UNDERSTANDING DIVERSITY BASED NEURAL NETWORK PRUNING IN TEACHER STUDENT SETUP

Rupam Acharyya
Department of Mathematics
University at Buffalo
rupamach@buffalo.edu

Ankani Chattoraj *
Department of Brain & Cognitive Science
University of Rochester
achattor@ur.rochester.edu

Boyu Zhang *
Department of Computer Science
University of Rochester
bzhang25@u.rochester.edu

Shouman Das
Department of Mathematics
University of Rochester
sdas13@ur.rochester.edu

Daniel Štefankovič
Department of Computer Science
University of Rochester
stefanko@cs.rochester.edu

ABSTRACT

Despite multitude of empirical advances, there is a lack of theoretical understanding of the effectiveness of different pruning methods. We inspect different pruning techniques under the statistical mechanics formulation of a teacher-student framework and derive their generalization error (GE) bounds. In the first part, we theoretically prove empirical observations of a recent work that showed *Determinantal Point Process* (DPP) based *node* pruning method is notably superior to competing approaches when tested on real datasets. In the second part, we use our theoretical setup to prove that the baseline *random edge pruning* method performs better than the *DPP node pruning* method, consistent with the finding in literature that sparse neural networks (*edge pruned*) generalize better than dense neural networks (*node pruned*) for a fixed number of parameters.

1 INTRODUCTION

Deep neural networks have achieved impressive results in a wide variety of applications such as classification Krizhevsky et al. (2012); Liu et al. (2017), image processing Litjens et al. (2017); Badrinarayanan et al. (2017), natural language processing Devlin et al. (2018); Deng & Liu (2018); Socher et al. (2013), etc. Most of these networks use millions of parameters which makes inference computationally expensive and memory intensive Devlin et al. (2018). To address this, researchers explore pruning techniques with the primary goal of comparing performance on real datasets. The broad scientific paradigm explored by most pruning techniques is to empirically and heuristically determine either how to prune a network or what to prune in a network (sometimes both). In this work, we take a step towards theoretical understanding of these two prime aspects of pruning methods.

We compare the quality of different pruning methods for feedforward neural networks under the *teacher-student* framework Saad & Solla (1995a;b; 1997); Goldt et al. (2019) in the thermodynamic limit (input dimension goes to infinity) using *generalization error bounds* (GE), a theoretical measure of performance of machine learning models on unseen test data Vapnik (1999). A fairly recent work by Mariet & Sra (2016) empirically investigates a Determinantal Point Process (DPP) based node pruning technique Macchi (1975); Kulesza et al. (2012). In the first part, we provide theoretical guarantees for their empirical observations thereby taking a step towards theoretical understanding of the question: how to prune? A very recent review Blalock et al. (2020) discusses

*equal contribution

Table 1: Different pruning methods and notations for their GE. Here f denotes the pruned student network. u.a.r. and w.p. stand for *uniformly at random* and *with probability* respectively.

Pruning Method	**Procedure**	**Retained Parameters**	**GE without reweighting**	**GE with reweighting**
Random Node	Keep k_n nodes u.a.r.	k_n hidden nodes	$\epsilon_{k_n}^{Rand\ Node}(f)$	$\hat{\epsilon}_{k_n}^{Rand\ Node}(f)$
Importance Node	He et al. (2014)	k_n hidden nodes	$\epsilon_{k_n}^{Imp\ Node}(f)$	$\hat{\epsilon}_{k_n}^{Imp\ Node}(f)$
DPP Node	see Section A	k_n hidden nodes	$\epsilon_{k_n}^{DPP\ Node}(f)$	$\hat{\epsilon}_{k_n}^{DPP\ Node}(f)$
Random Edge	Keep an edge w.p. c for each hidden node	k_e incoming edges per hidden node	$\epsilon_{k_e}^{Rand\ Edge}(f)$	$\hat{\epsilon}_{k_e}^{Rand\ Edge}(f)$

empirical results across several papers (81 research articles) to conclude that sparse models obtained after edge/connection (used interchangeably) pruning outperforms dense ones obtained after node pruning for a fixed number of parameters. In the second part, we extend our theoretical setup and compare node and edge pruning techniques thereby addressing the question: what to prune?

2 Online Learning in Teacher-Student Setup Goldt et al. (2019):

We use a two-layer perceptron which has N input units, M hidden units and 1 output unit as the *teacher network* to generate labels for i.i.d Gaussian input, $\boldsymbol{x}^t = (x_1^t, \ldots, x_N^t)$ where $x_i^t \in \mathcal{N}(0,1)$ $\forall i \in \{1, \ldots, N\}$. Let $\theta^* = \{\boldsymbol{w}^* (\in \mathbb{R}^{M \times N}), v^* \in \mathbb{R}^M\}$ denote the fixed parameters of the teacher network. The label y^t of the input $\boldsymbol{x}^t$ $(t = 1, 2, \ldots)$ is given as, $y^t = \sum_{m=1}^{M} v_m^* g\left(\frac{w_m^* \boldsymbol{x}^t}{\sqrt{N}}\right) + \sigma\zeta^t$, where $\zeta^t \sim \mathcal{N}(0,1)$ is the output noise, and g is the sigmoid activation function. The input and teacher generated labels ($\{(\boldsymbol{x}^1, y^1), \ldots\}$) are used to train a two-layer *student network* with N input units, K hidden units ($K \geq M$) and 1 output unit using online SGD learning method. We consider the quadratic training loss, i.e., $L(\theta) = \frac{1}{2}\left[\sum_{k=1}^{K} v_k g\left(\frac{w_k \boldsymbol{x}^t}{\sqrt{N}}\right) - y^t\right]^2$, where $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$ denotes the parameter of the student network. Goldt et al. (2019) showed that GE $\epsilon(f)$ (expected error on the unseen data, for details see S31 of Goldt et al. (2019)) for the overparameterized student network is a function of the *order parameters*, which are $Q_{ik} = \frac{w_i^T w_k}{N}$, $R_{in} = \frac{w_i^T w_n^*}{N}$, $R_{mn} = \frac{w_m^{*T} w_n^*}{N}$. Intuitively, these order parameters measure the similarities between and within the hidden nodes of teacher and student networks.

