BRAIN-INSPIRED SPARSE TRAINING ENABLES TRANSFORMERS AND LLMS TO PERFORM AS FULLY CONNECTED

Anonymous authors
Paper under double-blind review

ABSTRACT

This study aims to enlarge our current knowledge of the application of brain-inspired network science principles for training artificial neural networks (ANNs) with sparse connectivity. The Cannistraci-Hebb training (CHT) is a brain-inspired method for growing connectivity in dynamic sparse training (DST). 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: high time complexity of the link predictor and easy stack into the epitopological local minima. Here, 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 CHT 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. To further improve performance, we integrate CHTs with a sigmoid gradual density decay strategy, referred to as CHTss. Empirical results show that 1) using 5% of the connections, CHTss outperforms fully connected networks in two Transformer-based machine translation tasks; 2) using 30% of the connections, CHTss achieves superior performance compared to other dynamic sparse training methods in language modeling (LLaMA-130M) across different sparsity levels, and it surpasses the fully connected counterpart in zero-shot evaluations.

1 INTRODUCTION

Fully connected layers in large models pose computational challenges during training and deployment. In contrast, the brain's neural networks exhibit sparse connectivity Drachman (2005); Walsh (2013), suggesting more scalable architectures. Dynamic sparse training (DST) Mocanu et al. (2018); Jayakumar et al. (2020); Evci et al. (2020); Yuan et al. (2021); Zhang et al. (2024b) reduces computational and memory costs while maintaining performance. Unlike pruning methods Han et al. (2016); Frantar & Alistarh (2023); Zhang et al. (2024a), DST starts with sparse networks and evolves their topology during training. Key innovations of DST focus on regrowth criteria, such as the gradient-free Cannistraci-Hebb training (CHT) Zhang et al. (2024b), inspired by brain-inspired network science Cannistraci et al. (2013); Daminelli et al. (2015); Durán et al. (2017); Cannistraci (2018); Narula (2017). CHT excels in ultra-sparse ANNs but faces challenges such as stacking in epitopological local minima and high time complexity of link prediction, making it impractical for large-scale models.

This article introduces the Cannistraci-Hebb Training soft rule (CHTs), which addresses CHT's limitations. CHTs **1)** uses a multinomial distribution for both link removal and regrowth that balance the exploration and exploitation of network topology, **2)** reduces time complexity of the path-based link predictor to $\mathcal{O}(N^3)$ with a node-based solution, and **3)** leverages small-world properties for sparse initialization. Combined with a sigmoid density decay strategy, CHTss enables sparse neural networks to perform as fully connected on large-scale models.

From the experimental results, 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, CHTss outperforms the fully

