for the understanding of CHT, as evidenced by the article [9] indicating that the overlap rate between removed and regrown links becomes significantly high within just a few epochs, leading to rapid topological convergence towards the ELM. Previously, CHT implemented a topological early stop strategy to avoid predicting the same links iteratively. However, it will stop the topological exploration very fast and potentially trap the model within the ELM.

In this article, we adopt a probabilistic approach where the regrowth process is modeled as sampling from a $\{0, 1\}$ multinomial distribution, with probabilities determined by link prediction scores, thereby introducing a "soft sampling" mechanism. Under this formulation, each mask value is not rigidly dictated by the link prediction score; instead, low-score links can still be selected with lower probability, facilitating escape from epitopological local minima (ELM). To demonstrate that soft sampling effectively balances exploration and exploitation, we evaluate its behavior in Figure 3, which presents the impact of varying softness levels during training of LLaMA-60M for 5000 steps under 90% sparsity.

We compare the in-time over-parameterization (ITOP) rate [33], which quantifies the cumulative proportion of links activated throughout training. As shown, deterministic regrowth leads to rapid topological convergence after approximately 1000 steps, indicating that it becomes trapped in an ELM without further exploration. Random regrowth, while capable of escaping ELMs by introducing new connections, fails to exploit the underlying topological structure effectively. In contrast, soft regrowth achieves a balance by both exploiting the current topology and exploring new link combinations probabilistically. This balance enables a more appropriate exploration of the connectivity space, ultimately leading to superior performance, as evidenced by the results.

Link removal alternating from weight magnitude and relative importance. We illustrate the link removal part of CHTs in Figure 1b1) and b2). We employ two methods, Weight Magnitude (WM) $|\mathbf{W}|$ and Relative Importance (RI) [34], to remove the connections during dynamic sparse training. Given an input node i and an output node j connected with weight W_{ij}, we define the relative importance RI_{ij} as:

$$\mathbf{RI}_{ij} = \frac{|\mathbf{W}_{ij}|}{\sum |\mathbf{W}_{*j}|} + \frac{|\mathbf{W}_{ij}|}{\sum |\mathbf{W}_{i*}|} \tag{1}$$

As illustrated in Equation 1, RI assesses connections by normalizing the absolute weight of links that share the same input or output neurons. This method does not require calibration data and can perform comparably to the baseline post-training pruning methods like sparsegpt [35] and wanda [36]. Generally, WM and RI are straightforward, effective, and quick to implement in DST for link removal, but give different directions for network percolation. WM prioritizes links with higher weight magnitudes, leading to rapid network percolation, whereas RI inherently values links connected to lower-degree nodes, thus maintaining a higher active neuron post-percolation (ANP) rate. The ANP rate is the ratio of the number of active neurons after training over the original number of neurons before training. These methods are equally valid but cater to different scenarios. For instance, using RI significantly improves results on the Fashion MNIST dataset compared to WM, whereas WM performs better on the MNIST and EMNIST datasets.

Soft link removal. In the early stages of training, both WM and RI are not reliable due to the model's underdevelopment. Therefore, rather than strictly selecting top values based on WM and RI, we also sample links from a multinomial distribution using an importance score calculated by the removal metrics. The final formula for link removal is defined in Equation 2.

$$\mathbf{S}_{ij} = \left(\frac{|\mathbf{W}_{ij}|/2}{\alpha + (1-\alpha) \sum |\mathbf{W}_{i*}|} + \frac{|\mathbf{W}_{ij}|/2}{\alpha + (1-\alpha) \sum |\mathbf{W}_{*j}|} \right)^{\frac{\delta}{1-\delta}} \tag{2}$$

Here, α determines the removal strategy, shifting from weight magnitude ($\alpha = 1$) to relative importance ($\alpha = 0$). In all experiments, we only evaluate these two α values. δ adjusts the softness of the sampling process. As training progresses and weights become more reliable, we adaptively increase δ from 0.5 to 0.75 to refine the sampling strategy and improve model performance. These settings are constant for all the experiments in this article.

Node-based link regrowth. Another significant challenge for CHT lies in the time complexity of link prediction. In the original CHT framework [9], the path-based CH3-L3p metric is employed for link regrowth, as discussed in Appendix C. However, this method incurs a high computational cost due to the need to compute and store all length-three paths, resulting in a time complexity of $O(N \cdot d^3)$, where N is the number of nodes and d is the network's average degree. This complexity is prohibitive for large models with numerous nodes and higher-density layers.

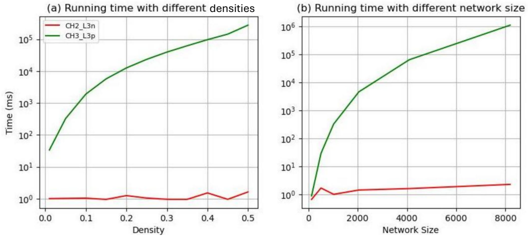

Figure 2: **One-time Link Prediction Runtime Performance Evaluation** of node-based and path-based methods across varying densities and network sizes. In (a), the network size is fixed at 1024 × 1024, while in (b), the density is fixed at 5%.

To address this issue, we introduce a more efficient, node-based paradigm that eliminates the reliance on length-three paths between seed nodes. Instead, this approach focuses on the common neighbors of seed nodes. The node-based version of CH3-L3p, denoted as CH2-L3n, also depends on the internal local community links (iLCL, the links between the common neighbors [32]) to enhance the expressiveness of the formula. This variant is formulated as:

$$\textbf{CH2-L3n}(u, v) = \sum_{z \in L3} \frac{di_z^*}{de_z^*} \tag{3}$$

Here, u and v denote the seed nodes, while z is the common neighbor on the L3 path [32], a walk of three consecutive links that connects u to v via two of those intermediate nodes. The terms di_z^* de_z^* represent the number of internal local community links (iLCL) and external local community links (eLCL) of node i, with a default increment of 1 to prevent division by zero. The detailed explanation of iLCL and eLCL can be found in Appendix C. The pseudocode is provided in the supplementary material. The new node-based variant, CH2-L3n, reduces the computational complexity to $O(N^3)$ and enables efficient matrix-based computations on GPUs. We evaluate the one-time link prediction runtime performance of both CH3-L3p and CH2-L3n across different network sizes and sparsity levels, as illustrated in Figure 2. The red and green lines depict the actual runtime for the path-based and node-based methods, respectively. The results reveal that the node-based version achieves significantly faster runtime performance compared to the path-based methods. Furthermore, the node-based method demonstrates consistently stable runtime across diverse network sizes and density levels, making it an ideal link prediction as the CH theory-based link regrowth component of CHTs in dynamic sparse training for large-scale models.

3.2 Bipartite receptive field network modeling.

In this article, we propose the Bipartite Receptive Field (BRF) network model, a novel sparse topological initialization method capable of generating brain-network-like receptive field connectivity. 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 [37], the Watts-Strogatz (WS) model [38], and the Barabási-Albert (BA) model [39], which are not brain-inspired. Since the standard WS and BA models are not directly designed for bipartite networks, they were recently extended [9] into their bipartite counterparts and term as Bipartite Small-World (BSW) and Bipartite Scale-Free (BSF), respectively. BSW generally outperforms BSF for dynamic sparse training (see Appendix D).

During BSW initialization, each output node is connected to its spatially nearest input nodes. This spatially local connectivity pattern aligns with the concept of receptive fields observed in biological neural systems, where neurons respond selectively to localized regions of input space. However, the rewiring process of BSW does not follow brain mechanisms: it simply deletes a set of links from the closer input nodes to rewire them uniformly at random anywhere on the input layer. Conversely, the random allocation of connectivity in brain network topologies is guided by the spatial distance of the neurons [40, 41]. Unlike the BSW model that introduces random connectivity by a rewiring process, which cannot control the extent of spatial-dependent randomness injected in the topology, the BRF model directly generates a connectivity with a customized level of spatial-dependent randomness using a parameter $r \in [0, 1]$. A low value of r results in links that are densely clustered around the diagonal, while a higher value of r leads to less clustered connectivity patterns, which tend to be uniformly at random only for $r = 1$. Specifically, a bipartite adjacency matrix with links near the diagonal indicates that adjacent nodes from the two layers are linked, whereas links far from the diagonal correspond to more distant node pairs. The mathematical formula and implementation are detailed in Appendix D.

Furthermore, the degree distribution of the BSW model is fixed to the same value for all the nodes at the same layer before rewiring, whereas after rewiring, the degree distribution is not conserved, and the more links it rewires, the more it will be similar to the ER model. Instead, the BRF model has the important property to conserve the degree distribution of the output layer, which ensures that it maintains a designed receptive field connectivity. This means that an initialization setting of the BRF model is the output degree distribution, which in this study we consider fixed or uniformly at random, as shown in Appendix D. We also conduct a sensitivity test of the influence of r in Figure 7a). It should be noted that when $r = 0$, the network is equivalent to the BSW with $\beta = 0$, and when $r = 1$, the network becomes an ER network. The examples of the adjacency matrices of BSF, BSW, and BRF are shown in Figure 5.

3.3 Sigmoid Gradual Decrease Density

As demonstrated in GraNet [42] and MEST$_{EM\&S}$ [8], incorporating a density decrease strategy can significantly improve the performance of dynamic sparse training. In MEST$_{EM\&S}$, the density is reduced discretely at predefined epochs. In GraNet, the network evolution process consists of three steps: pruning, link removal, and link regrowth. The method first prunes the network to reduce the density, followed by removing and regrowing an equivalent number of links under the updated density. The density decrease in GraNet follows the same approach as Gradual Magnitude Pruning (GMP) [43], which adheres to a cubic function.