All the assumptions of this setup are stated in the Appendix B.

3 Neural Network Pruning in Teacher Student Setup

In this work we consider three node pruning methods and one baseline edge pruning method (DPP, random and importance node pruning and random edge pruning, see Table 1 for notations). We first prune the overparameterized student network using these various pruning methods. We then compare the performance of the pruned student networks by analyzing their GE bounds following Goldt et al. (2019) (concept explained in Figure 1). For node and edge pruning comparison, we choose the parameters k_n and k_e (see Table 1) such that the total number of parameters of the networks remain same, i.e.,

$$\frac{k_n}{K} = \lim_{N \to \infty} \frac{k_e}{N} = c, \tag{1}$$

where $c \in [0, 1]$ is a constant.

3.1 Comparing Node Pruning Methods

Following Goldt et al. (2019) we observe that the fully trained student network achieves 0 GE when there is no added output noise in the teacher network (see Sec. 2). However, the GE of the pruned network increases due to underparameterization. We theoretically show that this increment in GE due to DPP node pruning is less than that for random and importance node pruning methods. This justifies the empirical findings of Mariet & Sra (2016). Our result is formalized in Threorem 1.

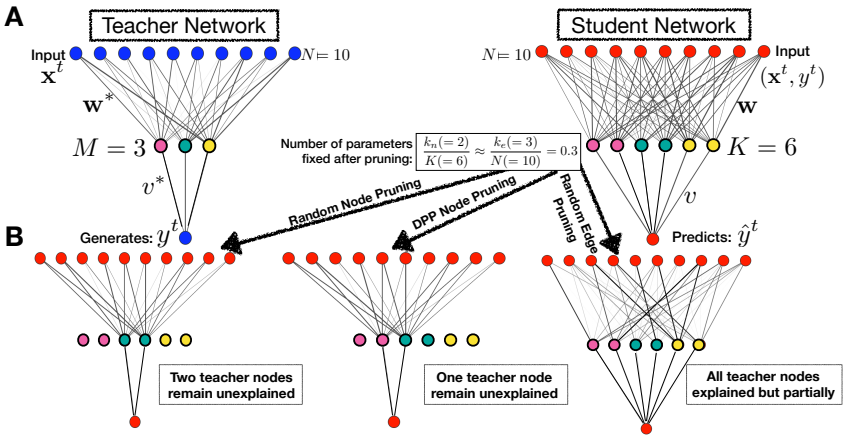

Figure 1: **(A)** Two layer teacher-student framework: A teacher neural network with 3 hidden nodes (left) and a student network with 6 hidden nodes (right). Input data (i.i.d) along with its label generated by teacher network are fed to student network to predict. **(B)** Intuitive example for 3 types of pruning on student network. For $k_n = 2$, random node pruning might only be able to explain 1 teacher hidden node, whereas DPP node pruning will always retain (partial) information about 2 teacher hidden nodes, hence preforms better. Random edge pruning retains sparse information about all 3 teacher nodes which is enough to outperform DPP node pruning. All notations follow Table 1

Theorem 1. *Assume* $(A1) - (A7)$ *(see Appendix B). Then for* $k_n \leq M$ *we have,*

$$\mathbb{E}_f\left[\epsilon_{k_n}^{Rand\ Node}(f)\right] \geq \epsilon_{k_n}^{DPP\ Node}(f) \ \text{ and } \ \mathbb{E}_f\left[\hat{\epsilon}_{k_n}^{Rand\ Node}(f)\right] \geq \hat{\epsilon}_{k_n}^{DPP\ Node}(f) \tag{2}$$

and,

$$\epsilon_{k_n}^{Imp\ Node}(f) \geq \hat{\epsilon}_{k_n}^{DPP\ Node}(f), \tag{3}$$

i.e., DPP node pruning outperforms random node pruning in the above setup. Here the expectation is taken over the the subsets of hidden nodes of size k_n *chosen u.a.r (see Table 1 for the notations).*

Proof Idea of Theorem 1: From Goldt et al. (2019) we notice that in the overparameterized setting (i.e., $K > M$), multiple student hidden nodes learn a single teacher hidden node. This results in an equivalence relation over the set of student hidden nodes. As a consequence, the order parameters denoting the correlation among the student and teacher hidden nodes have block structure where each block corresponds to the set of student hidden nodes which learn the same teacher hidden node (see Figure 3 in Appendix). We first observe that the expected kernel of the DPP node pruning is same as the order parameter Q and hence block diagonal. Then, using property of DPP, we prove the DPP pruning method will retain a subset of student hidden nodes with at most 1 hidden node from each block when $k_n \leq M$. However, in random and importance node pruning, two student nodes from the same block may survive after pruning with non-zero probability. Hence, more teacher nodes may remain unexplained by the student network after random or importance node pruning, resulting in increased GE (details in Appendix C).

