Sparse Neural Architectures and Deterministic Ramanujan Graphs

Anonymous authors
Paper under double-blind review

Abstract

We present a sparsely connected, neural network architecture constructed using the theory of Ramanujan graphs which provide comparable performance to a dense network. The deterministic Ramanujan graphs occur either as Cayley graphs of certain algebraic groups or as Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph on $k + l$ vertices. The bipartite graphs represent the convolution and the fully connected layers retaining desirable structural properties like path connectivity and symmetricity. The method is novel as a zero-shot, data independent, deterministic pruning at initialization technique. The approach helps in early identification of winning lottery tickets, unlike previous techniques which typically determine them in an iterative fashion. We demonstrate experimentally that the proposed architecture provides competitive accuracy and sparsity ratio with those achieved by previous pre-training pruning algorithms.

1 Introduction

Sparse neural architectures are attractive due to their parameter parsimony and reduced training time. Existence of sparse high performing subnetworks of a backbone dense network forms the basis of the well known lottery ticket hypothesis (Frankle & Carbin, 2019). Several approaches have been directed towards identifying winning lottery tickets with a minimal effort. Initial research were based on applying established pruning algorithms on a partially trained network (Renda et al., 2020; Fischer & Burkholz, 2022). Recently, a number of approaches has been suggested to obtain a sparse mask for pruning at initialization (PaI) (Frankle et al., 2020; Wang et al., 2021; Sreenivasan et al., 2022). These method use the structure of the initialized network, in a data dependent or independent manner, to prune the network to a high sparsity ratio (Sreenivasan et al., 2022; Lee et al., 2019a;b; Wang et al., 2020; Tanaka et al., 2020). Most of these techniques are multi-shot, obtaining desired connectivity structures from a random network initialization. Zero-shot pruning aims to construct an initialization topology without the need for iteration over network structures. We show that deterministic constructions of Ramanujan expander graphs can be effectively used for zero-shot pruning.

Expander graphs are connected sparse networks (Hoory et al., 2006) with bounded expansion factors. Higher spectral gap between the first and the second eigenvalues of a graph adjacency matrix points towards a better expansion. Ramanujan graphs (Lubotzky et al., 1988) are a class of regular spectral expanders with maximally high spectral gaps. It has been empirically shown that the expansion property is strongly correlated with the performance of sparse neural networks (Prabhu et al., 2018b; Pal et al., 2022).

In general, the expander networks provide a sparse initialization architecture which may be trained to a high accuracy (Stewart et al., 2023; Esguerra et al., 2023; Prabhu et al., 2018b). Spectral sparsification is a method of obtaining such expander like neural networks (Laenen, 2023). Most of the expander networks used for this purpose are obtained by first generating random bipartite graphs for each layer, and then selecting the ones with a large spectral gap. This is based on the fact that random graphs are weakly Ramanujan (a conjecture of Alon, proved by Freidman). However, the expander based techniques mentioned above often favors random network initialization which are sensitive to random reinitialization and rewiring (Ma et al., 2021). Additional spectral measures are necessary to arrest these possibilities (Hoang et al., 2023).

We propose a deterministic sparse network initialization technique based on Ramanujan graphs that are constructed either as Cayley graphs of certain algebraic groups or as Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph on $k+l$ vertices. Prior approaches to using Ramanujan expander graphs for PaI have relied on constructions based on iterated magnitude pruning techniques. This often leads to the formation of irregular graph networks that do not strictly adhere to the rigorous definition of Ramanujan graphs. Our approach of constructing a deterministic Ramanujan network circumvents this problem. Ramanujan initializers using these bipartite graphs suitably represent the fully connected as well as the convolutional layers.

Deterministic Ramanujan graph based sparse network initialization has several advantages. Path connectedness and regularity is guaranteed by our graph construction technique. This ensures good performance even at very low remaining weight ratios. The sparse networks generated are data independent, structurally pre-defined, with a static mask across the training iterations. The deterministic construction algorithm does not degenerate to random networks.