However, this density decay scheduler exhibits a sharp decline in the initial stages of training, which risks pruning a substantial fraction of weights before the model has sufficiently learned. To mitigate this issue, we propose a sigmoid-based gradual density decrease strategy, defined as:

$$s_t = s_i + (s_i - s_f) \left(\frac{1}{1 + e^{-k\left(t - \frac{t_f + t_0}{2}\right)}} \right), \tag{4}$$

where $t \in \{t_0, t_0 + \Delta t, \ldots, t_0 + n\Delta t\}$, s_i is the initial sparsity, s_f is the target sparsity, t_0 is the starting epoch of gradual pruning, t_f is the end epoch of gradual pruning, and Δt is the pruning frequency. k controls the curvature of the decrease. We set k=6 for all the experiments in this article. This strategy ensures a smoother initial pruning phase, allowing the model to warm up and stabilize before undergoing significant pruning, thereby enhancing training stability and performance. We explain how to adjust the training FLOPs of sigmoid density decay to the same as cubic decay in Appendix I.

In addition to refining the decay function, we replace the weight magnitude criterion used in the original GMP and GraNet processes with relative importance (RI). This adjustment is motivated by prior work [34], which has shown that RI provides a significant performance advantage over weight magnitude, particularly when pruning models initialized with high density.

4 Experiments

4.1 Experimental Setup

We evaluate the performance of CHTs using MLPs for image classification tasks on the MNIST [16], Fashion MNIST [18], EMNIST [17], and CIFAR10 [44] datasets. To further validate our approach, we apply the sigmoid gradual density decay strategy to Transformers for machine translation tasks on the Multi30k en-de [20], IWSLT14 en-de [21], and WMT17 en-de [22] datasets. Additionally, we conduct language modeling experiments using the OpenWebText dataset [45] and evaluate zero-shot performance on the GLUE [23] and SuperGLUE [24] benchmark with LLaMA-130M [1]. For MLP training, we sparsify all layers except the final layer, as ultra-sparsity in the output layer may lead to disconnected neurons, and the connections in the final layer are relatively minor compared to the previous layers. For Transformers and LLaMA models, we apply dynamic sparse training (DST) to all linear layers, excluding the embedding and final generator layer. Detailed hyperparameter settings for each experiment are provided in Tables 5, 6, and 7. We also conduct a series of ablation and sensitivity tests on the components proposed in this article for CHTs and CHTss in Appendix H

Table 1: Performance comparison of different fixed sparsity dynamic sparse training methods on the CIFAR10 dataset trained on an MLP at 99% sparsity. ACC represents accuracy, and ANP denotes the active neuron percolation rate, indicating the final size of the network. The lowest anp rate and the best dynamic sparse training method are highlighted in bold, and performances surpassing the fully connected model are marked with “*". The results present a standard error taken over three seeds of the experiments.

Method	ACC (%)	Comparison to FC	ANP
FC	62.85 ± 0.16	–	–
CHTs	**69.97 ± 0.06***	**+11.33%**	**54%**
CHT	59.10 ± 0.06	-5.97%	96%
RigL	63.90 ± 0.19*	+1.67%	59%
SET	62.70 ± 0.11	-0.24%	100%

Baseline Methods. We compare our method with the baseline approaches in the literature. We divide the dynamic sparse training (DST) methods into two categories: fixed sparsity DST and density decay DST. For the fixed sparsity DST, we compare CHTs with the SET [5], RigL [7], and CHT [9], and for the density decay DST methods, we compare CHTss with MEST$_{EM\&S}$ [8], GMP [43], and GraNet [42]. We explain the detailed implementations and the reason we split GMP as a type of DST method in Appendix G.

Table 2: Performance comparison on machine translation tasks of Multi30k, IWSLT, and WMT with varying final sparsity levels. The scores indicate BLEU scores, which is the higher the better. CHTs (GMP) represents CHTs with GMP's density decay strategy. Bold values denote the best performance among fixed sparsity DST methods or density decay DST methods. The performances that surpass the fully connected model are marked with "*". s_i indicates the starting sparsity of the DST methods that use the density decay strategy.

Method	Multi30k		IWSLT		WMT	
	95%	90%	95%	90%	95%	90%
FC	31.28		24.20		25.22	
SET	27.89	28.72	18.48	19.54	20.21	21.61
RigL	27.63	28.89	20.29	21.03	20.52	22.16
CHT	27.79	28.38	18.59	19.91	19.03	21.08
CHTs	29.45	29.89	21.39	22.02	21.05	22.36
MEST$_{EM\&S}$	28.71	28.26	18.95	20.77	20.79	22.3
GMP ($s_i = 0.5$)	26.42	27.06	22.44	22.62	22.29	23.52
GraNet ($s_i = 0.5$)	30.90	31.06	23.05	22.88	22.11	23.49
CHTss ($s_i = 0.5$)	**32.82***	**33.11***	**24.84***	**24.76***	**22.84**	**24.80**

4.2 MLP for image classification

Main results. In the MLP evaluation, we aim to assess the fundamental capacity of DST methods to train the fully connected module, which is common across many ANNs. The sparse topological initialization of CHT and CHTs is CSTI [9] since the input bipartite layer can directly receive information from the input pixels. Table 8 displays the performance of DST methods compared to their fully connected counterparts across three basic datasets of MNIST, Fashion MNIST, and EMNIST. The DST methods are tested at 99% sparsity. As shown in Table 8, both of the two regrowth methods of CHTs outperform the other fixed sparsity DST methods. Notably, the path-based CH3-L3p outperforms the fully connected one in all the datasets. The node-based CH2-L3n also achieves comparable performance on these basic datasets. However, considering the running time of CH3-L3p is unacceptable, especially in large scale models, in the rest of the experiments of this article, we only use CH2-L3n as the representative method to regrow new links. Table 1 presents a comparison of fixed-sparsity dynamic sparse training (DST) methods against the fully connected (FC) baseline. Notably, CHTs outperform all other DST methods and achieve an 11% improvement in accuracy over the fully connected model. In addition, we present the active neuron post-percolation rate (ANP) for each method in Table 8 and Table 1. It is evident that CHTs adaptively percolates the network more effectively while retaining performance.

4.3 Transformer on Machine Translation

We assess the Transformer's performance on a classic machine translation task across three datasets. We take the best performance of the model on the validation set and report the BLEU on the test set. Beam search, with a beam size of 2, is employed to optimize the evaluation process. In our evaluation, CHTs configures the topology of each layer using the BRF model, employs a weight magnitude soft link removal technique, and regrows new links using CH2-L3n-soft. Additionally, we apply an adjusted network percolation technique to the Transformer, as detailed in Appendix F. The findings, presented in Table 2, demonstrate that 1) CHTs surpasses other fixed density DST methods on all the sparsity scenarios. 2) Incorporating the sigmoid density decrease, CHTss outperforms even the fully connected ones with only 5% density on Multi30K and IWSLT.

4.4 Natural Language Generation

Language modeling. We utilize the LLaMA model family [1] across 60M, 130M, and 1B architecture as the baseline for our language generation experiments. We follow the experiment setup from [46] detailed in Table 7. To ensure that the FLOPs are the same for all the density decrease methods, for CHTss, we implement a half-step strategy for the density decay steps, as GMP and GraNet.

Table 3 shows the validation perplexity results of each algorithm across different density levels on LLaMA-60M and LLaMA-130M. CHTs stably outperforms SET, CHT, and RigL, while CHTss are constantly better than GraNet and GMP. At 70% sparsity, CHTss is already able to perform comparably to the fully connected model. It is important to note that RigL and GraNet exhibit subpar

Table 3: **Validation perplexity of different dynamic sparse training (DST) methods on OpenWebText using LLaMA-60M, LLaMA-130M, and LLaMA-1B across varying sparsity levels.** Bold values denote the best performance among fixed sparsity DST methods or density decay DST methods. Lower perplexity corresponds to better model performance. GMP, GraNet, and CHTss are run with an initial sparsity of $s_i = 0.5$. The test of CHT over LLaMA-1B is missing due to its excessive runtime. The performances that surpass the fully connected model are marked with “*".

Method	LLaMA-60M				LLaMA-130M				LLaMA-1B
	70%	80%	90%	95%	70%	80%	90%	95%	70%
FC	26.56				19.27				14.62
SET	31.77	30.69	35.26	39.70	20.82	22.02	24.73	28.37	16.37
RigL	39.96	41.33	45.34	51.49	25.85	66.35	37.18	49.39	149.17
CHT	31.02	32.99	35.01	41.87	21.02	22.82	26.27	30.01	–
CHTs	**28.12**	**29.84**	**33.03**	**36.47**	**20.10**	**21.33**	**23.71**	**26.45**	**14.53***
MEST	28.26	29.94	33.60	37.87	21.32	22.21	24.98	27.96	60.36
GMP	29.22	30.59	33.68	39.00	20.49	22.28	23.61	27.16	31.76
GraNet	30.55	31.51	33.76	39.98	22.84	29.03	26.81	61.31	79.44
CHTss	**27.62**	**29.00**	**31.42**	**35.10**	**19.85**	**20.70**	**22.51**	**25.07**	**15.41**

performance in this evaluation due to the use of bfloat16 precision in this configuration. This lower precision adversely impacts gradient accuracy, particularly in the early stages of training. Since both RigL and GraNet rely on gradient information to regrow new links, the imprecise gradients lead to regrowing the wrong links, thereby hindering their performance. To further validate this observation, we conduct additional experiments under FP32 precision, presented in Table 11. The results confirm that RigL and GraNet perform significantly better under high-precision training, although they still fall behind CHTs and CHTss. Importantly, industry trends are increasingly moving toward low-precision training and inference to improve efficiency [47, 48]. In this context, CHTs and CHTss demonstrate greater robustness compared to RigL and GraNet, making them better aligned with practical deployment needs under reduced-precision settings.