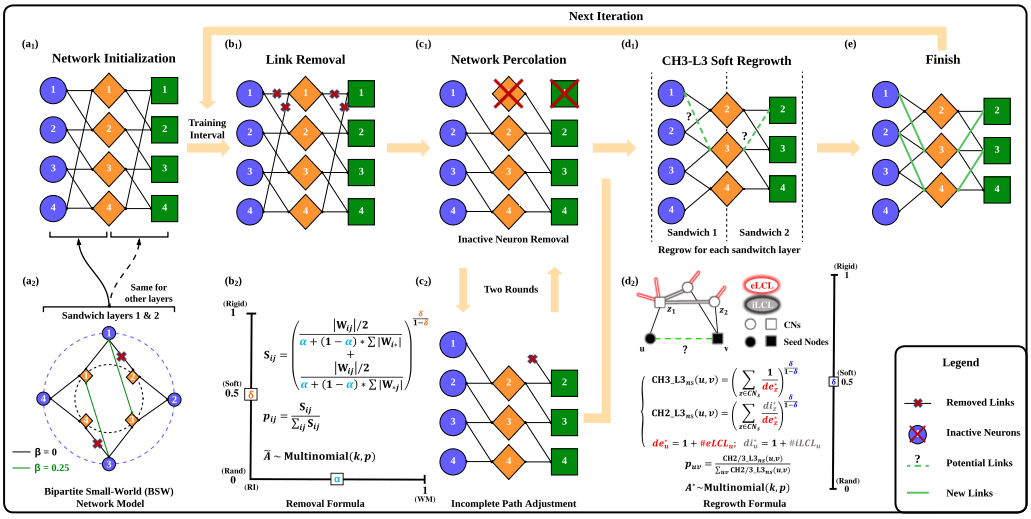

Figure 1: **Illustration of the CHTs process.** One training iteration follows the path of (a1) $->$ (b1) $->$ (c1) $->$ (c2) $->$ (d1) $->$ (e). (a1) Network initialization with each of the sandwich layers being a bipartite small-world (BSW) network. (a2) One sample BSW network with different β values (for $\beta = 0$, the network contains the black links; for $\beta = 0.25$, the network is formed by removing the marked black links and regrowing the green links). (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 CH2/3-L3 node-based soft rule. (d2) Detailed illustration of the CH2/3-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.

connected LLaMA-130M on zero-shot evaluation tasks on GLUE Wang et al. (2019) and SuperGLUE Wang et al. (2020) with only 30% density. These findings underscore the potential of CHTss in enabling highly efficient and effective large-scale sparse neural network training.

2 Cannistraci-Hebb training soft rule

> **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. See Appendix B for detailed descriptions.

Cannistraci-Hebb soft removal and regrowth. In this article, we adopt a probabilistic approach where the process of both regrowth and removal can be viewed as sampling from a $\{0, 1\}$ multinomial distribution, with the score assigned by either removal metrics or link prediction scores, introducing a "soft sampling" mechanism. In this setup, each mask value is not rigidly determined by the scores but allows for selecting (with lower probability) low-score links as the target links to remove or regrow, facilitating the escape from the epitopological local minima (ELM).

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) Zhang et al. (2024a), to remove the connections during dynamic sparse training. Detailed information can be found in Appendix C.

Node-based link regrowth. In the original CHT framework, the time complexity of the path-based CH3-L3 (**CH3-L3p**, see Appendix D) metric is $O(N \cdot d^3)$, 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

Table 1: Performance comparison on machine translation tasks of Multi30k, IWSLT, and WMT with varying sparsity levels. The scores indicate BLEU scores, which is the higher the better. CHTs (GMP) indicates CHTs uses GMP's density decay strategy. The best performance on each dataset is highlighted in bold and the performances better than the fully connected ones are marked with "*". s_i indicates the starting sparsity of the dst methods that use density decay strategy.

Method	Multi30k		IWSLT		WMT	
	95%	90%	95%	90%	95%	90%
FC	31.28		24.2		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
CHTs	28.12	30.35	20.55	21.60	21.14	22.68
$\text{MEST}_{EM\&S}$	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
CHTs (GMP) ($s_i = 0.5$)	30.49	30.33	23.68	23.64	**22.8**	23.22
CHTss ($s_i = 0.5$)	**32.04***	**32.79***	**24.86***	**24.57***	22.68	**24.05**

that eliminates the reliance on length-three paths between seed nodes, which also incorporates internal local community links (iLCL) to enhance the expressiveness of the formula. This variant, referred to as **CH2-L3n**, is formulated as:

$$\textbf{CH2-L3n}(u, v) = \sum_{z \in L3} \frac{di_z^*}{de_z^*} \tag{1}$$

Here, u and v denote the seed nodes, while z_1 and z_2 are common neighbors on the L3 path Muscoloni et al. (2022). The term de_z^* and di_z^* represents the number of external local community links (eLCL) and iLCL of node z, with a default increment of 1 to prevent division by zero. We evaluate the runtime performance of CH3-L3p and CH2-L3n across different network sizes and sparsity levels, as illustrated in Figure 2. The results reveal that the node-based version achieves significantly faster runtime performance compared to the path-based methods.

Sparse topological initialization with bipartite small-world model. In this work, we initialize the network of DST with the bipartite small-world model (BSW). The BSW model, with its small-world properties, ensures both high clustering and a short average path length. The high clustering increases the probability that seed nodes share common neighbors along L3 paths (paths of length three). This, in turn, improves the effectiveness of the CH-based link predictor in generating accurate predictions early in the training process. A detailed discussion of the sparse topological initialization can be found in Appendix E.

3 Sigmoid Gradual Decrease Density

As demonstrated in GraNet Liu et al. (2021) and $\text{MEST}_{EM\&S}$ Yuan et al. (2021), incorporating a density decrease strategy can significantly improve the performance of dynamic sparse training. In this article, we propose a sigmoid-based gradual density decrease strategy, defined as:

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

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. A detailed discussion of the decay strategy can be found in Appendix G.