Experimental results on benchmark image classification data sets show that Ramanujan sparse network initialization provides comparable performance with dense networks. The paper is organized as follows. We present a brief literature survey in the next section. Contributions of the paper are highlighted next. The properties and mathematical formulation of deterministic Ramanujan graphs are then presented, along with the construction techniques of sparse neural network layers. Finally, the experimental results are outlined.

1.1 Related Work

Pruning at initialization (PaI) has been well studied in literature (Cheng et al., 2023). The baseline consists of random pruning techniques based on either uniform edge sampling or Erdos-Renyi graphs (Liu et al., 2022; Evci et al., 2020; Mocanu et al., 2018; Gadhikar et al., 2023). More advanced techniques like SNIP use edge sensitivities (Lee et al., 2019b). Gradient flows over the edge weights are used in recent techniques like GraSP Wang et al. (2020), and SynFlow (Tanaka et al., 2020).

Expander based winning lottery ticket generation has been studied in (Stewart et al., 2023). The methodology is based on generating random d-regular graphs for the bipartite layers. These graphs are Ramanujan with a high probability. A deep expander sparse network, the X-Net, is presented in (Prabhu et al., 2018b). It is constructed by sampling d-left regular graphs from the space of all bipartite graphs. Ramanujan graph based sparsity aware network initialization is proposed in (Esguerra et al., 2023).

One-shot neural network pruning using spectral sparsification is presented in (Laenen, 2023). It is based on the effective resistance algorithm for obtaining spectrally sparse bipartite graphs. RadiX-Net (Kepner & Robinett, 2019) is a deterministic sparse neural architecture with mixed-radix topologies. It has desirable symmetry properties that preserves path connectedness and eliminates training bias. Connectivity properties are used in other graph theoretic initialization schemes that define an initial sparse network topology (Vysogorets & Kempe, 2023; Chen et al., 2022; 2023).

2 Research Gap and Contributions

Existing pruning at initialization techniques are often iterative and data dependent. Zero-shot data independent algorithms have advantages in terms of reduced computational overhead and generalization capabilities. Recently, there has been a flurry of works on the construction of pruned sparse networks based on various graph theoretic properties including expansion, path-connectivity, symmetry (Prabhu et al., 2018a; Kepner & Robinett, 2019; Pal et al., 2022; Stewart et al., 2023; Arnav Kalra et al., 2024). These methods use either a random network structure or are data dependent. None could guarantee the following three properties at the same time: (i) Ramanujan property - allows us to construct the best possible expanders given a set of vertices and maintaining a high level of sparsity, (ii) Path-connectedness - a desirable property for all PaI architectures, and, (iii) High symmetricity - a desirable property for computational purposes. *This is the first such implementation of deterministic Ramanujan graph based neural networks.* It should be pointed out that these Ramanujan graphs cannot be obtained using random sampling as random regular graphs are not known to be Ramanujan.

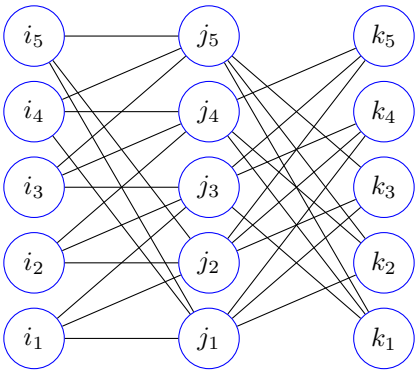


Figure 1: An example of a double layered regular graph where each bi-partite layer is Ramanujan. Note that one obtains something stronger than layer-wise path-connectivity. From each vertex of layer I one has at least $d_1 d_2$ pathways to reach layer K where d_1, d_2 denotes the regularities of layers I, J respectively. For instance, to reach layer K from i_1 there are the following 9 pathways: $i_1 \to j_1 \to k_2$, $i_1 \to j_1 \to k_3$, $i_1 \to j_1 \to k_4$, $i_1 \to j_2 \to k_3$, $i_1 \to j_2 \to k_4$, $i_1 \to j_2 \to k_5$, $i_1 \to j_3 \to k_1$, $i_1 \to j_3 \to k_4$, $i_1 \to j_3 \to k_5$.