3.2 Comparing Node and Edge Pruning Methods

In random edge pruning method, for each student hidden node, an incoming edge is kept with probability $c = \lim_{N\to\infty} \frac{k_e}{N}$. Majority of empirical studies throughout literature use random edge or node pruning as a baseline for empirical comparison (see papers in Blalock et al. (2020)) making it an obvious candidate for our theoretical comparisons as well. It has been shown empirically by Mariet & Sra (2016) and theoretically by us that DPP node pruning is an above baseline node pruning method. In this section we show that baseline random edge pruning outperforms DPP node pruning which is consistent with the empirical observations that sparse models outperform dense models (section 3.2 of Blalock et al. (2020)). Specifically, here we show that GE after random edge pruning is less than GE after DPP node pruning which is formalized below.

Theorem 2. *Assume* $(A1) - (A7)$ *(see Appendix B). Let* k_n *and* c *satisfy equation 1, and* $0 \leq c \leq \frac{1}{Z}$ *and* $Z(= \frac{K}{M}) \geq 4$. *Then*

$$\epsilon_{k_n}^{DPP\ Node}(f) \geq \epsilon_c^{Rand\ Edge}\left(\mathbb{E}\left[f\right]\right), \tag{4}$$

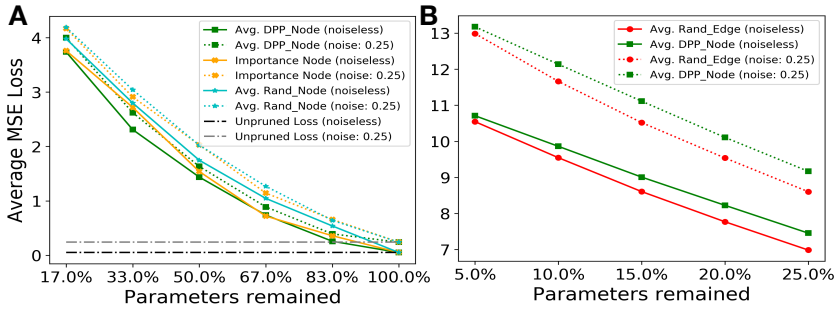

Figure 2: **(A)** Simulation results comparing node pruning methods in teacher student setup with $M = 2$ and $K = 6$. DPP Node pruning performs better than importance and random node pruning which is consitent with Theorem 1 and empirical results from Mariet & Sra (2016). **(B)** Baseline random edge pruning beats DPP node pruning (Theorem 2) with $M = 5$ and $K = 20$.

i.e., Random edge pruning outperforms DPP node pruning in the above setup.

Proof Idea of Theorem 2: When $k_n \leq M$, node pruned student network leaves at least $(M - k_n)$ teacher nodes unexplained, whereas after random edge pruning, student network can retain at least partial information about every teacher node (see Figure 1 (B)). After a pruning routine, the sum of partial information about all teacher nodes in an edge pruned student network dominates the sum of information for the explained subset of teacher nodes in a node pruned student network.

4 Simulations

We run the DPP node, random edge/node, and importance node pruning simulations under the teacher-student setup. For all the simulations, we sampled the 800000 i.i.d input samples from $\mathcal{N}(0, 1)$ as training data and 80000 as testing data. Following notations from Section 2, we set $M = 2$, $K = 6$, $N = 500$, and $v^* = 4$. The first layer teacher network weights $\boldsymbol{w}^*$ and all the student network parameters $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$ were drawn independently from $\mathcal{N}(0, 1)$ as initialization. We choose learning rate $\eta = 0.50$, and it is scaled to $\frac{\eta}{\sqrt{N}}$ for $\boldsymbol{w}$ and $\frac{\eta}{N}$ for $\boldsymbol{v}$. We run the simulations for both noiseless ($\sigma = 0$ in Sec. 2) and noisy ($\sigma = 0.25$) output labels. For comparisons between node and edge pruning, we use the node-to-edge ratio $[1 : 83, 2 : 166, 3 : 250, 4 : 333, 5 : 417, 6 : 500]$ to keep the number of parameters the same, given $N = 500$, $K = 6$, and $M = 2$. In addition, we run the same simulation with $K = 5$ and $M = 20$ to compare random Edge Pruning with DPP node pruning. We observe that 1) DPP node pruning outperforms random and importance node in both noisy and noiseless case (see Figure 2A), which confirms Theorem 1. 2) Random edge pruning is better than DPP node pruning for $c \leq \frac{1}{Z}(= 0.25)$ with $Z = 4$ and $M = 5$ in both noisy and noiseless cases (see Figure 2B), validating Theorem 2.