Zero-shot evaluations. The pretrained model of CHTs and CHTss with 30% sparsity and the fully connected model are evaluated for zero-shot performance on the GLUE [23](cola, sst2, mrpc, qqp, mnli, qnli, rte, wnli) and SuperGLUE [24] (boolq, hellaswag, CB, copa) benchmarks. We show the results in Table 4. Both the win rate and average performance clearly indicate that CHTs and CHTss outperform fully connected models in zero-shot settings, demonstrating their superior generalization capabilities despite high sparsity.

5 Conclusion and Discussion

We advance current knowledge in brain-inspired dynamic sparse training, proposing the Cannistraci-Hebb Training soft rule with sigmoid gradual density decay (CHTss). First, we introduce a matrix multiplication mathematical formula for GPU-friendly approximation of the CH link predictor. This significantly reduces the computational complexity of CHT and speeds up the running time, enabling the implementation of CHTs in large-scale models. Second, we propose a Cannistraci-Hebb training soft rule (CHTs), which innovatively utilizes a soft sampling rule for both removal and regrowth links, striking a balance for epitopological exploration and exploitation. Third, we propose the Bipartite Receptive Field (BRF) model to initialize the sparse network topology in a brain-inspired manner, enabling the network to preferentially capture spatially closer features. Finally, in transformer-based models, we integrate CHTs with a sigmoid gradual density decay strategy (CHTss). Empirically, CHTs demonstrate a remarkable ability to achieve ultra-sparse configurations—up to 99% sparsity in MLPs for image classification—surpassing fully connected networks. Notably, the regrowth process under CHTs does not rely on gradients. With the sigmoid gradual density decay, CHTss surpasses the fully connected Transformer using only 5% density and achieves comparable language modeling performance. Moreover, both CHTs and CHTss outperforms the fully connected LLaMA-130M on zero-shot evaluation tasks with only 30% density. This represents a relevant result for dynamic sparse training. We describe the limitations of this study and future works in Appendix A.

Acknowledgements

This work was supported by the Zhou Yahui Chair Professorship award of Tsinghua University (to CVC), the National High-Level Talent Program of the Ministry of Science and Technology of China (grant number 20241710001, to CVC).

References

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[2] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

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

[5] 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):1–12, 2018.

[6] 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.

[7] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 2020.

[8] 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.

[9] 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*, 2024.

[10] Anthony Holtmaat and Karel Svoboda. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nature Reviews Neuroscience*, 10(9):647–658, 2009.

[11] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific reports*, 3(1):1613, 2013.

[12] Simone Daminelli, Josephine Maria Thomas, Claudio Durán, and Carlo Vittorio Cannistraci. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics*, 17(11):113037, nov 2015.

[13] Claudio Durán, Simone Daminelli, Josephine M Thomas, V Joachim Haupt, Michael Schroeder, and Carlo Vittorio Cannistraci. Pioneering topological methods for network-based drug–target prediction by exploiting a brain-network self-organization theory. *Briefings in Bioinformatics*, 19(6):1183–1202, 04 2017.

[14] Carlo Vittorio Cannistraci. Modelling self-organization in complex networks via a brain-inspired network automata theory improves link reliability in protein interactomes. *Sci Rep*, 8(1):2045–2322, 10 2018.

[15] Vaibhav et al Narula. Can local-community-paradigm and epitopological learning enhance our understanding of how local brain connectivity is able to process, learn and memorize chronic pain? *Applied network science*, 2(1), 2017.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[17] 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)*, pages 2921–2926. IEEE, 2017.

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

[19] 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, 2017.

[20] 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*, pages 70–74. Association for Computational Linguistics, 2016.

[21] 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, editors, *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17, Lake Tahoe, California, December 4-5 2014.

[22] 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.

[23] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

[24] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2020.

[25] Ameya Prabhu, Girish Varma, and Anoop Namboodiri. Deep expander networks: Efficient deep networks from graph theory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–35, 2018.

[26] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[27] Tri Dao, Beidi Chen, Nimit S Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning*, pages 4690–4721. PMLR, 2022.

[28] James Stewart, Umberto Michieli, and Mete Ozay. Data-free model pruning at initialization via expanders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4518–4523, 2023.

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

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

[31] Donald Hebb. The organization of behavior. emphnew york, 1949.

[32] Alessandro Muscoloni, Umberto Michieli, Yingtao Zhang, and Carlo Vittorio Cannistraci. Adaptive network automata modelling of complex networks. *Preprints*, May 2022.

[33] Shiwei Liu, Lu Yin, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. In *International Conference on Machine Learning*, pages 6989–7000. PMLR, 2021.

[34] Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Elias Frantar and Dan Alistarh. Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*, 2023.

[36] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[37] 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.

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

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

[40] Maria Ercsey-Ravasz, Nikola T Markov, Christophe Lamy, David C Van Essen, Kenneth Knoblauch, Zoltán Toroczkai, and Henry Kennedy. A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron*, 80(1):184–197, 2013.

[41] Danielle S Bassett and Edward Bullmore. Small-world brain networks. *The Neuroscientist*, 12(6):512–523, 2006.

[42] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. *Advances in Neural Information Processing Systems*, 34:9908–9922, 2021.

[43] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.

[44] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[45] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[46] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection, 2024.

[47] Pengle Zhang, Jia Wei, Jintao Zhang, Jun Zhu, and Jianfei Chen. Accurate int8 training through dynamic block-level fallback. *arXiv preprint arXiv:2503.08040*, 2025.

[48] Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024.

[49] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[50] Vithursan Thangarasa, Abhay Gupta, William Marshall, Tianda Li, Kevin Leong, Dennis DeCoste, Sean Lie, and Shreyas Saxena. Spdf: Sparse pre-training and dense fine-tuning for large language models. In *Uncertainty in Artificial Intelligence*, pages 2134–2146. PMLR, 2023.

[51] Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Bill Nell, Nir Shavit, and Dan Alistarh. Inducing and exploiting activation sparsity for fast inference on deep neural networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5533–5543, Virtual, 13–18 Jul 2020. PMLR.

[52] P ERDdS and A R&wi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

[53] Ming Li, Run-Ran Liu, Linyuan Lü, Mao-Bin Hu, Shuqi Xu, and Yi-Cheng Zhang. Percolation on complex networks: Theory and application. *Physics Reports*, 907:1–68, 2021.

[54] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[55] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[56] Aleksandra I. Nowak, Bram Grooten, Decebal Constantin Mocanu, and Jacek Tabor. Fantastic weights and how to find them: Where to prune in dynamic sparse training, 2023.

Table 4: Zero-shot evaluation of LLaMA-130M between fully connected pretrained models, CHTs, and CHTss with 70% sparsity across GLUE and SuperGLUE datasets. MRPC and QQP use F1 scores while the others use ACC. s_i indicates the starting sparsity of the DST methods that use a density decay strategy. Results present a standard deviation across five seeds, taken with the library lm-eval [49].

Dataset	FC	CHTs	CHTss ($s_i = 0.5$)
CoLA	65.29 ± 1.47	**68.94 ± 1.43**	64.24 ± 1.48
MNLI	32.44 ± 0.47	**32.77 ± 0.47**	32.65 ± 0.47
MRPC	64.96 ± 2.36	**81.00 ± 1.65**	68.77 ± 2.23
QNLI	**50.38 ± 0.68**	49.59 ± 0.68	49.92 ± 0.68
QQP	52.09 ± 0.28	53.76 ± 0.26	**53.78 ± 0.26**
RTE	48.38 ± 3.01	44.40 ± 2.99	**49.46 ± 3.01**
SST-2	49.54 ± 1.69	**50.00 ± 1.69**	49.31 ± 1.69
WNLI	**49.30 ± 5.98**	43.66 ± 5.93	47.89 ± 5.97
Hellaswag	26.95 ± 0.44	26.85 ± 0.44	**26.96 ± 0.44**
Boolq	43.85 ± 0.87	**58.35 ± 0.86**	46.39 ± 0.87
CB	46.43 ± 6.72	**48.21 ± 6.74**	**48.21 ± 6.74**
Copa	**56.00 ± 4.99**	55.00 ± 5.00	55.00 ± 5.00
AVG	48.80	**51.68**	49.64
Win rate	25%	50%	31.7%

Table 5: **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}

A Limitations and Future Work

A potential limitation of this work is that the hardware required to accelerate sparse training with unstructured sparsity has not yet become widely adopted. Consequently, this article does not present a direct comparison of training speeds with those of fully connected networks. However, several leading companies [50, 51] have already released devices that support unstructured sparsity in training.

For future work, we aim to develop methods for automatically determining the temperature for soft sampling at each epoch, guided by the topological features of each layer. This could enable each layer to learn its specific topological rules autonomously. Additionally, we plan to test CHTs and CHTss in larger LLMs such as LLaMA-7b to evaluate the performance in scenarios with denser layers.

B Broader Impact

In this work, we introduce a novel methodology for dynamic sparse training aimed at enhancing the efficiency of AI model training. This advancement holds potential societal benefits by increasing interest in more efficient AI practices. However, the widespread availability of advanced artificial neural networks, particularly large language models (LLMs), also presents risks of misuse. It is essential to carefully consider and manage these factors to maximize benefits and minimize risks.

Table 6: **Hyperparameters of Transformer on Machine Translation Tasks.** `inoam` refers to a learning rate scheduler that incorporates iterative warm-up phases, specifically designed for dynamic sparse training (DST) methods. The purpose is to allow newly regrown connections to accumulate momentum, preventing potential harm to the training process. For the fully connected (FC) baseline, only the standard `noam` scheduler is used.

