Supplementary: Statistical Mechanical Analysis of Neural Network Pruning

Rupam Acharyya[1] **Ankani Chattoraj**[*2] **Boyu Zhang**[*1] **Shouman Das**[3] **Daniel Štefankovič**[1]

[1]Computer Science Dept., University of Rochester, Rochester, New York, USA
[1]Brain anf Cognitive Science Dept., University of Rochester, Rochester, New York, USA
[1]Mathematics Dept., University of Rochester, Rochester, New York, USA

A GE IN TWO LAYER NETWORK

For the theoretical analysis we consider the following assumptions from [2]

(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) M, K denotes the number of hidden nodes for the teacher and student network respectively and $K \geq M$ 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 satisfy the ansatz as in (S58) - (S60) of [2].
(A7) No noise is added to the labels generated by the teacher network, i.e., $\sigma = 0$.

With the above assumptions, authors of [2] gave a closed form of the GE as follows:

$$\epsilon_g = f_1(Q) + f_2(T) - f_3(R, Q, T) \tag{1}$$

where,

$$f_1(Q) = \frac{1}{\pi} \sum_{i,k} v_i v_k \arcsin \frac{Q_{ik}}{\sqrt{1 + Q_{ii}}\sqrt{1 + Q_{kk}}} \tag{2}$$

$$f_2(T) = \frac{1}{\pi} \sum_{n,m} v_n^* v_m^* \arcsin \frac{T_{nm}}{\sqrt{1 + T_{nn}}\sqrt{1 + T_{mm}}} \tag{3}$$

$$f_3(R, Q, T) = \frac{2}{\pi} \sum_{i,n} v_i v_n^* \arcsin \frac{R_{in}}{\sqrt{1 + Q_{ii}}\sqrt{1 + T_{nn}}} \tag{4}$$

where Q, R, T are the order parameters as defined in main text. We also have the assumption (5) about the relation between number of edges and nodes kept after pruning.

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., they satisfy,

$$\frac{k_n}{K} = \lim_{N \to \infty} \frac{k_e}{N} = c, \tag{5}$$

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

[*]equal contribution

Supplement for the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021).

Table 1: Notations used in Theorems

Notations	Explanations	Notations	Explanations	Notations	Explanations
n	number of inputs	N	dimension of the input	n_l	number of nodes in layer l
v_i^l	i^{th} node in layer l $(1 \leq i \leq n_l)$	a_{ij}^l	activation of v_i^l on j^{th} input	M	number of teacher hidden nodes
e_{ij}^l	edge from v_i^l to v_j^{l+1} $(1 \leq i \leq n_l$ and $1 \leq j \leq n_{l+1})$	w_{ij}^l	weight of e_{ij}^l $(1 \leq i \leq n_l$ and $1 \leq j \leq n_{l+1})$	K	number of student hidden nodes
k_n	number of student hidden nodes kept after node pruning	k_e	number of incoming edges of a hidden node kept after edge pruning	v^*	second layer weight of teacher network

B 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

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

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 [2] 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.

C PROOF OF THE THEOREMS

Theorem 1. *Assume $(A1) - (A7)$. Let $k_n \leq M$ nodes are selected by the DPP Node pruning method,*

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

and

$$\hat{\epsilon}_{k_n}^{DPPNode}(f) = (M - k_n) \times \frac{(v^*)^2}{6}. \tag{7}$$

Theorem 2. *Assume $(A1) - (A7)$. Then for $k_n \leq M$ we have,*

$$\mathbb{E}_f\left[\epsilon_{k_n}^{Rand\ Node}(f)\right] > \epsilon_{k_n}^{DPP\ Node}(f') \tag{8}$$

and

$$\mathbb{E}_f\left[\hat{\epsilon}_{k_n}^{Rand\ Node}(f)\right] > \hat{\epsilon}_{k_n}^{DPP\ Node}(f') \tag{9}$$

and,

$$\epsilon_{k_n}^{Imp\ Node}(f') > \hat{\epsilon}_{k_n}^{DPP\ Node}(f'), \tag{10}$$

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.