5 Discussion and Future Work

All our theoretical results have been proved on single hidden layer neural networks which gives future scope of extending them to multiple hidden layer networks. Throughout this work, we focus only on pruning methods in which a feedforward pre-trained neural network is pruned once without retraining. We choose this class because it is feasible to make theoretical comparisons with closed form solutions of GE. The various existing pruning methods can be broadly subsumed into a couple of categories Blalock et al. (2020), mainly governed by the principles of pruning heuristics. First category is the magnitude-based approaches which are not only good and common baselines in the literature but they also give comparable performance to other methods such as the gradient-based methods Lee et al. (2019); Yu et al. (2018); Blalock et al. (2020). Another category is the random pruning which serves as an useful baseline for showing superior performance of any other pruning technique. We hence show all our theoretical results w.r.t these two categories. We do not focus on any specific algorithm within these categories but explore the general concept for theoretical results. Any specific algorithm based theoretical understanding can also be an extension of this work.

REFERENCES

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, pp. 6979–6989, 2019.

Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. Reshaping deep neural network for fast decoding by node-pruning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 245–249. IEEE, 2014.

Kumar Joag-Dev, Frank Proschan, et al. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. *arXiv preprint arXiv:1906.06307*, 2019.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42: 60–88, 2017.

Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. In *International Conference on Learning Representations*, 2016.

David Saad and Sara A Solla. Exact solution for on-line learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337, 1995a.

David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52 (4):4225, 1995b.

David Saad and Sara A Solla. Learning with noise and regularizers in multilayer neural networks. In *Advances in Neural Information Processing Systems*, pp. 260–266, 1997.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9194–9203, 2018.

A Determinantal Point Process (DPP)

DPP Macchi (1975) is a probability distribution over power set of a ground set $\mathcal{G}$, here finite. DPP is a special case of negatively associated distributions Joag-Dev et al. (1983) which assigns higher probability mass on diverse subsets. Formally, a DPP with a marginal kernel L ($\in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{G}|}$) is: $\mathbb{P}[\mathbf{Y} = Y] = \frac{\det(L_Y)}{\det(L+I)}$, where $Y \subseteq \mathcal{G}$ and L_Y is the principal submatrix defined by the indices of Y. We use k-DPP to denote the probability distribution over subsets of fixed size k.

Remark: DIVNET denotes DPP node pruning with reweighting as in Mariet & Sra (2016).

B Assumptions

Our theoretical results assume Goldt et al. (2019):

- (A1) If $\boldsymbol{x} = (x_1, \ldots, x_N)$ is an input then $x_i \in \mathcal{N}(0, 1)$. Also, $N \to \infty$.
- (A2) Both the teacher and the student networks have only one hidden layer.
- (A3) $K \geq M$ and $K = Z \cdot M$ where $Z \in \mathbb{Z}^+$.
- (A4) The activation in the hidden layer is sigmoidal for both teacher and student network.
- (A5) The output $\in \mathbb{R}$ (i.e., regression problem).
- (A6) The order parameters (see section 2) satisfy the ansatz as in (S58) - (S60) of Goldt et al. (2019).
- (A7) No noise is added to the labels generated by the teacher network, i.e., $\sigma = 0$ in Section A.

With the above assumptions, authors of Goldt et al. (2019) gave a closed form of the GE as follows:

$$\epsilon_g = f_1(Q) + f_2(T) - f_3(R, Q, T) \tag{5}$$

where,

$$f_1(Q) = \frac{1}{\pi} \sum_{i,k} v_i v_k \arcsin \frac{Q_{ik}}{\sqrt{1 + Q_{ii}}\sqrt{1 + Q_{kk}}} \tag{6}$$

$$f_2(T) = \frac{1}{\pi} \sum_{n,m} v_n^* v_m^* \arcsin \frac{T_{nm}}{\sqrt{1 + T_{nn}}\sqrt{1 + T_{mm}}} \tag{7}$$

$$f_3(R, Q, T) = \frac{2}{\pi} \sum_{i,n} v_i v_n^* \arcsin \frac{R_{in}}{\sqrt{1 + Q_{ii}}\sqrt{1 + T_{nn}}} \tag{8}$$

where Q, R, T are the order parameters as defined in main text. We also have the assumption equation 1 about the relation between number of edges and nodes kept after pruning.

C Properties of DPP Kernel

In main text we see that each node in the hidden layer of a student network carries certain amount of information about the training data and it is captured in a vector form. We create an information matrix by accumulating the information vectors of these hidden nodes. For simplicity of theoretical analysis, we have considered the kernel as the inner product of the information matrix. In the thermodynamic limit, the inner product is divided by the input dimension. Formally, if $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are the information at i^{th} hidden node and j^{th} hidden node respectively, then

$$L_{ij} = \frac{1}{N}\frac{1}{n}\boldsymbol{h}_i^T \boldsymbol{h}_j$$

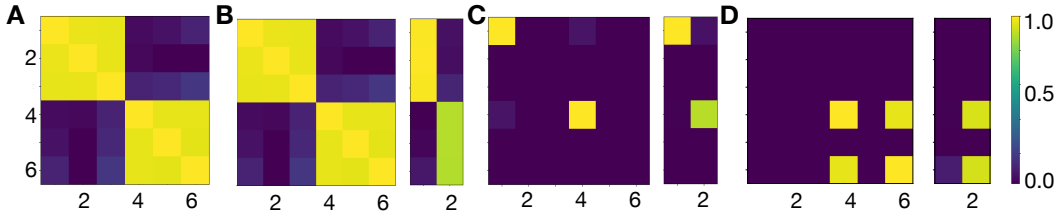

Figure 3: Order parameters and the kernel of DPP node pruning in teacher student setup with $M = 2$ and $K = 6$. **(A)** Kernel of the DPP mode pruning. **(B)** Order parameters Q and R. Q is same as the kernel in (A). **(C)** Order parameters after DPP node pruning with $k_n = 2$. DPP keeps exactly one node from each block. **(D)** Order parameters after random node pruning with $k_n = 2$. Two nodes from same block may survive.

where n is the total number of training examples. It can be seen that the analysis for the kernel defined in main text is similar. Note that all analyses are for the student network trying learn from the teacher network. Refer to main text for details of notations.