Hyper-parameter	**Multi30k**	**IWSLT14**	**WMT17**
Embedding Dimension	512	512	512
Feed-forward Dimension	1024	2048	2048
Batch Size	1024 tokens	10240 tokens	12000 tokens
Training Steps	5000	20000	80000
Dropout	0.1	0.1	0.1
Attention Dropout	0.1	0.1	0.1
Max Gradient Norm	0	0	0
Warmup Steps	1000	6000	8000
Learning rate Decay Method	inoam	inoam	inoam
Iterative warmup steps	20	20	20
Label Smoothing	0.1	0.1	0.1
Layer Number	6	6	6
Head Number	8	8	8
Learning Rate	0.25	2	2
ζ (fraction of removal)	0.3	0.3	0.3
Update Interval (for DST)	100 steps	100 steps	100 steps

Table 7: **Hyperparameters of LLaMA-60M, LLaMA-130M, and LLaMA-1B on OpenWebText.** `inoam` refers to a learning rate scheduler that incorporates iterative warm-up phases, specifically designed for dynamic sparse training (DST) methods. The purpose is to allow newly regrown connections to accumulate momentum, preventing potential harm to the training process. For the fully connected (FC) baseline, only the standard `noam` scheduler is used.

Hyper-parameter	**LLaMA-60M**	**LLaMA-130M**	**LLaMA-1B**
Embedding Dimension	512	768	2048
Feed-forward Dimension	1376	2048	5461
Global Batch Size	512	512	512
Sequence Length	256	256	256
Training Steps	10000	30000	100000
Learning Rate	3e-3 (1e-3 for FC)	3e-3 (1e-3 for FC)	3e-3 (4e-4 for FC)
Warmup Steps	1000	10000	10000
Learning rate Decay Method	inoam	inoam	inoam
Iterative warmup steps	20	20	20
Optimizer	Adam	Adam	Adam
Layer Number	8	12	24
Head Number	8	12	32
ζ (fraction of removal)	0.1	0.1	0.1
Update Interval (for DST)	100 steps	100 steps	100 steps

C Cannistraci-Hebb epitopological rationale

The original CHT framework leverages the Cannistraci-Hebb link predictor on Length 3 paths (CH3-L3p) metric for link regrowth. Given two seed nodes u and v in a network, this metric assigns a score

$$\textbf{CH3-L3p}(u,v) = \sum_{z_1,z_2 \in L3} \frac{1}{\sqrt{de^*_{z_1} \cdot de^*_{z_2}}} \tag{5}$$

Here, u and v denote the seed nodes, while z_1 and z_2 are common neighbors on the L3 path [32], a walk of three consecutive links that connects u to v via those two intermediate nodes. The term de^*_i represents the number of external local community links (eLCL) of node i, with a default increment

Table 8: Performance comparison of different dynamic sparse training methods on MNIST, Fashion MNIST (FMNIST), and EMNIST datasets trained on MLP at 99% sparsity. ACC represents accuracy, and ANP denotes the active neuron percolation rate, indicating the final size of the network. Accuracies present a standard error taken over three seeds. The best dynamic sparse training method for each dataset is highlighted in bold, and the performances that surpass the fully connected model are marked with "*".

Method	MNIST		FMNIST		EMNIST	
	ACC (%)	ANP	ACC (%)	ANP	ACC (%)	ANP
FC	98.78 ± 0.02	–	90.88 ± 0.02	–	87.13 ± 0.04	–
CHTs (CH3-L3p)	**98.81 ± 0.04***	20%	**90.93 ± 0.03***	89%	87.61 ± 0.07*	24%
CHTs (CH2-L3n)	98.76 ± 0.05	27%	90.67 ± 0.05	73%	**87.82 ± 0.04***	28%
CHT	98.48 ± 0.04	29%	88.70 ± 0.07	30%	86.35 ± 0.08	21%
RigL	98.61 ± 0.01	29%	89.91 ± 0.07	55%	86.94 ± 0.08	28%
SET	98.14 ± 0.02	100%	89.00 ± 0.09	100%	86.31 ± 0.08	100%

Table 9: Perplexity (PPL) results across different sparsities (0.7, 0.8, 0.9, 0.95) for CHTs and CHTss under different regrowth strategies (Fixed and Uniform) and r settings on LLaMA60M.

		Fixed				**Uniform**			
	Sparsity	$r = 0.0$	$r = 0.1$	$r = 0.2$	$r = 0.3$	$r = 0.0$	$r = 0.1$	$r = 0.2$	$r = 0.3$
CHTs	70%	28.16	28.39	28.25	28.32	30.11	**28.12**	28.43	28.56
	80%	30.22	**29.84**	30.04	30.03	30.19	29.86	30.23	30.06
	90%	33.32	33.37	**33.03**	33.77	33.45	33.36	33.88	33.72
	95%	37.29	37.51	37.24	37.46	37.23	**36.47**	37.33	37.67
CHTss	70%	**27.62**	30.05	27.82	28.43	**27.62**	27.74	27.74	27.68
	80%	**29.00**	**29.00**	29.66	32.91	29.49	29.69	29.09	29.24
	90%	31.51	31.67	31.65	31.59	31.66	32.61	31.68	**31.42**
	95%	38.66	35.31	36.24	37.50	42.20	37.40	35.36	**35.10**

of 1 to prevent division by zero. Path-based link prediction has demonstrated its effectiveness on both real-world networks [32] and artificial neural networks [9]. However, this method incurs a high computational cost due to the need to compute and store all length-three paths, resulting in a time complexity of $O(N \cdot d^3)$, where N is the number of nodes and d is the network's average degree. This complexity is prohibitive for large models with numerous nodes and higher-density layers. To address this issue, we introduce a more efficient, node-based paradigm that eliminates the reliance on length-three paths between seed nodes. Instead, this approach focuses on the common neighbors of seed nodes. The node-based version of CH3-L3p, denoted as CH2-L3n, is defined as follows:

$$\textbf{CH2-L3n}(u, v) = \sum_{z \in L3} \frac{di_z^*}{de_z^*} \tag{6}$$

Here, u and v denote the seed nodes, while z is the common neighbor on the L3 path [32], a walk of three consecutive links that connects u to v via two of those intermediate nodes. The terms di_z^* de_z^* represent the number of internal local community links (iLCLs) and external local community links (eLCLs) of node i, with a default increment of 1 to prevent division by zero. Internal local community links (iLCLs) are those that connect nodes belonging to the same local community. Contrarily, external local community links (eLCLs) connect nodes belonging to different communities. Figure 4 gives a visual representation of L2 and L3 paths between two seed nodes u and v, defining their local community.

D Sparse topological initialization

Correlated sparse topological initialization. Correlated Sparse Topological Initialization (CSTI) is a physics-informed topological initialization. CSTI generates the adjacency matrix by computing

Table 10: Perplexity (PPL) results across different sparsities (0.7, 0.8, 0.9, 0.95) for CHTs and CHTss under different regrowth strategies (Fixed and Uniform) and r settings on LLaMA-130M.

		Fixed				**Uniform**			
	Sparsity	$r = 0.0$	$r = 0.1$	$r = 0.2$	$r = 0.3$	$r = 0.0$	$r = 0.1$	$r = 0.2$	$r = 0.3$
CHTs	70%	20.24	20.16	**20.10**	20.25	20.62	20.18	20.15	20.20
	80%	**21.33**	21.37	21.36	21.48	21.34	21.40	21.40	22.49
	90%	23.72	23.76	23.76	23.94	23.74	23.73	**23.71**	24.99
	95%	28.05	**26.45**	26.90	26.91	26.78	27.97	29.05	27.10
CHTss	70%	20.63	19.88	19.93	**19.85**	21.43	19.90	20.93	19.94
	80%	20.71	22.60	20.86	**20.70**	20.73	20.74	20.72	20.82
	90%	22.58	22.72	22.61	**22.51**	22.53	22.59	22.60	23.12
	95%	25.28	25.12	25.20	25.12	**25.07**	25.15	25.23	25.12

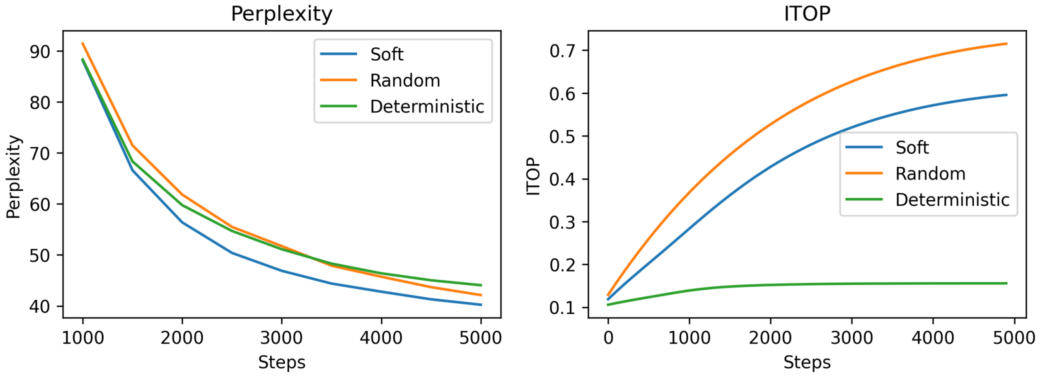

Figure 3: Comparison of link regrowth strategies in CHTs using a LLaMA-60M model trained on OpenWebText for 5000 steps. The left plot shows validation perplexity (lower is better), while the right plot reports the in-time over-parameterization (ITOP) rate, which measures the cumulative proportion of links activated during training. Results are presented for three strategies: Soft, Random, and Deterministic regrowth.

the Pearson correlation between each input feature across the calibration dataset and then selects the predetermined number of links, calculated based on the desired sparsity level, as the existing connections. CSTI performs remarkably better when the layer can directly receive input information. However, for layers that cannot receive inputs directly, it cannot capture the correlations from the start since the model is initialized randomly, as in the case of the Transformer. Therefore, in this article, we aim to address this issue by investigating different network models to initialize the topology, to improve the performance for cases where CSTI cannot be directly applied.