Proof of Theorem 1 and 2. Let $H_R = \left\{h_{i_1}, \ldots, h_{i_{k_n}}\right\}$ be the set of selected nodes by DPP Node pruning method. Recall from [2] 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 2 directly we can see that the GE for the pruned network is

$$\frac{(v^*)^2}{6}\left[\sum_{i=1}^{prn}\left(1 - \frac{l_i}{Z}\right)^2\right] + \frac{(M - prn)(v^*)^2}{6} \tag{11}$$

The first part of (11) 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 2 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 Theorem 1.

Next, we will prove Theorem 2. 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 2

we can see that for node pruning the GE only depends on the number of nodes survived in each block. From (11) 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 (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 2. 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 [2] the outgoing weights of all the hidden teacher nodes are equal (we call it v^*). Also, from Lemma 3 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 2. □

Remark 1. *The comparison between performance of importance node pruning and DIVNET depends on the fact that all the outgoing edges of the teacher hidden nodes are equal. However, when the outgoing weights are not equal the importance pruning first selects student hidden nodes from a group whose corresponding teacher node has the highest weight. Once all the student nodes are selected from that group then it selects the group whose corresponding teacher node has second highest outgoing edge weight and the process continues. Because of this approach, even without reweighting a complete information about the teacher node is preserved in the pruned network. However, in DPP node pruning one candidate from each group (representing a particular teacher node) is selected first. But if a member is selected from a group then the reweighting method can recover the complete lost information for the corresponding group. Hence, DIVNET is able to preserve information about more number of teacher hidden nodes than importance pruning which results in better performance.*

Theorem 3. *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* $\epsilon_c^{Rand\ Edge}\left(\mathbb{E}\left[f\right]\right)$ *is,*

$$\begin{aligned}
&\frac{M(v^*)^2}{\pi}\left[\frac{1}{Z}\arcsin\frac{c}{1+c} + \left(1-\frac{1}{Z}\right)\arcsin\frac{c^2}{1+c}\right. \\
&\left. + \frac{\pi}{6} - 2\arcsin\frac{c}{\sqrt{2(1+c)}}\right]. \qquad (13)
\end{aligned}$$

Proof of Theorem 3. 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 4, we have the order parameters of the expected network (call these Q', R', T'). However, the weights of the second

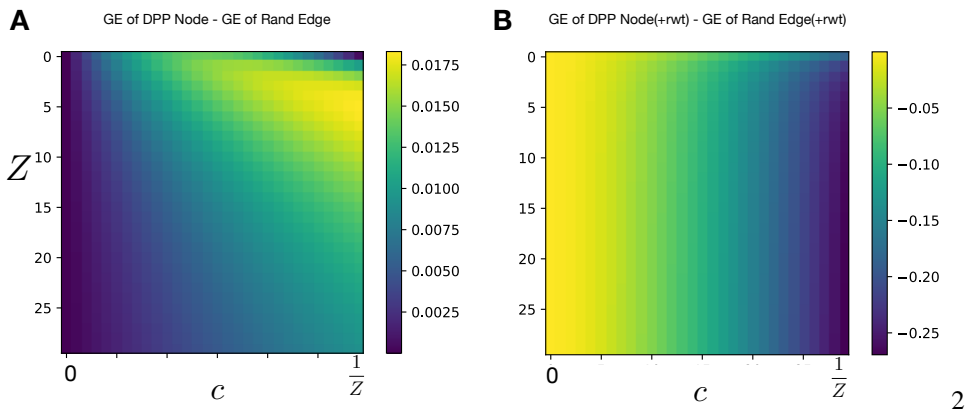

Figure 1: **(A)** Difference between the GE of DPP node pruning and Random edge pruning for $4 \geq Z \geq 30$. The matrix consist of only nonzero entries which proves that random edge pruning performs better than DPP node pruning when parameter count is same. **(B)** Difference between the GE of DPP 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.