Table 2: **Validation perplexity** of different dynamic sparse training (DST) methods on OpenWebText using LLaMA-130M across varying sparsity levels. 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			
	95%	90%	80%	70%
FC	19.27			
SET	28.37	24.73	22.02	20.82
RigL	49.39	37.18	66.35	25.85
CHTs	27.72	24.24	21.70	21.15
$\text{MEST}_{\text{EM\&S}}$	27.96	24.98	22.21	21.32
GMP ($s_i = 0.5$)	27.16	23.61	22.28	20.49
GraNet ($s_i = 0.5$)	61.31	26.81	29.03	22.84
CHTs (GMP) ($s_i = 0.5$)	26.81	22.94	20.94	20.01
CHTss ($s_i = 0.5$)	**25.29**	**22.71**	**20.78**	**19.92**

Table 3: **Zero-shot evaluation** of LLaMA-130M between fully connected pertrained and pretrained with CHTss (70% sparsity) across GLUE and Superglue datasets. MRPC and QQP use F1 scores while the others use ACC.

Dataset	FC	CHTss ($s_i = 0.5$)
CoLA	65.29 ± 1.47	**69.13 ± 1.43**
MNLI	32.44 ± 0.47	**32.72 ± 0.47**
MRPC	64.96 ± 2.36	**81.05 ± 1.64**
QNLI	**50.38 ± 0.68**	49.37 ± 0.68
QQP	52.09 ± 0.28	**53.82 ± 0.26**
RTE	48.38 ± 3.01	**50.54 ± 3.01**
SST-2	**49.54 ± 1.69**	49.08 ± 1.69
WNLI	49.30 ± 5.98	**52.11 ± 5.97**
Hellaswag	26.95 ± 0.44	**26.97 ± 0.44**
Boolq	43.85 ± 0.87	**56.79 ± 0.87**
CB	46.43 ± 6.72	**50.00 ± 6.74**
Copa	56.00 ± 4.99	**57.00 ± 4.98**
AVG	48.80	**52.38**
Win rate	0.17	**0.83**

4 EXPERIMENTS

Experimental details are provided in Appendix J. The baseline methods are detailed in Appendix I. We also demonstrate superiority with merely CHTs using MLP on image classification datasets (See Appendix K).

4.1 TRANSFORMER ON MACHINE TRANSLATION

We assess CHTs and CHTss using Transformer on classic machine translation tasks across three datasets. We report the BLEU in Table 1, which demonstrates that 1) CHTs surpasses other fixed density DST methods on all the sparsity scenrios. 2) Incorporating with the sigmoid density decrease, CHTss outperforms even the fully connected ones with only 5% density.

4.2 NATURAL LANGUAGE GENERATION

Language modeling. We utilize LLaMA-130M (Touvron et al., 2023a) architecture as the baseline for our language generation experiments. We show the validation perplexity results of each algorithm across the different sparsities in Table 2. As shown, CHTs stably outperforms SET and RigL while CHTss is constantly better than GraNet and GMP. At 70% sparsity, CHTss is already able to perform a comparable performance in comparison to the fully connected.

Zero-shot evaluations. The pretrained model of CHTss with 30% sparsity and the fully connected one are evaluated the zero-shot performance on the GLUE Wang et al. (2019) and SuperGLUE. We show the results in Table 3. The performance difference between FC and CHTss is statistically significant (p-value = 0.01) according to a paired two-sided Wilcoxon signed rank test, which means CHTss is significantly better than fully connected ones in zero-shot evaluations.

5 CONCLUSION

In this article, we propose 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 integrate CHTs with a sigmoid gradual density decay strategy. Empirically, CHTss surpasses the fully connected Transformer using only 5% density and achieves comparable language modeling performance, along with better zero-shot results, to the fully connected LLaMA-130M at just 30% density. This represents a relevant result for dynamic sparse training. We describe the limitations of this study and future works in Appendix L.

REFERENCES

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

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

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.

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. doi: 10.1038/s41598-018-33576-8.

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.

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

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

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. doi: 10.1088/1367-2630/17/11/113037. URL https://doi.org/10.1088/1367-2630/17/11/113037.