Bipartite scale-free model. In artificial neural networks (ANNs), fully connected networks are inherently bipartite. This article explores initializing bipartite networks using models from network science. The Bipartite Scale-Free (BSF) [9] network model extends the concept of scale-freeness to bipartite structures, making them suitable for dynamic sparse training. Initially, the BSF model generates a monopartite Barabási-Albert (BA) model [39], a well-established method for creating scale-free networks in which the degree distribution follows a power law (γ=2.76 in Figure 5). Following the creation of the BA model, the BSF approach removes any connections between nodes of the same type (neuron in the same layer) and rewires these connections to nodes of the opposite type (neuron in the opposite layer). This rewiring is done while maintaining the degree of each node constant to preserve the power-law exponent γ.

Bipartite small-world model. The Bipartite Small-World (BSW) network model [9] is designed to incorporate small-world properties and a high clustering coefficient into bipartite networks. Initially, the model constructs a regular ring lattice and assigns two distinct types of nodes to it. Each node is connected by an equal number of links to the nearest nodes of the opposite type, fostering high

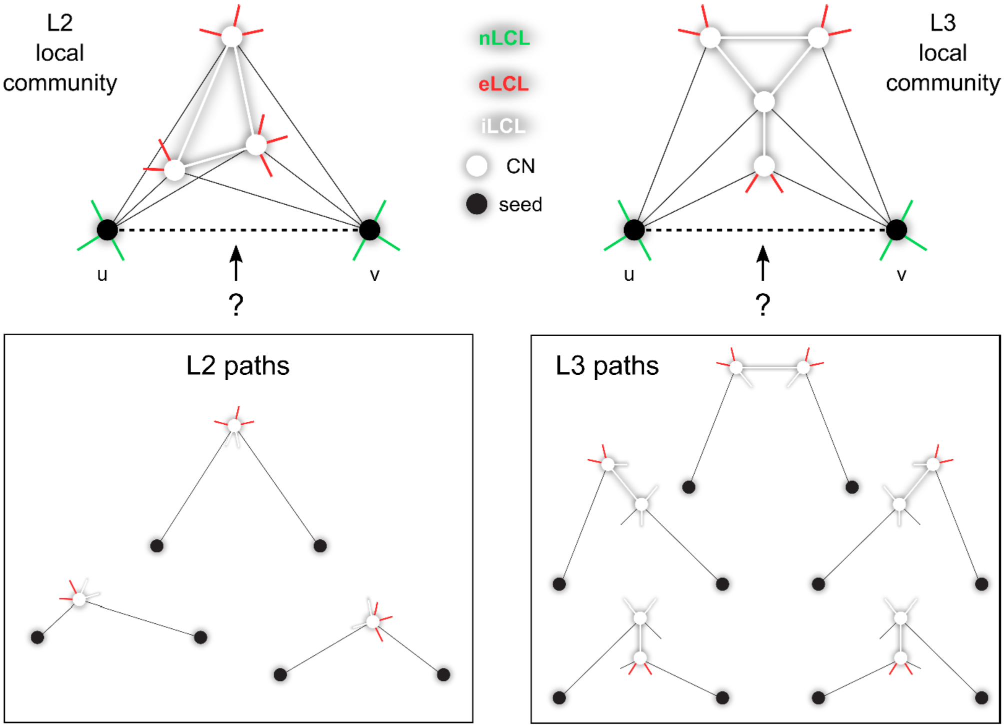

Figure 4: **Cannistraci-Hebb epitopological rationale.** [32] The figure illustrates an explanatory example of topological link prediction using the Cannistraci-Hebb epitopological rationale based on either L2 or L3 paths. The two black nodes represent the seed nodes whose unobserved interaction is to be assigned a likelihood score. White nodes denote the common neighbours (CNs) of the seed nodes at either L2 or L3 distance. Together, the set of CNs and the internal local community links (iLCL) constitute the local community. Different link types are color-coded: green for nLCLs, red for external local community links (eLCLs), and white for iLCLs. The L2 (path length 2) and L3 (path length 3) paths associated with the illustrated communities are highlighted. Notably, in artificial neural networks (ANNs), linear layers correspond to bipartite networks, which inherently support only L3 path predictions, as shown in Figure 1.

clustering but lacking the small-world property. Similar to the Watts-Strogatz model (WS) [38], the BSW model introduces a rewiring parameter, β, which represents the percentage of links randomly removed and then rewired within the network. At $\beta = 1$, the model transitions into an **Erdős-Rényi model** [52], exhibiting small-world properties but without a high clustering coefficient, which is popular as the topological initialization of the other dynamic sparse training methods [5, 7, 8].

Bipartite receptive field model. The Bipartite Receptive Field (BRF) model is a random network generation technique designed to mimic the receptive field phenomenon in the brain networks. The process involves adding links to the adjacency matrix of the bipartite network, with the connectivity structured around the main diagonal according to a parameter $r \in [0, 1]$. A low value of r results in links that are primarily clustered around the diagonal, while a higher value of r leads to a more random connectivity pattern. Specifically, a bipartite adjacency matrix with links near the diagonal indicates that adjacent nodes from the two layers are linked, whereas links far from the diagonal correspond to more distant node pairs. 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 of the matrix $m_{i,j}$ is set to 1 if input node i is connected to output node j, and 0 otherwise. A scoring function $S_{i,j}$ is assigned to each connection in the adjacency matrix based on its distance to the main diagonal. This score is given by:

$$S_{i,j} = d_{ij}^{\frac{1-r}{r}}, \tag{7}$$

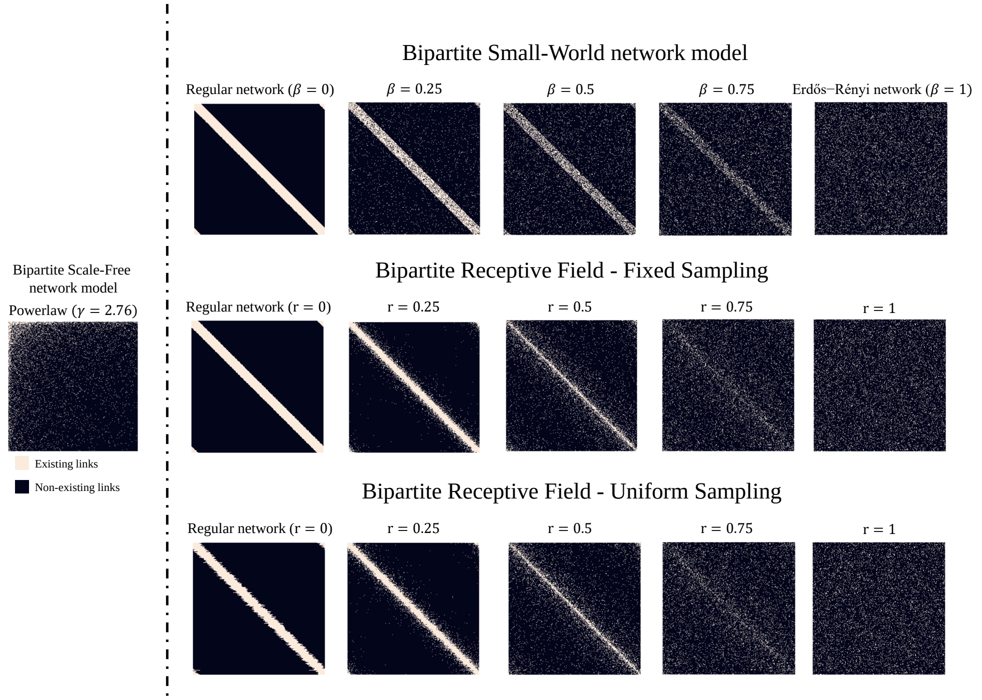

Figure 5: **The adjacency matrix** of the Bipartite Scale-Free (BSF) network model compared to those of the Bipartite Small-World (BSW) network, the Bipartite Receptive Field with fixed sampling (BRF$_f$), and the Bipartite Receptive field with uniform sampling (BRF$_u$) as parameters β and r vary between 0 and 1. a) The BSF model inherently forms a scale-free network characterized by a power-law distribution with $\gamma = 2.76$. b) As β changes from 0 to 1, the network exhibits reduced clustering. It is important to note that when $\beta = 0$, the BSW model does not qualify as a small-world network. c) As r increases towards 1, the adjacency matrix becomes more random, while sampling the output neurons' degrees from a fixed or uniform distribution.

where

$$d_{ij} = min\{|i-j|, |(i-M)-j|, |i-(j-N)|\} \tag{8}$$

is the distance between the input and output neurons. Therefore, $S_{i,j}$ represents the distance from the diagonal, raised to the power of $\frac{1-r}{r}$. The parameter r controls how structured or random the adjacency matrix is. As $r \to 0$, the scoring function becomes more deterministic, with high scores assigned to entries near the diagonal and low scores to entries farther away. Conversely, as $r \to 1$, all scores $S_{i,j}$ become more uniform, leading to a more random, less structured adjacency matrix. The next step is to determine the degree distribution for the output nodes. This can either be fixed, assigning the same degree to all output nodes, or uniform, where the degrees are randomly sampled from a uniform distribution. Hence, we propose two variations of the BRF model: the Bipartite Receptive Field with fixed sampling (BRFf), in which the degrees of output nodes are fixed, and the Bipartite Receptive Field with uniform sampling (BRFu), where the degrees of the output nodes follow a uniform distribution. This represents an additional enhancement to the WS scheme, which offers no way to control how connections are allocated among the output nodes. In conclusion, to run the BRF model, the user should input an output degree distribution and a spatial dependent distance randomness.