layer remain unchanged. Putting these values in (2), (3) and (4) we have,

$$\begin{aligned} f_1(Q') &= \frac{1}{\pi} \sum_{i,k} v_i v_k \arcsin \frac{Q'_{ik}}{\sqrt{1+Q'_{ii}}\sqrt{1+Q'_{kk}}} \\ &= \frac{M(v^*)^2}{\pi} \arcsin \frac{c^2}{1+c} + \frac{M(v^*)^2}{Z\pi} \left[\arcsin \frac{c}{1+c} - \arcsin \frac{c^2}{1+c} \right] \end{aligned} \tag{14}$$

and,

$$\begin{aligned} f_3(R', Q', T') &= \frac{2}{\pi} \sum_{i,n} v_i v_n^* \arcsin \frac{R'_{in}}{\sqrt{1+Q'_{ii}}\sqrt{1+T'_{nn}}} \\ &= \frac{2M(v^*)^2}{\pi} \arcsin \frac{c}{\sqrt{2(1+c)}} \end{aligned} \tag{15}$$

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. □

Theorem 4. *Assume* $(A1) - (A7)$. *Let* k_n *and* c *satisfy* (5)*, and* $0 \leq c \leq \frac{1}{Z}$ *and* $Z \geq 4$. *Then*

$$\epsilon_{k_n}^{DPP\ Node}(f) \geq \epsilon_c^{Rand\ Edge}(\mathbb{E}[f]), \tag{16}$$

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

Proof of Theorem 4. Theorem 1 and 3 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 1 A. Here k_n and c satisfy (5), 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 2. *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* $\left(1 - \frac{1}{Z}\right)^2$*).*

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 5. *Assume* $(A1)-(A7)$. *Let* k_n *and* c *satisfy* (5)*, 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,*

$$\hat{\epsilon}_{k_n}^{DPP\ Node}(f) \leq \hat{\epsilon}_c^{Rand\ Edge}(\mathbb{E}[f]) \tag{17}$$

Proof of Theorem 5. From Theorem 1 we know that the GE after rewighting the DPP node pruned network is

$$\frac{(v^*)^2}{6}(M-k_n) = \frac{M(v^*)^2}{6}(1-Zc) \tag{18}$$

where c satisfies (5). Now for the given reweighting scheme in the hypothesis the GE for random edge pruning will be,

$$\frac{M(v^*)^2}{\pi}\left[A^2\left(\frac{1}{Z}\arcsin\frac{c}{1+c}+\left(1-\frac{1}{Z}\right)\arcsin\frac{c^2}{1+c}\right)+\frac{\pi}{6}-2A\arcsin\frac{c}{\sqrt{2(1+c)}}\right] \tag{19}$$

(19) can be viewed as a quadratic equation of A whose minimum correspond to the best reweighting scheme in the scaling family. In Figure 1 B we compare this minimum with (18). 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. □

D PROOF OF LEMMAS

Lemma 2. *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,*

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

Proof. 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 (1) we can break down the GE into three parts. From (2), (3) and (4) we have,.

$$
\begin{aligned}
f_1(Q') &= \frac{1}{\pi}\sum_{i,k} v_i v_k \arcsin\frac{Q'_{ik}}{\sqrt{1+Q'_{ii}}\sqrt{1+Q'_{kk}}}, \\
&= \frac{1}{\pi}\sum_{n=1}^{M}\sum_{i,k\in P_n} v_i v_k \arcsin\frac{1}{2}, \qquad (20)\\
&= \frac{1}{\pi}\sum_{n=1}^{M}\sum_{i,k\in P_n} v_i v_k \frac{\pi}{6}, \\
&= \frac{1}{6}\sum_{n=1}^{M}\left(\sum_{i\in P_n} v_i\right)^2, \\
&= \frac{(v^*)^2}{6}\sum_{n=1}^{M}\left(\frac{l_i}{Z}\right)^2 \qquad (21)
\end{aligned}
$$

(20) 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 (21) follows from Lemma 3 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,

$$
\begin{aligned}
f_2(T') &= \frac{1}{\pi}\sum_{n,m} v_n^* v_m^* \arcsin\frac{T_{nm}}{\sqrt{1+T_{nn}}\sqrt{1+T_{mm}}}, \\
&= \frac{1}{6}\sum_{n=1}^{M}(v^*)^2 \qquad (22)
\end{aligned}
$$