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*, pp. 4690–4721. PMLR, 2022.

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

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. doi: 10.1093/bib/bbx041. URL https://doi.org/10.1093/bib/bbx041.

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

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

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*, pp. 2943–2952. PMLR, 2020. URL http://proceedings.mlr.press/v119/evci20a.html.

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

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

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.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1510.00149.

Donald Hebb. The organization of behavior. emphnew york, 1949.

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.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://doi.org/10.1109/5.726791.

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. URL https://openreview.net/forum?id=B1VZqjAcYX.

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.

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.

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.

Alessandro Muscoloni, Umberto Michieli, Yingtao Zhang, and Carlo Vittorio Cannistraci. Adaptive network automata modelling of complex networks. *Preprints*, May 2022. doi: 10.20944/preprints202012.0808.v3. URL https://doi.org/10.20944/preprints202012.0808.v3.

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. doi: 10.1007/s41109-017-0048-x.

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)*, pp. 20–35, 2018.

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*, pp. 4518–4523, 2023.

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.

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*, pp. 2134–2146. PMLR, 2023.

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*, 2023a.

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*, 2023b.

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

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. URL https://arxiv.org/abs/1804.07461.

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. URL https://arxiv.org/abs/1905.00537.

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

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

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

Yingtao Zhang, 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*, 2024a. URL https://openreview.net/forum?id=Tr0lPx9woF.

Yingtao Zhang, Jialin Zhao, Wenjing Wu, Alessandro Muscoloni, and Carlo Vittorio Cannistraci. Epitopological learning and cannistraci-hebb network shape intelligence brain-inspired theory for ultra-sparse advantage in deep learning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=iayEcORsGd.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017. URL https://arxiv.org/abs/1710.01878.

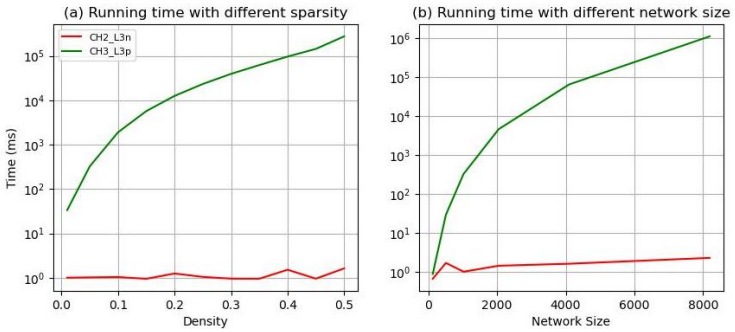

Figure 2: **Runtime Performance Evaluation** of node-based and path-based methods across varying densities and network sizes. In (a), the network size is fixed at 1024 × 1024, while in (b), the density is fixed at 5%.

A Related Work

A.1 Dynamic sparse training

Dynamic sparse training is a subset of sparse training methodologies. Unlike the static sparse training (also known as pruning at initialization) methods Prabhu et al. (2018); Lee et al. (2019); Dao et al. (2022); Stewart et al. (2023), dynamic sparse training allows for the evolution of network topology during the training process. The pioneering method in this field was Sparse Evolutionary Training (SET) Mocanu et al. (2018), 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 Bellec et al. (2017), a method that adjusts network connections based on stochastic gradient updates combined with a Bayesian-inspired update rule. Another significant contribution is the RigL Evci et al. (2020), which leverages the gradient information of non-existing links to guide the regrowth of new connections during training. MEST Yuan et al. (2021) utilizes both gradient and weight magnitude information to selectively remove and randomly regrow new links, which is the same as SET. In addition, it introduces an EM&S strategy that allows the model training with a larger density and finally convergence to the desired density. The Top-KAST Jayakumar et al. (2020) 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 Lasby et al. (2023) 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. Zhang et al. (2024b) 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.