Lemma 1. *Assume (A1) - (A7). Then the expected kernel of DPP Node for the hidden layer is the order parameter Q.*

Proof of Lemma 1. For the two-layer teacher-student setup, the hidden layer gets information $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K)$ from the input layer, where $\boldsymbol{h}_i = (h_{i1}, \ldots, h_{in})$ and $h_{ij} (= t_j^T \boldsymbol{w}_i)$ is the information at i^{th} hidden node on j^{th} input data (t_j). Hence,

$$\boldsymbol{h}_i^T \boldsymbol{h}_j = \sum_{k=1}^{n} h_{ik} h_{jk} = \sum_{k=1}^{n} t_k^T \boldsymbol{w}_i \cdot t_k^T \boldsymbol{w}_j = \sum_{k=1}^{n} \boldsymbol{w}_i^T t_k \cdot t_k^T \boldsymbol{w}_j = \sum_{k=1}^{n} \boldsymbol{w}_i^T (t_k t_k^T) \boldsymbol{w}_j$$

But for the given input distribution (i.i.d. Gaussian), $\mathbb{E}[t_k t_k^T] = \boldsymbol{I}_{N \times N}$. Hence, $\lim_{N \to \infty} \mathbb{E}[L_{ij}] = \lim_{N \to \infty} \mathbb{E}[\frac{1}{N} \frac{1}{n} \boldsymbol{h}_i^T \boldsymbol{h}_j] = \lim_{N \to \infty} \frac{1}{N} \boldsymbol{w}_i^T \boldsymbol{w}_j = Q_{ij}$, and we have the lemma. □

From Goldt et al. (2019) we know that Q is a block diagonal matrix where each "block" (or "group" used interchangeably henceforth) refers to the set of student hidden nodes that represent (explian/learn) one particular teacher hidden node.

D Proof of the Theorems

D.1 Required Lemmas

We use the following lemmas to prove the main theorems.

Lemma 2. *Assume $(A1) - (A7)$. Let $k_n \leq M$ nodes are selected by the DPP Node pruning method,*

$$\epsilon_{k_n}^{DPPNode}(f) = (v^*)^2 \left[\frac{k_n}{6} \left(1 - \frac{1}{Z} \right)^2 + \frac{M - k_n}{6} \right] \tag{9}$$

and

$$\hat{\epsilon}_{k_n}^{DPPNode}(f) = (M - k_n) \times \frac{(v^*)^2}{6}. \tag{10}$$

Lemma 3. *Assume (A1)-(A7). Let $t_1, \ldots, t_M$ denote the teacher hidden nodes and $l_1, \ldots, l_M$ denote the number of student hidden nodes in a node pruned network which learnt the corresponding teacher node. If $\sum_{m=1}^{M} l_m \leq M$, then the GE of this node pruned network is,*

$$\frac{(v^*)^2}{6} \left[\sum_{m=1}^{M} \left(1 - \frac{l_m}{Z} \right)^2 \right].$$

Lemma 4. *Assume* $(A1) - (A7)$. *Consider the random edge pruning method with parameter* $\lim_{N\to\infty} \frac{k_e}{N} = c$ *(here* c *is a constant between 0 and 1). Then the GE* $\epsilon_c^{Rand\ Edge}(\mathbb{E}[f])$ *is,*

$$\frac{M(v^*)^2}{\pi}\left[\frac{1}{Z}\arcsin\frac{c}{1+c} + \left(1-\frac{1}{Z}\right)\arcsin\frac{c^2}{1+c} + \frac{\pi}{6} - 2\arcsin\frac{c}{\sqrt{2(1+c)}}\right]. \tag{11}$$

Lemma 5. *Let* v^* *denotes the weight of the second layer of the teacher network and* $\{v_1, \cdots, v_K\}$ *be the weights of the student network after convergence. Then in the noiseless case for all* n *we have,* $v^* = \sum_{i\in G_n} v_i$.

D.2 Proof of the Thorems

Proof of Theorem 1. Now, we will prove Theorem 1. We will show, for any network pruned by Random Node, the GE is more than the expected GE of DPP Node pruning. Recall the randomly pruned network f discussed in the beginning of the proof. From Lemma 3 we can see that for node pruning the GE only depends on the number of nodes survived in each block. From equation 16 we have,

$$\begin{aligned}
\epsilon_{k_n}^{Rand\ Node}(f) &= \frac{(v^*)^2}{6}\left[\sum_{i=1}^{prn}\left(1-\frac{l_i}{Z}\right)^2\right] + \frac{(M-prn)(v^*)^2}{6} \\
&= \frac{(M-k_n)(v^*)^2}{6} + \sum_{i=1}^{prn}\left[(l_i-1)\frac{(v^*)^2}{6} + \frac{(v^*)^2}{6}\left(1-\frac{l_i}{Z}\right)^2\right] \\
&\geq \frac{(M-k_n)(v^*)^2}{6} + \sum_{i=1}^{prn} l_i\frac{(v^*)^2}{6}\left(1-\frac{1}{Z}\right)^2 \qquad (12) \\
&= \frac{(M-k_n)(v^*)^2}{6} + l\frac{(v^*)^2}{6}\left(1-\frac{1}{Z}\right)^2 \\
&= \epsilon_{k_n}^{DPP\ Node}(f)
\end{aligned}$$