The principal contributions of our paper are:

1. Proposing a new technique of zero-shot pruning neural networks without using any data. Previous pruning at initialization algorithms like SynFlow (Tanaka et al., 2020) are iterative and not zero-shot.
2. We present a deterministic Ramanujan graph construction technique for initializing sparse neural networks. To the best of our knowledge, no other work exists towards this direction.
3. Establishing that training sparse networks directly without previous pruning can work if the sparsification is done via the use of deterministic Ramanujan graphs. Previous research have indicated that vanilla training of sparse random networks are often unsuccessful to identify winning lottery tickets (Zhou et al., 2019).
4. In all previous works, for the identification of winning lottery ticket, sufficient to reach good generalization, is typically determined in an iterative fashion. However, zero-shot identification is more attractive (Tartaglione, 2022), which we develop in this work.
5. The construction technique is adapted for both fully connected and convolution layers.

Further, identifying the sparse existant pathways and their trained weights can help in better explainability and enables training with reduced computational effort.

3 Properties of the Constructed Networks

Previous approaches based on random network initialization and existent pruning strategies suffer from the issue of irregularity and are not guaranteed to be rigorously Ramanujan. For instance, application of the work of Hoory (Hoory et al., 2006) as mentioned in (Pal et al., 2022; Hoang et al., 2023) etc depends on the crucial fact that the minimal degrees of the base bipartite graphs needs to be ≥ 2 for the graphs to be Ramanujan. Our architecture based on deterministic regular Ramanujan graphs of degree ≥ 3 ensures that the initialized networks remain Ramanujan, are path-connected and are highly symmetric being either Cayley graphs of certain algebraic groups to replace the balanced dense bipartite graphs or the Ramanujan r-covering of full bi-regular bipartite graphs to replace the unbalanced dense bipartite graphs.

Path-connectedness: The fact that each layer of the bipartite graphs are either regular or bi-regular with the regularity bigger than 3 ensures that the entire architecture remains path-connected, i.e., starting from any node in the first layer we can reach a node in the last layer by a connected path. A proof of this is direct. In

Figure 1, suppose there are 3 layers I, J, K. We wish to reach layer K starting from any point in layer I by a connected path. Pick any $i_r, r \in \{1, 2, 3, 4, 5\}$. Use the fact that there is at least one edge going out from i_r to reach some j_s and from j_s again use the fact of outgoing edges bigger than 1 to reach a point in layer K. The general case follows by induction on the number of layers.

High-symmetricity: The adjacency matrices of Cayley graphs and that of covers of Cayley graphs have much more symmetry than that of general regular graphs. Often computations are optimised to use such symmetry viz. in the case of the software GAP (Group, 2022) for instance. In the future we wish to explore this direction of using the underlying symmetry to obtain fast computations on these sparse expander networks. This is also one of the reasons why we prefer deterministic constructions over random ones as the latter loses symmetry. In the graph of Figure 1, the adjacency matrix of the first layer (which can be represented as a Cayley graph on the group $\mathbf{Z}_2 \times \mathbf{Z}_5 = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 4\}$ with generating set $S = \{(1, 0), (1, 1), (1, 4)\}$) is $Adj = \begin{pmatrix} 0_{5\times5} & B_{5\times5} \\ B^T_{5\times5} & 0_{5\times5} \end{pmatrix}$ where $B = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$.

4 Formulation of Sparse Neural Ramanujan Graphs

In this section we present the mathematical framework which allows us to construct in a deterministic manner the sparse sub-network of the original dense neural networks. This forms the basis of our strategy of pruning at initialization. Recall that a Ramanujan graph is an extremal expander graph in the sense that its spectral gap is almost as large as possible. Here, we shall be concerned with bipartite Ramanujan graphs. Recall that a bi-partite graph is said to be balanced if the number of vertices in each of the partitions are the same and it is said to be unbalanced otherwise.