A.2 Cannistraci-Hebb Theory and Network Shape Intelligence

As the SOTA gradient-free link regrown method, CHT Zhang et al. (2024b) originates from a brain-inspired network science theory. Drawn from neurobiology, Hebbian learning was introduced in 1949 (Hebb, 1949) 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 (Cannistraci et al., 2013; Daminelli et al., 2015; Durán et al., 2017; Cannistraci, 2018; Narula, 2017). The latter is also called *epitopological plasticity* (Cannistraci et al., 2013) because plasticity means "to change shape," and *epitopological* means "via a new topology." *Epitopological Learning* (EL) (Daminelli et al., 2015; Durán et al., 2017; Cannistraci, 2018) 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

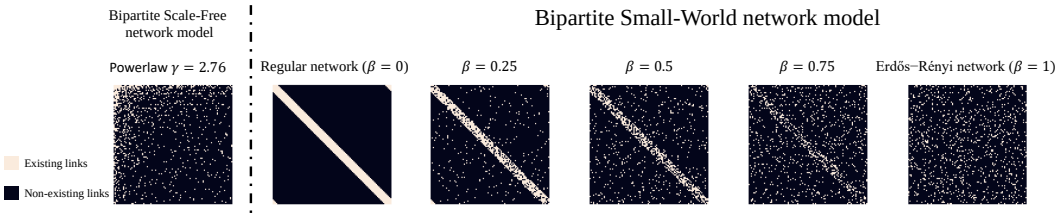

Figure 3: **The adjacency matrices** of the Bipartite Scale-Free (BSF) network model and the Bipartite Small-World (BSW) network model vary with different values of β. a) The BSF model inherently forms a scale-free network characterized by a power-law distribution with $\gamma = 2.76$. b) As β changes 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.

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 and most robust performing network automata which is inside Cannistraci-Hebb (CH) theory (Muscoloni et al., 2022) that can automatically evolve the network topology with the 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 (Muscoloni et al., 2022). This minimization of the external links induces a topological isolation of the local community, which is equivalent to forming a barrier around the local community. 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.

B EPITOPOLOGICAL LOCAL MINIMA

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 Zhang et al. (2024b) 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 implements 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.

C SOFT LINK REMOVAL ALTERNATING FROM RI AND WEIGHT MAGNITUDE

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) Zhang et al. (2024a), to remove the connections during dynamic sparse training.

$$\mathbf{RI}_{ij} = \frac{|\mathbf{W}_{ij}|}{\sum |\mathbf{W}_{*j}|} + \frac{|\mathbf{W}_{ij}|}{\sum |\mathbf{W}_{i*}|} \tag{3}$$

As illustrated in Equation 3, 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 Frantar & Alistarh (2023) and wanda Sun et al. (2023). 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 compared to 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 4.

$$\mathbf{S}_{ij} = \left(\frac{|\mathbf{W}_{ij}|/2}{\alpha + (1-\alpha)\sum |\mathbf{W}_{i*}|} + \frac{|\mathbf{W}_{ij}|/2}{\alpha + (1-\alpha)\sum |\mathbf{W}_{*j}|} \right)^{\frac{\delta}{1-\delta}} \tag{4}$$

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.

D PATH-BASED LINK PREDICTOR CH3-L3

One significant challenge for CHT lies in the time complexity of link prediction. In the original CHT framework, the CH3-L3 metric is employed for link regrowth, defined as follows:

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

Here, u and v denote the seed nodes, while z_1 and z_2 are common neighbors on the L3 path Muscoloni et al. (2022). The term de^*_i represents the number of external local community links (eLCL) of node i, with a default increment of 1 to prevent division by zero. Path-based link prediction has demonstrated its effectiveness on both real-world networks Muscoloni et al. (2022) and artificial neural networks Zhang et al. (2024b). 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.

E SPARSE TOPOLOGICAL INITIALIZATION

Correlated sparse topological initialization. Correlated Sparse Topological Initialization (CSTI) is a physical-informed topological initialization. CSTI generates the adjacency matrix by computing 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, with the goal of improving 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) Zhang et al. (2024b) 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 Barabási & Albert (1999), a well-established method for creating scale-free networks in which the degree distribution follows a power law (γ=2.76 in Figure 3). 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 Zhang et al. (2024b) is designed to incorporate small-world properties and 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 clustering but lacking the small-world property. Similar to the Watts-Strogatz model (WS) Watts & Strogatz (1998), 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** ERDdS & R&wi (1959), exhibiting small-world properties but without high clustering coefficient, which is popular as the topological initialization of the other dynamic sparse training methods Mocanu et al. (2018); Evci et al. (2020); Yuan et al. (2021).