where equation 12 follows from the inequality below:

$$(l_i-1)\frac{(v^*)^2}{6} + \frac{(v^*)^2}{6}\left(1-\frac{l_i}{Z}\right)^2 = l_i\frac{(v^*)^2}{6}\left[1+\frac{1}{Z^2}-\frac{2}{Z}\right] \geq l_i\frac{(v^*)^2}{6}\left(1-\frac{1}{Z}\right)^2$$

which proves the first part of Theorem 1. The proof for the reweighted network is similar.

In case of importance node pruning, the nodes with lowest absolute value of outgoing edges are dropped. Following Goldt et al. (2019) the outgoing weights of all the hidden teacher nodes are equal (we call it v^*). Also, from Lemma 5 we see that the sum of the weights of the outgoing edges of the student nodes which learn the same teacher node add up to the outgoing edge weight of the corresponding teacher hidden node. Moreover, we assume the ansatz $v_i = v_j$ when $i, j \in G_n$, where G_n denotes the set of student nodes which learn the same teacher node t_n. Hence, we can see that all the outgoing edges are approximately similar. We also verify this fact experimentally. Therefore, this defines an approximately uniform distribution on the set of hidden nodes. Hence, this is almost same as random node pruning and so the result follows from Theorem 1. □

Proof of Theorem 2. Lemma 2 and 4 provide the closed form of the GE after DPP node pruning and random node pruning respectively. Using this closed form we plot $\epsilon_{k_n}^{DPP\ Node}(f) - \epsilon_c^{Rand\ edge}(f)$ in Figure 4 A. Here k_n and c satisfy equation 1, i.e., parameter count is same after two kinds of pruning. We can see for $Z \geq 4$ this value is ≥ 0 given $0 \leq c \leq 1.0/Z$, which proves the theorem. □

Remark 1. *Our results hold for* $Z \geq 4$*, where* Z *is the number of student nodes which learn the same teacher node. This is because in DPP node pruning at most* 1 *student node survives per group. As a result for larger* Z *the lost information per group is higher (in the scale of* $\left(1-\frac{1}{Z}\right)^2$*).*

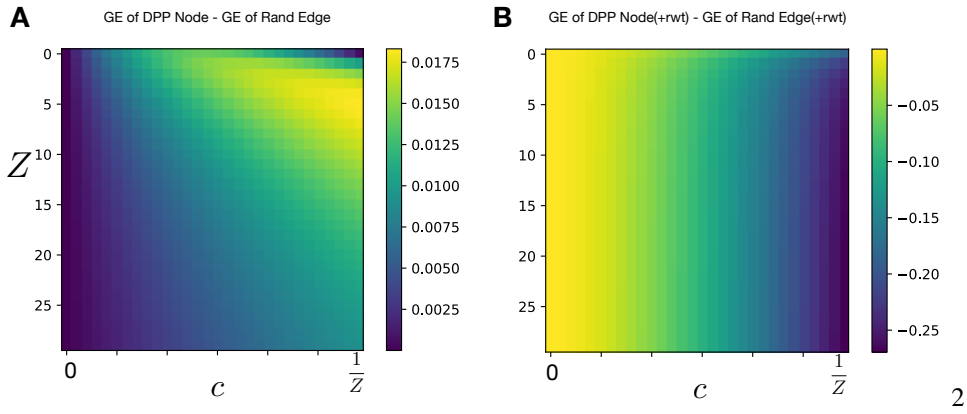

Figure 4: **(A)** Difference between the GE of DPP node pruning and Random edge pruning for $4 \geq Z \geq 30$. The matrix consist of only nonzero entries which proves that random edge pruning performs better than DPP node pruning when parameter count is same. **(B)** Difference between the GE of DPP node pruning with reweighting and Random edge pruning with reweighting for $4 \geq Z \geq 30$. The matrix consist of only negative entries which proves that random edge pruning can never perform better than DPP node pruning when reweighting is applied in the second layer.

Next we state the impossibility result as discussed inmain text. We will show that, no reweighting scheme in the second layer for random edge pruning which is based on scaling can beat DPP node pruning after reweighting. Formally we have the following:

Theorem 3. *Assume* $(A1) - (A7)$. *Let* k_n *and* c *satisfy equation 1, and* $0 \leq c \leq \frac{1}{Z}$ *and* $Z \geq 4$. *Assume the reweighting scheme for random edge in second layer such that,* $\hat{v}_i = Av_i$. *Then* $\forall A \in \mathbb{R}$ *we have,*

$$\hat{\epsilon}_{k_n}^{DPP\ Node}(f) \leq \hat{\epsilon}_c^{Rand\ Edge}(\mathbb{E}[f]) \tag{13}$$