Definition 4.1 (Bipartite Ramanujan graphs)**.** Let $\Gamma = (V, E)$ be a d-regular ($d \geq 3$) balanced bipartite graph. Let the eigenvalues of its adjacency matrix be $\lambda_n \leq \lambda_{n-1} \leq \ldots \leq \lambda_2 \leq \lambda_1$. Then Γ is said to be Ramanujan iff $|\lambda_i| \leq 2\sqrt{d-1}$, for $i = 2, \ldots, (n-1)$.

For an unbalanced $(d_1, d_2)-$biregular bipartite graph ($d_1, d_2 \geq 3$), the condition of being Ramanujan changes to $|\lambda_i| \leq \sqrt{d_1 - 1} + \sqrt{d_2 - 1}$, for $i = 2, \ldots, (n-1)$. We see that when $d_1 = d_2$, it transforms to the usual definition. A representation of an unbalanced bi-regular bi-partite Ramanujan network, see Figure 2. Note that we are considering undirected graphs, so the adjacency matrix is a $0-1$ symmetric matrix and the eigenvalues are all real. A bi-partite graph has adjacency eigenvalues symmetric around 0. A detailed description of Ramanujan graphs can be found in (Hoory et al., 2006, sec. 5.3).

Ramanujan graphs are excellent spectral expanders. They are also extremely difficult to construct. In fact, even the question of existence of (infinite families of) Ramanujan graphs is a non-trivial one and it is not yet fully resolved for the non-bipartite case. For the bi-partite case it has been resolved by the recent works of Markus–Spielmann–Srivastava (Marcus et al., 2015; 2018) and Gribinski–Markus (Gribinski & Marcus, 2021). The first such construction of graphs are due to Lubotzky–Phillips–Sarnak (LPS) (Lubotzky et al., 1988) (and independently by Margulis (Margulis, 1988)). We shall modelise our pruned network according to these constructions.

4.1 Theoretical framework

Let us begin this section by recalling the Cheeger constant and the Cheeger inequality which will help to shed light on the fact that why good spectral expanding networks are closely related with pruned networks. In the following, a graph $\Gamma = (V, E)$ is a tuple consisting of a vertex set V and an edge set E which is a subset of $V \times V$.

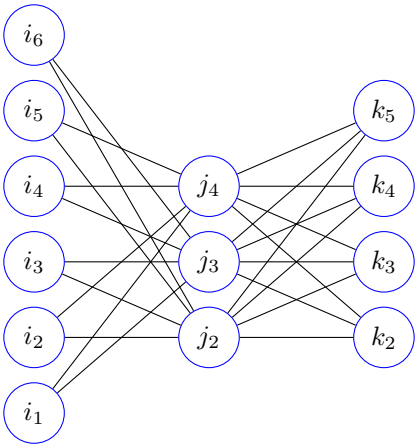


Figure 2: An example of a double layered bi-regular graph where each unbalanced bi-partite layer is Ramanujan. This can be checked from computing the adjacency eigenvalues of each and comparing with the Ramanujan bound for bi-regular bipartite graphs. The bi-regularity of the first layer is $(2, 4)$ and that of the second is $(4, 3)$. Note that if we have an unbalanced bi-regular, bi-partite network with (n_1, n_2) vertices and bi-regularity (d_1, d_2) then they must be related by the following equation: $n_1 d_1 = n_2 d_2$.

4.1.1 Combinatorial Expansion

Definition 4.2 (Expander and Cheeger constant)**.** A graph $\Gamma = (V, E)$ is an ϵ-vertex expander if for every non-empty subset $X \subset V$ with $|X| \leq \frac{|V|}{2}$, we have $\frac{|\delta(X)|}{|X|} \geq \epsilon$, where $\delta(X)$ denotes the outer vertex boundary of X i.e., the set of vertices in Γ which are connected to a vertex in X but do not lie in X. As X runs over all subsets of V, the infimum of $\frac{|\delta(X)|}{|X|}$ satisfying the conditions above is known as the vertex Cheeger constant and is denoted by $\mathfrak{h}(\Gamma)$.