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

As indicated in SET and CHT Mocanu et al. (2018); Zhang et al. (2024b), 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:

- 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) Zhang et al. (2024b) in the model. CAP is defined as the chain of the 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.
- 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 4, 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.

G Density Decay Strategies

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) Zhu & Gupta (2017), which adheres

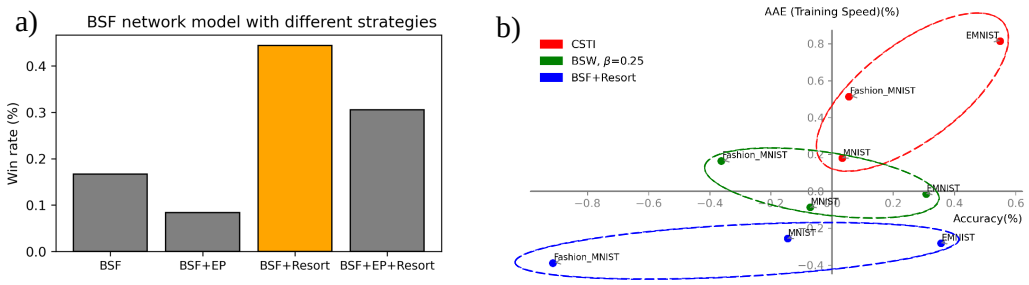

Figure 4: **The Performance** of the bipartite scale-free model and two enhanced techniques. a) shows the win rate of the Bipartite Scale-Free network model (BSF) with the different techinques. *EP* stands for equal partition of the first layer, and *Resort* refers to reordering the neurons based on 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 $\beta = 0.25$.

to a cubic function:

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3, \tag{6}$$

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.

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 Equation 2 in the main text. 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.

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 The integral of the GraNet decrease function from t_0 to t_f is:

$$\begin{aligned} &\int_{t_0}^{t_f} (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3 dt \\ &= \frac{1}{4}(s_i - s_f)(t_f - t_0). \end{aligned} \tag{7}$$

For the sigmoid decrease, the integral is:

$$\begin{aligned} &\int_{t_0}^{t'_f} (s'_i - s_f)\left(\frac{1}{1 + e^{-k\left(t - \frac{t'_f + t_0}{2}\right)}}\right) dt \\ &= \frac{(s'_i - s_f)(t'_f - t_0)}{2}. \end{aligned} \tag{8}$$

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.

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 Zhang et al. (2024a), which has shown that RI provides a significant performance advantage over weight magnitude, particularly when pruning models initialized with high density.

H NETWORK PERCOLATION AND EXTENSION TO TRANSFORMER.

We have adapted network percolation Li et al. (2021); Zhang et al. (2024b) to suit the architecture of the Transformer after link removal. The underlying concept involves identifying inactive neurons, which we define as those lacking connections on one or both sides of a neuron layer. 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 Touvron et al. (2023a;b).

I BASELINE METHODS

I.1 FIXED DENSITY DYNAMIC SPARSE TRAINING METHODS

SET Mocanu et al. (2018): Removes connections based on weight magnitude and randomly regrows new links.

RigL Evci et al. (2020): Removes connections based on weight magnitude and regrows links using gradient information, gradually reducing the proportion of updated connections over time.

CHT Zhang et al. (2024b): A state-of-the-art (SOTA) gradient-free method that removes links with weight magnitude and regrows links based on CH3-L3 scores. (Note: CHT is only evaluated on MLPs due to its computational cost in large models.)

I.2 GRADUAL DENSITY DECREASE DYNAMIC SPARSE TRAINING METHODS

GMP Han et al. (2015); Zhu & Gupta (2017): Prunes the network with weight magnitude and gradually decrease the density based on Equation 7. Although originally a pruning method, GMP is treated as a dynamic sparse training method in their implementation Zhu & Gupta (2017), as it stores historical weights and allows pruned weights to reappear during training since, during training, the pruning threshold might change.

MEST$_{EM\&S}$ Yuan et al. (2021): 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 Liu et al. (2021): Gradually decreases density using Equation 7. Similar to RigL, GraNet removes links based on the weight magnitude and regrows new links with the gradient of the existing links.