Proof of Theorem 3. From Lemma 2 we know that the GE after rewighting the DPP node pruned network is

$$\frac{(v^*)^2}{6}(M - k_n) = \frac{M(v^*)^2}{6}(1 - Zc) \tag{14}$$

where c satisfies equation 1. Now for the given reweighting scheme in the hypothesis the GE for random edge pruning will be,

$$\frac{M(v^*)^2}{\pi}\left[A^2\left(\frac{1}{Z}\arcsin\frac{c}{1+c} + \left(1 - \frac{1}{Z}\right)\arcsin\frac{c^2}{1+c}\right) + \frac{\pi}{6} - 2A\arcsin\frac{c}{\sqrt{2(1+c)}}\right] \tag{15}$$

equation 15 can be viewed as a quadratic equation of A whose minimum correspond to the best reweighting scheme in the scaling family. In Figure 4 B we compare this minimum with equation 14. Formally we plotted $\hat{\epsilon}_{k_n}^{DPP\ Node}(f) - \hat{\epsilon}_c^{Rand\ Edge}(\mathbb{E}[f])$. It can be seen that this value is $-ve$ for all $0 \leq c \leq \frac{1}{Z}$, which implies GE of reweighted DPP node pruned network is always lower than reweighted random edge pruned network. □

E Proof of Lemmas

Proof of Lemma 2. Let $H_R = \{h_{i_1}, \ldots, h_{i_{k_n}}\}$ be the set of selected nodes by DPP Node pruning method. Recall from Goldt et al. (2019) that every student hidden node specializes in learning a teacher node. Denote $t(h)$ to be the teacher node learnt by h. $S_m \subseteq H_R$ be the set of selected hidden nodes of the pruned network which learnt the m^{th} teacher node , i.e., $S_m = \{h \in H_R | t(h) = t_m\}$ (t_m is the m^{th} teacher node). Hence, $prn = |\{\mathbb{1}(|S_m| > 0) | 1 \leq m \leq M\}|$ is the number of teacher nodes explained by the pruned network and W.L.O.G. we can assume that $t_1, \ldots, t_{prn}$ are those set of teacher nodes. Let $l_1, \ldots, l_{prn}$ be the number of student nodes in the pruned network which learn

the corresponding teacher node. Note that, $\sum_{i=1}^{prn} l_i = k_n$ and $l_i \leq Z$ (where Z is the number of student nodes dedicated to learn a single teacher node in the unpruned network) for all i. Applying Lemma 3 directly we can see that the GE for the pruned network is

$$\frac{(v^*)^2}{6}\left[\sum_{i=1}^{prn}\left(1-\frac{l_i}{Z}\right)^2\right]+\frac{(M-prn)(v^*)^2}{6} \tag{16}$$

The first part of equation 16 is the GE for the group whose corresponding teacher node is partially explained and the second part accounts for the GE due to unexplained teacher nodes (number of such teacher nodes are $M-prn$). From Lemma 1 we know that the expected kernel matrix for DPP Node pruning is the order parameter Q and it becomes a block diagonal matrix after the training converges, where size of each block is Z (which is also the number of student nodes dedicated to learn a single teacher node in the unpruned network). Because of the block diagonal property of the DPP kernel matrix, at most 1 student hidden node will be chosen from each block, i.e., $l_i = 1\ \forall i$. Hence, $prn = k_n$. From Lemma 3 we can see that the GE of node pruned network only depends on the number of student node survived in each block after pruning, and, for DPP node pruning, it is always 1 (given $k_n \leq M$). This is why there is no expectation in the GE term. So for DPP node pruning the GE is,

$$\epsilon_{k_n}^{DPP\ Node}(f) = (v^*)^2\left[\frac{k_n}{6}\left(1-\frac{1}{Z}\right)^2+\frac{M-k_n}{6}\right].$$

Each of the k_n student nodes in the pruned network learns a different teacher node. Consider one such teacher node and call it t_i. In the unpruned network, there are Z student hidden nodes which learn a single teacher node t_i, only one of which survives after DPP node pruning. The first part of the error is due to the removal of student nodes ($Z-1$ student nodes for each t_i). However, these errors can be retrieved by reweighting the survived student node. On the contrary, there are $M-k_n$ teacher nodes which don't have any representative (some student hidden node from the set of student nodes which specialized in this particular teacher node) in the pruned network. And the error (second part of the GE) due to those nodes can not be retrieved even after reweighting. Hence, the GE after reweighting becomes,

$$(M-k_n)\times\frac{(v^*)^2}{6}$$

Thus, we have the lemma 2. □

Proof of Lemma 3. Let $G_1,\ldots,G_M$ be the subsets of student nodes such that all student nodes in G_m learn the m^{th} teacher node. From the assumption we have, $|G_m| = Z$ for all m. After pruning, a subset $P_m \subseteq G_m$ is chosen, and $|P_m| = l_m$. Denote the order parameters of the pruned network as Q', R', T'. For node pruning we can see that

$$Q'_{ik} = \begin{cases} Q_{ik} & \text{if } \exists m \text{ s.t. } h_i \in P_m \text{ and } h_k \in P_m \\ 0 & \text{otherwise} \end{cases}$$

Also, for the unpruned network we have

$$Q_{ik} = \begin{cases} 1 & \text{if } \exists m \text{ s.t. } h_i \in G_m \text{ and } h_k \in G_m \\ 0 & \text{otherwise} \end{cases}$$

Now from equation 5 we can break down the GE into three parts. From equation 6, equation 7 and equation 8 we have,.