Similar to the above, when we consider the edge boundary i.e., the set of edges which have one vertex in X and the other outside of X, we obtain the edge Cheeger constant $\mathbf{h}(\Gamma)$. The vertex Cheeger constant $\mathfrak{h}(\Gamma)$ and the edge Cheeger constant $\mathbf{h}(\Gamma)$ are related by the following equivalence $\frac{\mathfrak{h}(\Gamma)}{D} \leqslant \mathbf{h}(\Gamma) \leqslant \mathfrak{h}(\Gamma)$, where D denotes the maximum degree of the graph. The equivalence allows us to speak about vertex expansion and edge expansion interchangeably. Intuitively, given a graph with high vertex (or edge) Cheeger constant, it is more difficult to separate any subset of the vertices from the rest of the graph. This allows for free flow of information throughout the network which the graph modelises. In the literature, having a high Cheeger constant is also known as having high combinatorial expansion. However, computing the Cheeger constants of graphs is in general an NP-hard problem. To overcome this, we need the notion of spectral expansion.

4.1.2 Spectral Expansion

Spectral expansion is a bit different from combinatorial expansion. Given a finite undirected graph Γ the eigenvalues $\lambda_n \leqslant \cdots \leqslant \lambda_1$ of its adjacency matrix are all real and $\lambda_1 \leqslant D$ with equality iff the graph is D-regular. (a graph is said to be d-regular if there are exactly d-edges attached to a vertex). Thus, a d-regular bipartite graph is a graph which has the same number of vertices in each partition and every vertex of each partition has exactly d edges attached to it. A graph $\Gamma = (V, E)$ is said to be a spectral expander if the quantities $\{|\lambda_1| - |\lambda_2|, |\lambda_1| - |\lambda_k|\}$ are both bounded away from zero, where $k = n - 1$ if the graph is bipartite and $k = n$ otherwise.

Question 4.3. *Why is there a need to establish strong spectral expansion?*

The answer to this lies in the fact that spectral expansion implies combinatorial expansion via the discrete Cheeger-Buser inequality (see appendix) and the bigger the spectral expansion, the more combinatorially expanding the base network graphs are. In general combinatorial expansion (counting the values of the vertex

or the edge cheeger constants) is an NP-hard problem. Now the natural question arises that whether there is a limit to the spectral expansion or can it become as large as possible. This leads us to Ramanujan graphs which are the optimal spectral expanders. See appendix for the Alon-Bopanna theorem.

For the construction of the deterministic Ramanujan networks, we shall need the following notions from arithmetic.

Definition 4.4 (Quadratic residue and Legendre symbol)**.** An integer q is called a quadratic residue modulo n if there exists an integer x such that $x^2 \equiv q \pmod{n}$. Otherwise, q is called a quadratic non-residue modulo n.
Let p be an odd prime number and a be an integer. The Legendre symbol of a and p is defined as

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ & \text{and } a \not\equiv 0 \pmod{p}, \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p, \\ 0 & \text{if } a \equiv 0 \pmod{p}. \end{cases}$$

Given a prime a, there are infinitely many primes p such that Legendre symbol of a and p is -1 (and also there are infinite many primes p such that it is $+1$)

Definition 4.5 ($PGL_2(K)$)**.** Let K be a field. Let us denote by $GL_2(K)$ the group of invertible 2-by-2 matrices with coefficients in K, ie, the matrices with non-zero determinant. Let $PGL_2(K)$ be the quotient group

$$PGL_2(K) = GL_2(K)/Z(K)$$

where

$$GL_2(K) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : ad - bc \neq 0 \, (in \ K) \right\}$$

and

$$Z(K) = \left\{ \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} : a \neq 0 (in \ K) \right\}$$

Remark: If $K = \mathbb{F}_q$, then a simple computation gives the size of $PGL_2(K)$ to be $|PGL_2(K)| = q(q^2 - 1)$. In Section 4.3, we shall use this property to construct bipartite $\frac{q(q^2-1)}{2}$ by $\frac{q(q^2-1)}{2}$ Ramanujan networks.