E Equal Partition and Neuron Resorting to enhance bipartite scale-free network initialization

As indicated in SET and CHT [5, 9], trained sparse models typically converge to a scale-free network. This suggests that initiating the network with a scale-free structure might initially enhance performance. However, starting directly with a Bipartite Scale-Free model (BSF, power-law exponent $\gamma = 2.76$) does not yield effective results. Upon deeper examination, two potential reasons emerge:

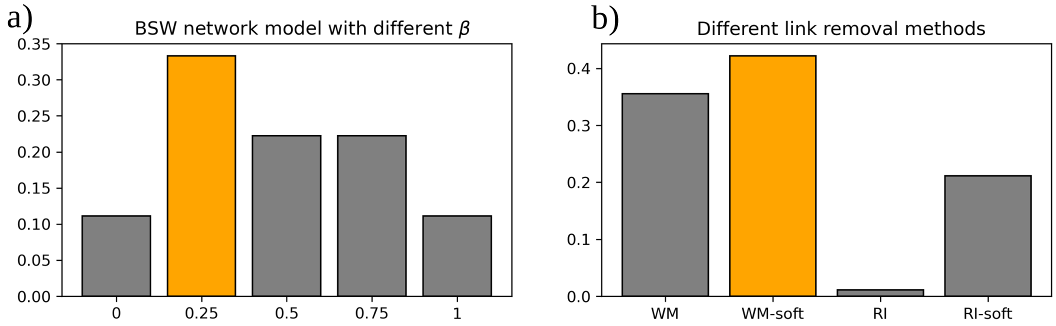

Figure 6: **The ablation test** of the β of the bipartite small world (BSW) model and the removal methods in CHTs. a) evaluates the influence of the rewiring rate β on the model performance when initialized with the BSW model. b) assesses the influence of link removal selecting from the weight magnitude (WM), weight magnitude soft (WM-soft), relative importance (RI), and relative importance soft (RI-soft). We utilize the win rate of the compared factors under the same setting across each realization of 3 seeds for all experiment combinations on MLP. The factor with the highest win rate is highlighted in orange.

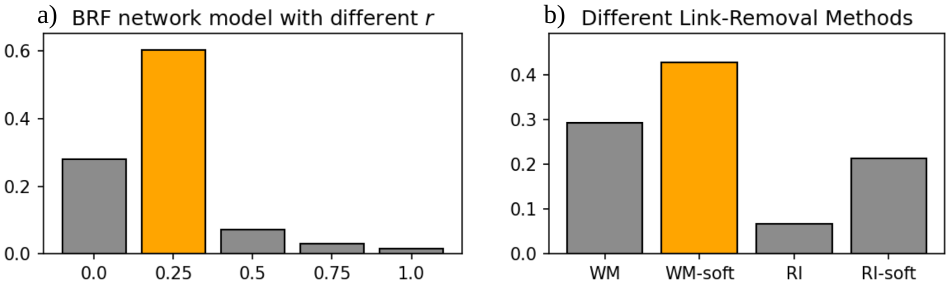

Figure 7: **The ablation test of** the parameter r in the bipartite receptive field (BRF) model and the removal methods in CHTs using the BRF initialization technique. a) evaluates the influence of the parameter r on the model performance when initialized with the BRF model. b) assesses the influence of link removal in the CHTs model with BRF initialization. We utilize the win rate of the compared factors under the same setting across each realization of 3 seeds for all experiment combinations on MLP. The factor with the highest win rate is highlighted in orange.

- The BSF model generates hub nodes randomly. However, this random assignment of hub nodes to less significant inputs leads to a less effective initialization, which is particularly detrimental in CHT, which merely utilizes the topology information to regrow new links.
- As demonstrated in CHT, in the final network, the hub nodes of one layer's output should correspond to the input layer of the subsequent layer, which means the hub nodes should have a high degree on both sides of the layer. However, the BSF model's random selection disrupts this correspondence, significantly reducing the number of Credit Assignment Paths (CAP) [9] in the model. CAP is defined as the chain of transformation from input to output, which counts the number of links that go through the hub nodes in the middle layers.

To address these issues, we propose two solutions:

- Equal Partitioning of the First Layer: We begin by generating a BSF model, then rewire the connections from the input layer to the first hidden layer. While keeping the out-degrees of the output neurons fixed, we randomly sample new connections to the input neurons until each of the input neurons' in-degrees reaches the input layer's average in-degree. This approach ensures all input neurons are assigned equal importance while maintaining the power-law degree distribution of output neurons.

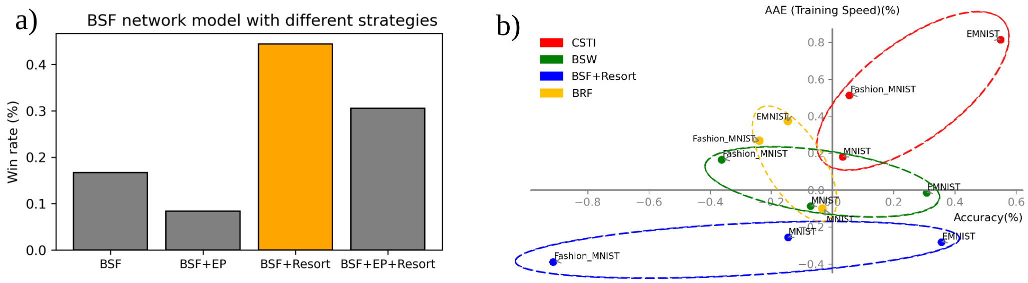

Figure 8: **The Performance** of the bipartite scale-free model and two enhanced techniques. a) shows the win rate of the Bipartite Scale-Free network model (BSF) with the different techniques. *EP* stands for equal partition of the first layer, and *Resort* refers to reordering the neurons based on their degree. b) assesses the comparison between Correlated Sparse Topological Initialization (CSTI), the Bipartite Scale-Free (BSF) model with the best solution from a), and the Bipartite Small-World (BSW) model with $\beta = 0.25$.

- Resorting Middle Layer Neurons: Given the mismatch in hub nodes between consecutive layers, we suggest permuting the neurons between the output of one layer and the input of the next, based on their degree. A higher degree in an output neuron increases the likelihood of connecting to a high-degree input neuron in the subsequent layer, thus enhancing the number of CAPs.

As illustrated in Figure 8, while the two techniques enhance the performance of the BSF initialization, they remain inferior to the BSW initialization. As noted in the main text, achieving scale-freeness is more effective when the model is allowed to learn and adapt dynamically rather than being directly initialized as a predefined structure.

F Network percolation and extension to Transformer.

We have adapted network percolation [53, 9] to suit the architecture of the Transformer after link removal. The core idea is to identify inactive neurons, which are characterized by having no connections on either one or both sides within a layer of neurons. Such neurons disrupt the flow of information during forward propagation or backpropagation. In addition, Layer-wise computation of the CH link prediction score further implies that neurons without connections on one side are unlikely to form connections in the future. Therefore, network percolation becomes essential to optimize the use of remaining links.

As shown in Figure 1, network percolation encompasses two primary processes: c1) inactive neuron removal to remove the neurons that lack connections on one or both sides; c2) incomplete path adjustment to remove the incomplete paths where links connect to the inactive neurons after c1). Typically applied in simpler continuous layers like those in an MLP, network percolation requires modification for more complex structures. For example, within the Transformer's self-attention module, the outputs of the query and key layers undergo a dot product operation. It necessitates percolation in these layers to examine the activity of the neurons in both output layers at the same position. Similar interventions are necessary in the up_proj and gate_proj layers of the MLP module in the LLaMA model family [1, 54].

G Baseline Methods

G.1 Fixed Density Dynamic Sparse Training Methods

SET [5]: Removes connections based on weight magnitude and randomly regrows new links.

RigL [7]: Removes connections based on weight magnitude and regrows links using gradient information, gradually reducing the proportion of updated connections over time.

CHT [9]: A state-of-the-art (SOTA) gradient-free method that removes links with weight magnitude and regrows links based on CH3-L3 scores. CHT is often applied with early stopping to mitigate its computational complexity when working with large models.

G.2 Gradual Density Decrease Dynamic Sparse Training Methods

GMP [55, 43]: Prunes the network with weight magnitude and gradually decreases the density based on Equation 10. Although originally a pruning method, GMP is treated as a dynamic sparse training method in their implementation [43], as it stores historical weights and allows pruned weights to reappear during training, since, during training, the pruning threshold might change.

MEST$_{EM\&S}$ [8]: Implements a two-stage density decrease strategy as described in the original work. It removes links based on the combination of weight magnitude and 0.01*gradient and regrows new links randomly.

GraNet [42]: Gradually decreases density using Equation 10. Similar to RigL, GraNet removes links based on the weight magnitude and regrows new links with the gradient of the existing links.

Table 11: Float32 Precision Comparison on LLaMA-130M. Bold values denote the best performance among DST methods. Lower perplexity corresponds to better model performance. s_i represents the initial sparsity for DST methods employing a density decay strategy.

Method	Sparsity	
	70%	80%
FC	17.07	
RigL	18.34	19.64
CHTs	17.99	19.25
GraNet ($s_i = 0.5$)	17.92	18.79
CHTss ($s_i = 0.5$)	**17.76**	**18.69**

H Ablation and Sensitivity Tests

An overall ablation test To fully assess each component's effectiveness, we conduct several ablation and sensitivity tests that help us understand how to select a sparse topological initialization and identify the best link removal and regrowth methods. We first made a global test for all the components in Table 12, which shows the effectiveness of each element introduced by this article. The node-based and path-based link regrowth methods have comparable performance, but the node-based versions are much faster.

Sparse topological initialization. For sparse topological initialization, we compare BRF, BSW, BSF, and CSTI [9] across three image classification datasets, as shown in Figure 8b. The results indicate that when the inputs can directly access task-relevant information, CSTI consistently achieves the best performance. In general, BRF and BSW perform similarly under these conditions, but outperform the BSF initialization.