$$f_1(Q') = \frac{1}{\pi}\sum_{i,k} v_i v_k \arcsin\frac{Q'_{ik}}{\sqrt{1+Q'_{ii}}\sqrt{1+Q'_{kk}}} = \frac{1}{\pi}\sum_{n=1}^{M}\sum_{i,k\in P_n} v_i v_k \arcsin\frac{1}{2}, \tag{17}$$

$$= \frac{(v^*)^2}{6}\sum_{n=1}^{M}\left(\frac{l_i}{Z}\right)^2 \tag{18}$$

equation 17 follows from the fact that h_i and h_k belong to the same group G_n. So we have,

$$\frac{Q'_{ik}}{\sqrt{1+Q'_{ii}}\sqrt{1+Q'_{kk}}} = \frac{1}{\sqrt{2}\sqrt{2}} = \frac{1}{2}$$

We can also see that equation 18 follows from Lemma 5 and the ansatz $v_i = v_j$ when $i, j \in G_n$. The order parameters T_{nm} doesn't change after pruning, and so we have,

$$f_2(T') = \frac{1}{\pi}\sum_{n,m} v_n^* v_m^* \arcsin \frac{T_{nm}}{\sqrt{1+T_{nn}}\sqrt{1+T_{mm}}} = \frac{1}{6}\sum_{n=1}^{M}(v^*)^2 \tag{19}$$

And similarly,

$$f_3(R', Q', T') = \frac{2}{\pi}\sum_{i,n} v_i v_n^* \arcsin \frac{R'_{in}}{\sqrt{1+Q'_{ii}}\sqrt{1+T'_{nn}}} = \frac{2}{6}\sum_{n=1}^{M} v_n^* \sum_{i\in P_n} v_i. \tag{20}$$

Then from equation 18,equation 19 and equation 20 the GE of node pruning is,

$$\frac{(v^*)^2}{6}\left[\sum_{m=1}^{M}\left(1-\frac{l_m}{Z}\right)^2\right]. \tag{21}$$

Hence we have the lemma. □

Intuitively, this lemma states that for teacher hidden node t_n if l_n student hidden nodes survive after node pruning, then the fraction of information lost due to the deletion of nodes is $1 - \frac{l_n}{Z}$, where Z is the number of student nodes learn a particular teacher node in the unpruned network.

Proof of Lemma 5. From $(S36)$ of Goldt et al. (2019) we have,

$$\frac{dv_i}{dt} = \eta_v\left[\sum_{n=1}^{M} v_n^* I_2(i,n) - \sum_{j=1}^{K} v_j I_2(i,j)\right] = \eta_v \arcsin\frac{1}{2}\left[v^* - \sum_{j\in G_n} v_j\right]$$

Hence, a fixed point (in terms of v_i's) of the ODE is,

$$\{(v_1,\ldots,v_K)|\sum_{i\in G_n} v_i = v^*, \forall 1 \le n \le M)\}$$

□

Intuitively, this lemma states that the sum of the outgoing edges of the student hidden nodes which learn a particular teacher hidden node is approximately equal to the weight of the outgoing edge of that teacher hidden node.

Lemma 6. *Let Q, R, T are the order parameters of the unpruned network, and Q', R', T' are the respective order parameters after applying the Random Edge pruning where c fraction of the edges are kept. Then we have 1) $\mathbb{E}[Q'_{ik}] = cQ_{ik}$ if $i = k$ and $c^2 Q_{ik}$ otherwise , 2) $\mathbb{E}[R'_{st}] = cR_{st}$ and 3) $T'_{mn} = T_{mn}$.*

Proof. Follows directly from the pruning procedure. □

Intuitively, this lemma states that the order parameters of the pruned network using random edge pruning is a scaled version of the order parameters of the unpruned networks. However, the scaling of diagonal elements are different from that of off-diagonal elements.

Proof of Lemma 4. In this theorem, we will give the GE of the expected network pruned by the Random Edge method. Pruning is performed on the edges between input layer and the hidden layer. Hence, the order parameter changes. From Lemma 6, we have the order parameters of the expected

network (call these Q', R', T'). However, the weights of the second layer remain unchanged. Putting these values in equation 6, equation 7 and equation 8 we have,

$$f_1(Q') = \frac{1}{\pi}\sum_{i,k} v_i v_k \arcsin \frac{Q'_{ik}}{\sqrt{1+Q'_{ii}}\sqrt{1+Q'_{kk}}}$$
$$= \frac{M(v^*)^2}{\pi}\arcsin\frac{c^2}{1+c} + \frac{M(v^*)^2}{Z\pi}\left[\arcsin\frac{c}{1+c} - \arcsin\frac{c^2}{1+c}\right] \tag{22}$$

and,

$$f_3(R', Q', T') = \frac{2}{\pi}\sum_{i,n} v_i v_n^* \arcsin\frac{R'_{in}}{\sqrt{1+Q'_{ii}}\sqrt{1+T'_{nn}}} = \frac{2M(v^*)^2}{\pi}\arcsin\frac{c}{\sqrt{2(1+c)}} \tag{23}$$

Therefore, the GE of the expected network after Random Edge pruning is,

$$\frac{M(v^*)^2}{\pi}\left[\arcsin\frac{c^2}{1+c} + \frac{\pi}{6} - 2\arcsin\frac{c}{\sqrt{2(1+c)}}\right] + \frac{M(v^*)^2}{Z\pi}\left[\arcsin\frac{c}{1+c} - \arcsin\frac{c^2}{1+c}\right]$$

This proves the first part of the theorem. □