And similarly,

$$
\begin{aligned}
f_3(R', Q', T') &= \frac{2}{\pi}\sum_{i,n} v_i v_n^* \arcsin\frac{R'_{in}}{\sqrt{1+Q'_{ii}}\sqrt{1+T'_{nn}}}, \\
&= \frac{2}{\pi}\sum_{n=1}^{M} v_n^* \sum_{i\in P_n} v_i \arcsin\frac{1}{2}, \\
&= \frac{2}{6}\sum_{n=1}^{M} v_n^* \sum_{i\in P_n} v_i. \qquad (23)
\end{aligned}
$$

Then from (21),(22) and (23) the GE of node pruning is,

$$\frac{(v^*)^2}{6}\left[\sum_{m=1}^{M}\left(1-\frac{l_m}{Z}\right)^2\right]. \qquad (24)$$

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.

Lemma 3. *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,*

$$v^* = \sum_{i\in G_n} v_i$$

Proof of Lemma 3. From $(S36)$ of [2] 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,

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

□

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 4. *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 the following:*

- $$\mathbb{E}[Q'_{ik}] = \begin{cases} cQ_{ik} & \text{if } i = k \\ c^2 Q_{ik} & \text{otherwise} \end{cases}$$
- $\mathbb{E}[R'_{st}] = cR_{st}$
- $T'_{mn} = T_{mn}$

Proof. In case of Random Edge pruning each edge is kept with probability c. Then we have,

$$\mathbb{E}[Q'_{st}] = \frac{1}{N} \sum_{i=1}^{N} c \cdot w_{is} \times c \cdot w_{it} = c^2 \frac{1}{N} \sum_{i=1}^{N} w_{is} w_{it} = c^2 Q_{st}$$

and

$$\mathbb{E}[Q'_{ss}] = \frac{1}{N} \sum_{i=1}^{N} c^2 \cdot w_{ss}^2 = c^2 Q_{ss}.$$

Similarly,

$$\mathbb{E}[R'_{st}] = \frac{1}{N} \sum_{i=1}^{N} c \cdot w_{is} w_{it}^* = cR_{st}$$

The teacher node is not affected by the pruning. So T is not modified by the pruning process. This proves the lemma. □

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 (for more see Figure 3 A).

E SIMULATION DETAILS

In total, 10 rounds of simulations are run for each of the 5 pruning methods, and we report the average and standard deviations (as error bars). The standard deviations are negligible (in the magnitude of 10^{-3}). A *round* is the entire process of generating a new teacher network with datasets, training the student from scratch, performing pruning and finally testing with the pruned network. For DPP and random methods, we sampled 100 masks per round and reported the average performance in each round. Given $M = 2$ and $K = 6$, we tried pruning with $[1, 2, 3, 4, 5]$ nodes (and the equivalent number of edges) left in the student, respectively. We keep the total number of weights same to compare different pruning methods. The