To further validate our findings, we evaluate BRF and BSW network initializations on machine translation tasks using Transformer models. Figure 9 and Figure 10 present the performance comparisons between BSW and BRF on the Multi30k and IWSLT datasets, while Figure 11 shows the win-rate analysis. These comparisons demonstrate that BRF consistently outperforms BSW across most cases. Additionally, Figure 7a analyzes the impact of the receptive field range r on BRF initialization for MNIST, Fashion MNIST, and EMNIST tasks using MLPs, with results indicating that $r = 0.25$ yields the best performance.

Building on this prior knowledge, we further evaluate BRF on LLaMA-60M and LLaMA-130M models, testing r values in the range $[0, 0.3]$ and comparing two different degree distributions. The results, shown in Table 9 and Table 10, indicate that on LLaMA models, the choice of r and distribution has limited impact. While $r = 0.1$ wins slightly more often, the improvements remain

Table 12: Ablation results of Transformer on Multi30K and IWSLT datasets at 90% sparsity. The scores indicate BLEU scores, the higher the better. Bold values denote the best performance among DST methods.

Variant	Multi30K (90% sparsity)	IWSLT (90% sparsity)
a. CHT	28.38	19.91
b. CHTss without node-based implementation	32.68 (2.42 hours)	**24.82** (18 hours)
c. CHTss without soft sampling	28.92	21.88
d. CHTss without sigmoid decay (= CHTs)	30.35	21.60
e. CHTss (full model)	**32.79** (0.25 hours)	24.57 (1.5 hours)

marginal. Finally, Table 3 reports the best performance combinations of r and degree distributions derived from these evaluations.

Table 13: Performance comparison of CHTs and CHTss at 90% sparsity across different removal methods. The tested dataset is Multi30K, and the reported metric is BLEU, which is the higher the better.

Remove Method	CHTs	CHTss
set	28.82	25.76
wm	28.17	31.15
wm_soft	**30.35**	**32.79**
ri	28.91	32.20
ri_soft	27.86	31.86
MEST	28.70	32.07
snip	28.23	31.66
sensitivity	29.02	29.73
Rsensitivity	28.18	30.67

Link removal. We first conduct a simple evaluation of the link removal methods introduced in this article when changing the α and δ inside Figure 1b2) on Figure 6b) and Figure 7b). The removal methods are selected from Weight Magnitude (WM), Weight Magnitude soft (WMs), Relative Importance (RI), and Relative Importance soft (RIs). For WM we fix the hyperparameters $\alpha = 1$ and $\delta = 1$; for RI we fix the hyperparameters $\alpha = 0$ and $\delta = 0.5$; for WMs we fix $\alpha = 1$ and we let δ increase linearly from 0.5 to 0.9; for RIs we fix $\alpha = 0$ and let δ increase linearly from 0.5 to 0.9. From the results, it can be observed that WMs performs the best in most cases. We compare these methods with those in [56] in Table 13 on two machine translation tasks. The results indicate that using WMs as a link removal method generally outperforms the alternatives.

We also evaluate how to define the softness in WMs. During sampling, we have a hyperparameter to decide the temperature of the scores that convert to the probability of being removed. We perform a test using a linear decay solution, since, generally, the weights in the model become more reliable as training progresses. Figure 12 shows the variation in BLEU scores as we change the starting and ending values of the δ parameter in the soft weight magnitude removal method on transformer models. Recalling that we define the temperature by $T = \frac{1}{1-\delta}$, we observe that for a simple benchmark like Multi30k, a high starting temperature produces better performance. This is motivated by the fact that loss decreases very fast through epochs, meaning that weights are learned quickly, and we can deterministically remove weights with high reliability. In more complex datasets, like IWSLT, low starting temperatures are preferred. This is because during the early stages of training, weights are learned slowly, meaning that a deterministic removal can be less reliable. To be more consistent, we select a start $\delta = 0.5$ and end $\delta = 0.9$ for all the tasks in the main article.

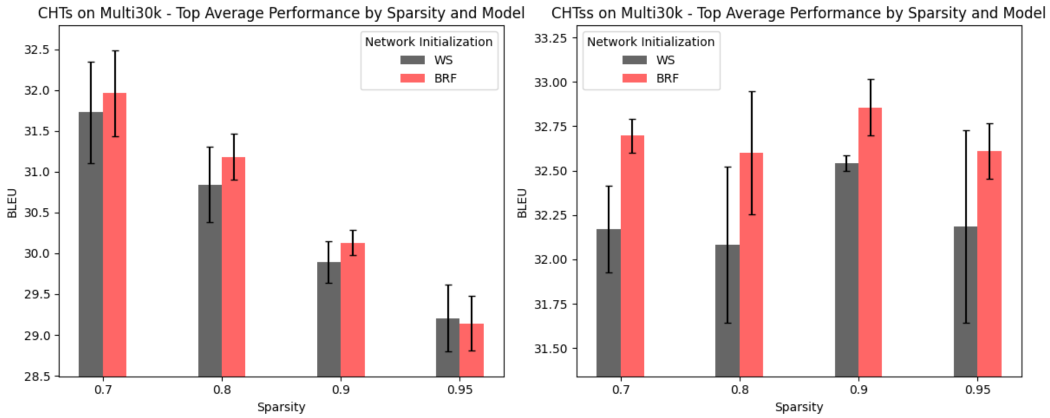

Figure 9: Top average BLEU for WS (grey) and BRF initialization methods on the Multi30k translation dataset of CHTs (left) and CHTss (right, with sigmoid decay) at different sparsity levels. Error bars denote the standard error across three seeds.

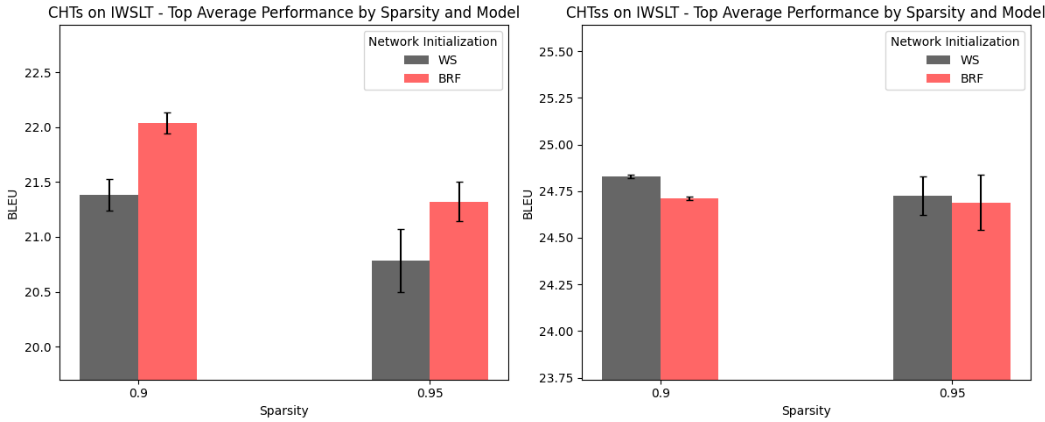

Figure 10: Top average BLEU for WS (grey) and BRF initialization methods on the IWSLT translation dataset of CHTs (left) and CHTss (right, with sigmoid decay) at different sparsity levels. Error bars denote the standard error across two seeds.

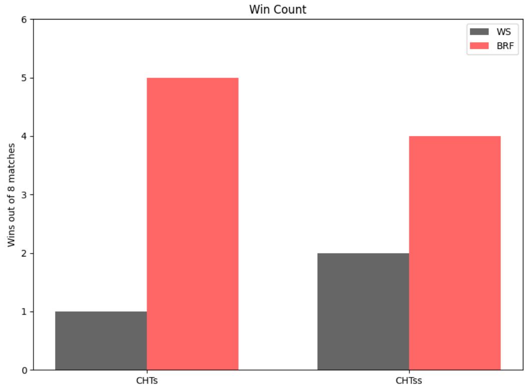

Figure 11: Win rates of BRF against WS over CHTs and CHTss models on different datasets (Multi30k and IWSLT) and different sparsities (0.9 and 0.95 for IWSLT and 0.7, 0.8, 0.9, 0.95 for Multi30k).

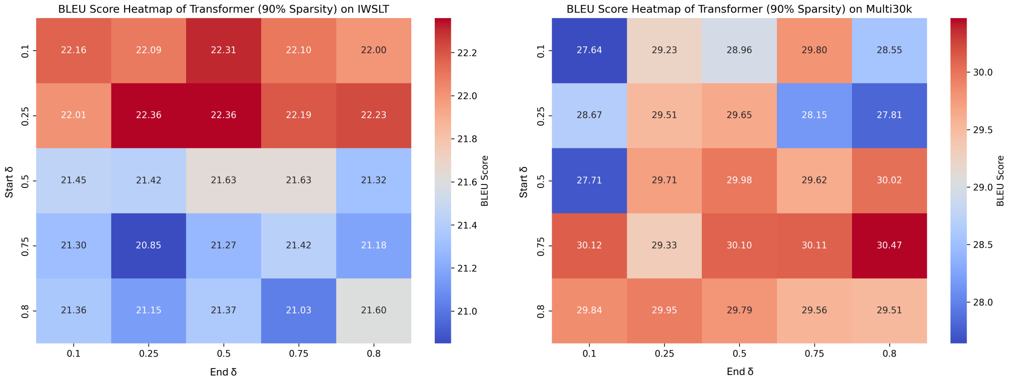

Figure 12: **Investigating the level of randomness in link removal strategies.** Top BLEU scores of the transformer model using CHTs with weight magnitude soft removal strategy, as the initial and final values of δ take values in $\{0.1, 0.25, 0.5, 0.75, 0.8\}$.

I Integral of Sigmoid Density Decay and Cubic Density Decay

In this section, we show the formula for the proposed Sigmoid Density Decay and the Cubic decay implemented in GraNet [42] and GMP [55].