J EXPERIMENTAL SETUP

We evaluate the performance of CHTs using MLPs for image classification tasks on the MNIST LeCun et al. (1998), Fashion MNIST Xiao et al. (2017), and EMNIST Cohen et al. (2017) 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 Elliott et al. (2016), IWSLT14 en-de Cettolo et al. (2014), and WMT17 en-de Bojar et al. (2017) datasets. Additionally, we conduct language modeling

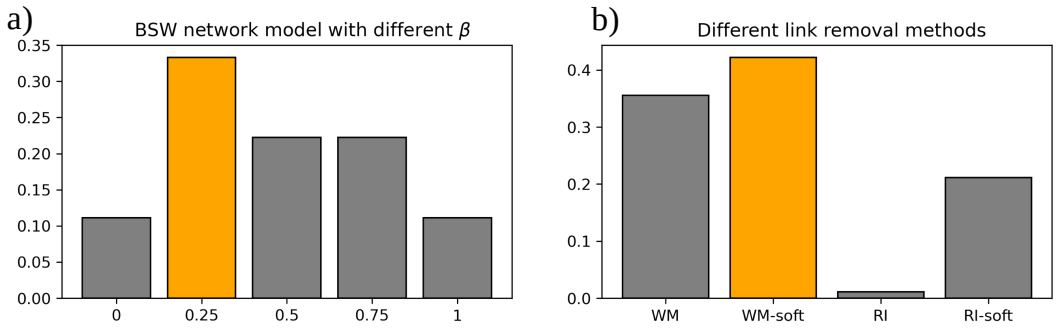

Figure 5: **The ablation test** of the β of the bipartite small world model and the removal methods in CHTs. a) evaluates the influence of the rewiring rate β on the model performance when initialized with the Bipartite Small-World network model (BSW). b) assess the influence of link removal selecting 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.

experiments using the OpenWebText dataset Gokaslan & Cohen (2019) and evaluate zero-shot performance on the GLUE Wang et al. (2019) and SuperGLUE Wang et al. (2020) benchmark with LLaMA-130M Touvron et al. (2023a). 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-130M, 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.

K MLP FOR IMAGE CLASSIFICATION

Ablation Test. Using MLP, we conduct an ablation study on each component proposed within the CHTs framework to determine the most effective implementation to apply next for the Transformer model. Figure 5a) compares the topologies initialized with the Bipartite Small-World (BSW) model at different values of β, clearly indicating that $\beta = 0.25$ yields the best results. Figures 5b) assess the link removal methods, concluding that the weight magnitude soft (WM-soft) method outperforms all others. We consider the best settings showcased in these results to decide the CHTs strategy for training Transformers and LLaMA-130M.

Table 4: Performance comparison of different fixed sparsity 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 that indicates the final size of the network. The best method for each dataset is highlighted in bold and the performances better than the fully connected ones 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%

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 CHTs is CSTI since the input bipartite layer can directly receive information from the input pixels. Table 4 displays the performance of DST methods compared to their fully connected counterparts across three basic datasets. The DST methods are tested at 99% sparsity. As shown in Table 4, 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. In addition, we present the active neuron post-percolation rate (ANP) for each method in Table 4. It is evident that CHTs adaptively percolates the network more effectively while retaining performance.

L LIMITATION

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 Thangarasa et al. (2023) 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 CHTss in larger LLMs such as LLaMA-1b and LLaMA-7b to evaluate the performance in scenarios with denser layers.

Table 5: **Hyperparameters of MLP on Image Classification Tasks.**

Hyper-parameter	**MLP**
Hidden Dimension	1568
# Hidden layers	3
Batch Size	32
Training Epochs	100
LR Decay Method	Linear
Learning Rate	0.025
ζ (fraction of removal)	0.3
Update Interval (for DST)	1

Table 6: **Hyperparameters of Transformer on Machine Translation Tasks.**

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
Decay Method	inoam	inoam	inoam
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	100	100

Table 7: **Hyperparameters of LLaMA-130M on OpenWebText.**

Hyper-parameter	**LLaMA-130M**
Embedding Dimension	768
Feed-forward Dimension	2048
Global Batch Size	512
Sequence Length	256
Training Steps	30000
Learning Rate	3e-3
Warmup Steps	10000
Optimizer	Adam
Layer Number	12
Head Number	12
ζ (fraction of removal)	0.1
Update Interval (for DST)	100