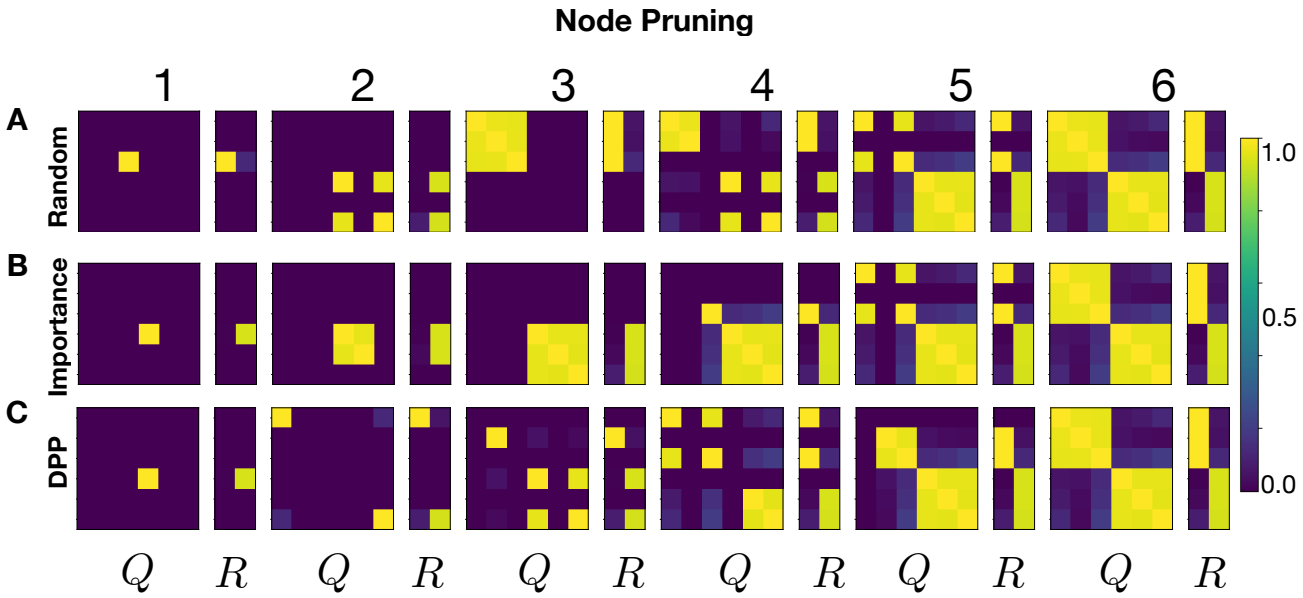

Figure 2: Order parameters after different node pruning methods in the teacher student setup. For this example, number of student hidden nodes $M = 2$ and number of teacher hidden nodes $K = 6$. From [3] we know that the first 3 student nodes (call 1^{st} group) learn one teacher node, and the next 3 (call 2^{nd} group) learn the second teacher node. Recall that k_n is the number of student hidden nodes survived after node pruning. In this figure each row represents a particular node pruning method and each column (Q, R) shows results for different choices of k_n (left to right goes from most pruned to unpruned network). **(A)** In case of random node pruning when $k_n = 2$, two student node survives from the 2^{nd} group after pruning. As a result, information about the 1^{st} teacher node is completely lost in the pruned network. **(B)** Importance pruning keeps a student hidden node depending on its outgoing edge weights. The outgoing edge weights of each group is almost equally distributed among themselves, and they sum up to the second layer weight of corresponding teacher node (see Lemma 3). As all the group size is equal (3 for this example), importance node pruning first selects node from the group whose corresponding second layer teacher weight is highest. In our example, it is the second group and hence for $k_n = 1, 2, 3$, it selects node from the second group. Once a group is exhausted, it then selects from another group according to the aforementioned policy and so on. **(C)** For DPP node pruning when $k_n = 2$, two student hidden nodes are chosen from different groups which preserve information about both the teacher nodes. It can also be shown that, in case of node pruning, if at least one representative from a group survives after pruning, then the reweighting can recover the complete information about that block. Hence, in teacher student framework DPP node pruning performs the best among the node pruning methods especially after reweighting.

node-to-edge ratio, given $N = 500$, $K = 6$, and $M = 2$, is $[1 : 83, 2 : 166, 3 : 250, 4 : 333, 5 : 417]$. This is calculated, for the teacher-student setup (single output node) specifically, as $k_e = \frac{k_n(1+N)-K}{K}$. We grid-searched η in the range of $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$ and found 0.50 to be the optimal. We used $\beta = 0.3$ for all DPP node kernel calculations in all simulations.

F HYPERPARAMETERS FOR REAL DATASETS

Besides the hyperparameters and setup we proposed in Section 5.2 on the synthetic dataset, we report the hyperparameters used for the results on the `MNIST` and `CIFAR10`. As stated in Section 5.2, we used that exact same experiment setup (network architectures, error thresholds, etc.) as in [3] for fair and consistent comparisons. We used SGD optimizers, a learning rate of 0.001, and a momentum of 0.9 for traning on both datasets. For `MNIST`, the training batch size was 1000. For `CIFAR10`, the training batch size was 128. All pruning methods were performed 10 times, and we report the means and standard deviations in Figure 4 (with reweighting).