For the Cubic function, it is formulated as:

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3, \tag{9}$$

where $t \in \{t_0, t_0 + \Delta t, \ldots, t_0 + n\Delta t\}$, s_i is the initial sparsity, s_f is the target sparsity, t_0 is the starting epoch of gradual pruning, t_f is the end epoch of gradual pruning, and Δt is the pruning frequency.

Since our work focuses on MLP, Transformer, and LLMs, where FLOPs are linearly related to the density of the linear layers, the FLOPs of the whole training process are linearly related to the integral of the density function across the training time. The integral of the cubic decay function from t_0 to t_f is:

$$\begin{aligned} &\int_{t_0}^{t_f} (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3 dt \\ &= \frac{1}{4}(s_i - s_f)(t_f - t_0). \end{aligned} \tag{10}$$

For the sigmoid decrease, the integral is:

$$\begin{aligned} &\int_{t_0}^{t'_f} (s_i - s_f)\left(\frac{1}{1 + e^{-k\left(t - \frac{t'_f + t_0}{2}\right)}}\right) dt \\ &= \frac{(s_i - s_f)(t'_f - t_0)}{2}. \end{aligned} \tag{11}$$

To maintain consistency in the computational cost (FLOPs) during training compared to the cubic decay strategy, we reduce the number of steps in the sigmoid-based gradual density decrease by half.

J Historical weights

Inspired by GMP [55, 43], we incorporate historical weights into our CHTs and CHTss implementation. During training, we maintain a historical weight matrix that records previously learned weights throughout the training process. When CHTs and CHTss predict new links, we initialize them using their corresponding historical weights - specifically, the values they held before being pruned. In this way, CHTs and CHTss enable weight recovery with preserved memory, allowing the model to retain valuable prior information.

K Pseudo code of node-based CH link predictor

We present the pseudocode for the node-based CH2-L3 regrowth method in Algorithm 1, which comprises three main steps.

Step 1: We compute the sets of path length-2 (L2) neighbors from nodes U to U and from V to V. The computational complexity of this step is

$$\mathcal{O}(m \langle d_m d_n \rangle + n \langle d_m d_n \rangle),$$

where $\langle d_m d_n \rangle$ denotes the average product of degrees d_m and d_n across all relevant node pairs. In the case of an ultra-sparse network (e.g., with average degree close to 1), this simplifies to $\mathcal{O}(m+n)$.

Step 2: We treat each destination node along the UU and VV paths as a potential common neighbor and compute the CH2-L3 score according to Equation (3) in the main text. This step has a time complexity of

$$\mathcal{O}(m^2 + n^2).$$

Step 3: We aggregate the CH2-L3 scores across all hop-3 nodes. This step requires

$$\mathcal{O}(mn \langle d_m \rangle + nm \langle d_n \rangle)$$

time. Under the ultra-sparse assumption, this further reduces to $\mathcal{O}(mn + nm)$.

Combining all steps, the overall time complexity of the node-based CH2-L3 regrowth procedure is

$$\mathcal{O}(mn \langle d_m \rangle + nm \langle d_n \rangle).$$

Since all operations can be implemented in a matrix-wise manner leveraging GPU acceleration assuming all matrices as dense, the total time complexity becomes

$$\mathcal{O}(nm^2 + mn^2).$$

L Extra results of LLaMA1b

Table 14: **Validation perplexity of different dynamic sparse training (DST) methods on OpenWebText using LLaMA-1B across varying sparsity levels.**. Lower perplexity corresponds to better model performance. The performances that surpass the fully connected model are marked with “*".

Sparsity	**0.7**	**0.9**	**0.95**
FC		14.62	
CHTs	14.53*	17.14	18.93
CHTss	15.15	15.62	16.51

Language modeling. We present a comparison of CHTs, CHTss, and fully connected network on language modeling tasks using the LLaMA-1B model on Table 14. The results clearly demonstrate that CHTs consistently outperform the fully connected (FC) baseline at 70%, even at a high sparsity of 95%, CHTss achieves a perplexity of 16.51, which is remarkably close to the FC baseline.

Zero-shot performance. We also compare the performance of CHTs, CHTss, and the fully connected network on zero-shot tasks, as summarized in Table 15. The results demonstrate that CHTs generally outperforms the fully connected baseline at 90% and 95% sparsity levels. This suggests that CHTs enhances the model’s generalization capability on previously unseen data.

M Experiments compute resources

All experiments were conducted on NVIDIA A100 80GB GPUs. MLP and Transformer models were trained using a single GPU, while LLaMA models were trained using eight GPUs in parallel.

Algorithm 1 CH2–L3n

```
Require: Binary bipartite adjacency matrix $A_{UV} \in \{0,1\}^{m \times n}$        ▷ $U$–to–$V$ edges
Ensure: Score matrix $S \in \mathbb{R}^{m \times n}$ with $S[i,j] > 0$ only if $A_{UV}[i,j] = 0$
function CH2-L3N($A_{UV}$)
    $A_{VU} \leftarrow A_{UV}^{\top}$                                          ▷ $V \to U$ edges
    $d_U \leftarrow \text{row-sum}(A_{UV})$
    $d_V \leftarrow \text{row-sum}(A_{VU})$
                                  ▷ Step 1: two–step paths inside each partition
    $UU \leftarrow A_{UV} A_{VU}$                                       ▷ $U \to V \to U$
    $VV \leftarrow A_{VU} A_{UV}$                                       ▷ $V \to U \to V$
                                  ▷ Step 2: CH2-L3n preparatory scores
    init $e_{UU} \leftarrow 0_{m \times m}$, $e_{VV} \leftarrow 0_{n \times n}$
    for $i \leftarrow 1$ to $m$ do
        for $j \leftarrow 1$ to $m$ s.t. $j \neq i$ and $UU[i,j] > 0$ do
            $ext \leftarrow d_U[j] - UU[i,j] - 1$
            $e_{UU}[i,j] \leftarrow \dfrac{UU[i,j] + 1}{ext + 1}$                ▷ # According to Equation (3)
        end for
    end for
    for $a \leftarrow 1$ to $n$ do
        for $b \leftarrow 1$ to $n$ s.t. $b \neq a$ and $VV[a,b] > 0$ do
            $ext \leftarrow d_V[b] - VV[a,b] - 1$
            $e_{VV}[a,b] \leftarrow \dfrac{VV[a,b] + 1}{ext + 1}$                ▷ # According to Equation (3)
        end for
    end for
                                  ▷ Step 3: final CH2–L3n scores
    init $S \leftarrow 0_{m \times n}$
    for $i \leftarrow 1$ to $m$ do
        for $a \leftarrow 1$ to $n$ s.t. $A_{UV}[i,a] = 0$ do
            $S_{UV} \leftarrow \sum_{j=1}^{m} e_{UU}[i,j]\, A_{UV}[j,a]$
            $S_{VU} \leftarrow \sum_{b=1}^{n} e_{VV}[a,b]\, A_{VU}[b,i]$
            $S[i,a] \leftarrow S_{UV} + S_{VU}$
        end for
    end for
    return $S$
end function
```

Table 15: Zero-shot evaluation of LLaMA-1B across GLUE and SuperGLUE. ACC scores are shown. Values are mean ± sd over 5 seeds (lm-eval). Red values indicate the top scores over models and sparsities for a benchmark.

Dataset	FC	70 % sparsity		90 % sparsity		95 % sparsity	
		CHTs	CHTss	CHTs	CHTss	CHTs	CHTss
CoLA	40.27 ± 1.52	38.83 ± 1.51	44.20 ± 1.54	50.53 ± 1.55	67.88 ± 1.45	68.84 ± 1.43	31.83 ± 1.44
MNLI	32.89 ± 0.47	32.77 ± 0.47	32.65 ± 0.47	32.83 ± 0.47	32.73 ± 0.47	32.91 ± 0.47	32.68 ± 0.47
MRPC	39.95 ± 2.43	40.44 ± 2.43	36.03 ± 2.38	66.42 ± 2.34	38.48 ± 2.41	67.65 ± 2.32	46.81 ± 2.47
QNLI	49.95 ± 0.68	50.96 ± 0.68	49.70 ± 0.68	50.36 ± 0.68	49.79 ± 0.68	49.26 ± 0.68	51.42 ± 0.68
QQP	47.94 ± 0.25	43.40 ± 0.25	49.12 ± 0.25	50.57 ± 0.25	48.22 ± 0.25	36.92 ± 0.24	41.70 ± 0.25
RTE	47.29 ± 3.01	46.93 ± 3.00	55.96 ± 2.99	46.57 ± 3.00	52.71 ± 3.01	52.35 ± 3.01	49.46 ± 3.01
SST-2	65.71 ± 1.61	52.98 ± 1.69	49.66 ± 1.69	49.08 ± 1.69	53.21 ± 1.69	54.59 ± 1.69	49.08 ± 1.69
WNLI	50.70 ± 5.98	49.30 ± 5.98	47.89 ± 5.97	56.34 ± 5.93	50.70 ± 5.98	50.70 ± 5.98	40.85 ± 5.88
Hellaswag	28.98 ± 0.45	28.75 ± 0.45	28.70 ± 0.45	27.57 ± 0.45	28.13 ± 0.45	27.57 ± 0.45	27.53 ± 0.45
Boolq	41.07 ± 0.86	50.89 ± 0.87	48.62 ± 0.87	44.59 ± 0.87	47.46 ± 0.87	55.38 ± 0.87	52.14 ± 0.87
CB	48.21 ± 6.74	50.00 ± 6.74	50.00 ± 6.74	50.00 ± 6.74	50.00 ± 6.74	37.50 ± 6.53	48.21 ± 6.74
Copa	65.00 ± 4.79	62.00 ± 4.88	64.00 ± 4.82	66.00 ± 4.76	63.00 ± 4.85	60.00 ± 4.92	60.00 ± 4.92
Average	46.50	45.60	46.38	49.24	48.53	49.47	44.31