The node-to-edge ratio for pruning, which keeps the number of parameters in the pruned network the same, is $[397 : 614, 472 : 921, 548 : 1228, 623 : 1536, 699 : 1843, 774 : 2150, 849 : 2457, 925 : 2764]$ for `CIFAR10` and $[256 : 156, 287 : 235, 317 : 313, 348 : 392, 378 : 470, 409 : 548, 439 : 627, 470 : 705]$ for `MNIST`, given the network architecture

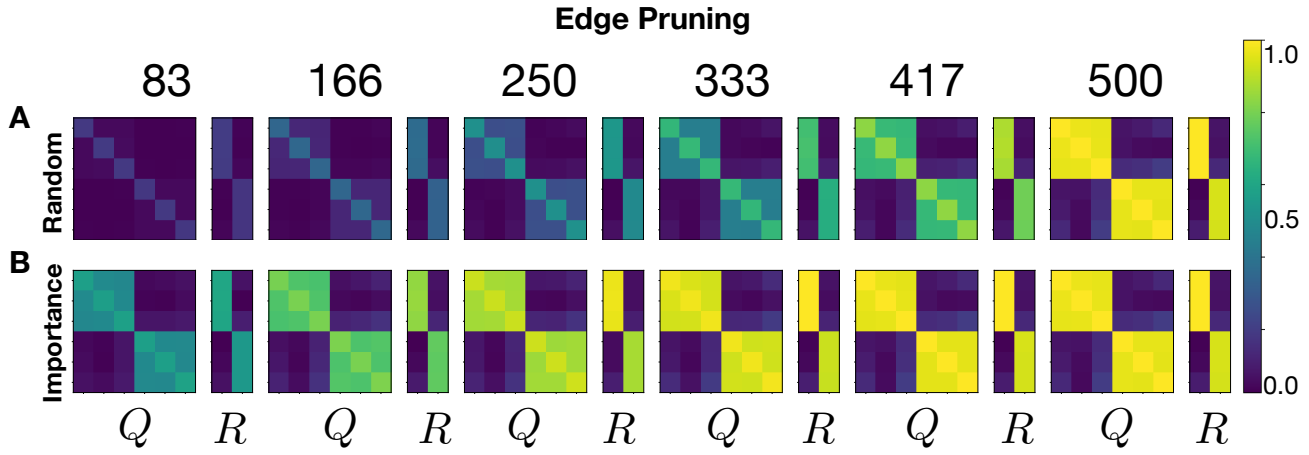

Figure 3: Order parameters after different edge pruning methods in the teacher student setup. For this example, number of student hidden nodes $M = 2$ and number of teacher hidden nodes $K = 6$. From [3] we know that the first 3 student nodes (call 1^{st} group) learn one teacher node, and the next 3 (call 2^{nd} group) learn the second teacher node. Recall that k_e is the number of incoming edges for each student hidden nodes survived after edge pruning. In this figure each row represents a particular edge pruning method and each column (Q, R) shows results for different choices of k_e (left to right goes from most pruned to unpruned network). **(A)** In case of random edge pruning, the expected order parameters have the form described in Lemma 4. **(B)** Order parameters for importance edge pruning. For importance edge pruning, the edges with lowest absolute values are removed. As the input dimension goes to infinity, the order parameters of the pruned network are close to that of the unpruned network ($k_e = 500$). In particular, for any fix k_e, let $Q^{imp}_{k_e}$ be the order parameter of the pruned network when importance pruning is used. $Q^{rand}_{k_e}$ is defined similarly. Our simulations show that, $\left\|Q^{unpruned} - Q^{imp}_{k_e}\right\| \leq \left\|Q^{unpruned} - Q^{rand}_{k_e}\right\|$. This is why the blocks in the Q matrix are the brightest in case of importance pruning. Hence, importance edge pruning performs the best without reweighting.

in Table 1 of [3]. These ratios correspond to 20% to 90% of the edges left for each node, as shown on the x-axis of Figure 4. These node-to-edge ratios are calculated based on the conversion equation in Section 5.2. We used $\beta = 10/|T|$ where $|T|$ is the size of the training dataset for all DPP node and edge kernel calculations on real data, following the choice of [3].

G TABLES AND FIGURES

Table 2 shows the experimental results on the synthetic data with the setup discussed in main text. For all the node-to-edge ratios in (5), given $K = 6$ and $M = 2$, we calculated the mean square GEs for both the noiseless and noisy case ($\sigma = 0.25$). We sampled 100 masks per simulation, and there are in total 10 rounds of simulations. As mentioned earlier, DPP methods are stable, and the standard deviations are in the magnitude of 10^{-3} for all ratios.

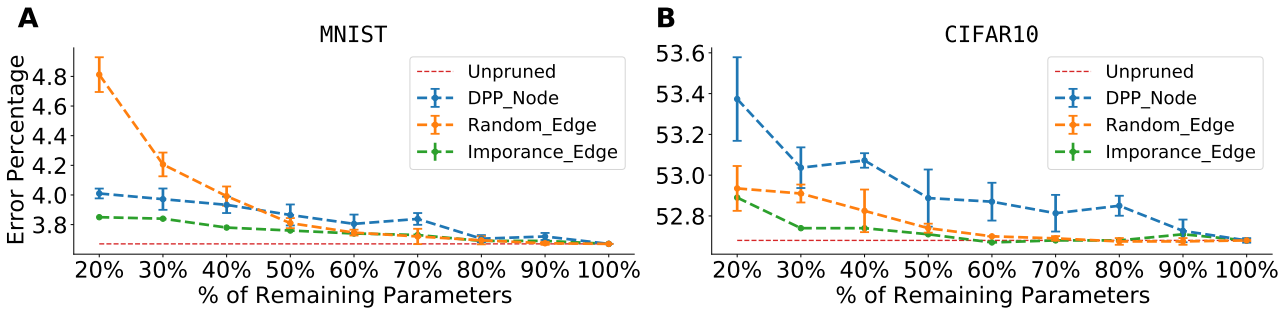

Figure 4: Comparing different edge pruning methods with DPP Node pruning method on the `MNIST` (A) and `CIFAR10` (B) dataset. Horizontal axis represents the percentage of remaining parameters in 1^{st} layer after pruning. The vertical axis shows corresponding test error. Both magnitude based edge pruning method (importance pruning) and baseline random edge pruning method outperforms DPP Node pruning which confirms Theorem 4 and the conjecture proposed in [1].

Table 2: The mean square GE on synthetic data for all pruning methods. The left-most row indicates the percentage of parameters left in the network. For specific node-to-edge ratio, see 5. The upper table shows the noiseless case, and the lower shows the noisy case ($\sigma = 0.25$). We also observed the implicit regularization effects of pruning proposed by [3]

% OF PARAMETERS	DPP NODE	RAND. EDGE	RAND. NODE	IMP. EDGE	IMP. NODE
17.0%	3.737± 0.009	3.451± 0.011	3.978 ± 0.016	1.911	3.760
33.0%	2.310± 0.012	2.300± 0.015	2.800 ± 0.035	0.814	2.719
50.0%	1.438± 0.015	1.402± 0.006	1.748 ± 0.036	0.311	1.540
67.0%	0.740± 0.017	0.730± 0.006	1.046 ± 0.018	0.110	0.721
83.0%	0.258± 0.008	0.204± 0.005	0.540 ± 0.010	0.040	0.360
ORIGINAL TEST LOSS: 0.051 (NOISELESS)					
17.0%	4.000± 0.005	3.769± 0.012	4.188 ± 0.001	1.963	4.167
33.0%	2.622± 0.015	2.558± 0.011	3.041 ± 0.024	0.905	2.910
50.0%	1.633± 0.002	1.675± 0.010	2.023 ± 0.035	0.450	2.031
67.0%	0.890± 0.018	1.007± 0.007	1.269 ± 0.022	0.271	1.144
83.0%	0.394± 0.001	0.490± 0.003	0.643 ± 0.002	0.253	0.659
ORIGINAL TEST LOSS: 0.241 ($\sigma = 0.25$)					

REFERENCES

[1] 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.

[2] 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*, pages 6979–6989, 2019.

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