4.2 Regular Ramanujan graphs

Let $p, q \equiv 1 (mod\, 4)$ be distinct odd primes (the condition of $1(mod\, 4)$ can be removed at the cost of making the analysis more technical and complicated, we shall mention later how it is achieved). The graph $X^{p,q}$ is constructed using the following general method.

1. It is a Cayley graph (see Appendix A.1) on the subgroup of 2 by 2 matrices, $PGL_2(\mathbb{F}_q)$ where $\mathbb{F}_q$ is the finite field of characteristic q.
2. Consider the equation $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$. Jacobi's four square theorem states that there are $p + 1$ solutions to the equation $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$ with $a_0 > 0$ odd (i.e., $a_0 \equiv 1 \pmod 2$) and a_1, a_2, a_3 even. Now, for each such solution (a_0, a_1, a_2, a_3) consider the matrix $\begin{pmatrix} a_0 + ia_1 & a_2 + ia_3 \\ -a_2 + ia_3 & a_0 - ia_1 \end{pmatrix}$ where i is some fixed solution to $i^2 = -1 \pmod q$. This matrix belongs to $PGL_2(\mathbb{F}_q)$. This can be checked from the definition of $PGL_2(\mathbb{F}_q)$.
3. Form the generating set S of the Cayley graph to be the set of these $(p + 1)$ matrices. Thus $X^{p,q} = Cay(PGL_2(\mathbb{F}_q), S)$.
4. The graphs are bipartite iff p is not a quadratic residue modulo q or in other words the Legendre symbol $\left(\frac{q}{p}\right) = -1$. The bipartite graphs $X^{p,q}$ will be $(p + 1)$-regular, of size $\frac{q(q^2-1)}{2}$ by $\frac{q(q^2-1)}{2}$ and are Ramanujan (Lubotzky et al., 1988).

Remark: If $p \equiv 3 \pmod 4$, then a similar strategy is employed, except in this case one looks at solutions of $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$ with $a_0 \equiv 0 \pmod 2$. See (Musitelli & de la Harpe, 2006, sec. 2).

4.3 Construction of the fully connected layers

For the fully connected layers consisting of balanced bipartite graphs, we prune them at initialization in accordance with the Ramanujan graph structure of LPS. For this we select a prime q such that $\frac{q(q^2-1)}{2}$ by $\frac{q(q^2-1)}{2}$ is closest to the size of the original bipartite layer. We then select the prime p such that the Legendre symbol $\left(\frac{q}{p}\right) = -1$ (note that this choice is always possible as given a prime q there are infinite number of primes p satisfying this property). Selecting the minimum possible value of p will give us the sparsest Ramanujan graph. For a 4096 by 4096 original network, our choice of $(p, q) = (5, 17)$ giving rise to 6 regular bipartite sparse Ramanujan networks. Note that here we have taken $p \equiv 1 \pmod 4$, but we could have also chosen $p = 3$ or even $p = 2$ (see construction of cubic Ramanujan graphs (Chiu, 1992)) resulting in even sparser networks.

4.4 Bi-regular Ramanujan graphs

A bipartite graph is said to be (d_1, d_2) bi-regular if each bi-partition has fixed regularity d_1, d_2 respectively. Note that a simple computation reveals that if (n_1, n_2) are the bi-partition sizes, then $n_1 d_1 = n_2 d_2$. Thus three parameters are needed to specify these types of graphs. One way to construct bi-regular Ramanujan graphs is the following, see (Burnwal et al., 2021):
Fix a prime q and a $q \times q$ cyclic shift permutation matrix $P = [P]_{ij}$ with $[P]_{ij} = 1$ if $j = i - 1 \pmod q$ and 0 otherwise. Recall that the adjacency matrix of any $m \times n$ bipartite graph can be written as $Adj = \begin{pmatrix} 0_{m\times m} & B_{m\times n} \\ B^T_{m\times n} & 0_{n\times n} \end{pmatrix}$, where B is called the bi-adjacency matrix. Define the bi-adjacency matrix of this bipartite graph to be $B = \begin{pmatrix} I_q & I_q & \dots & I_q \\ I_q & P & \dots & P^{l-1} \\ I_q & P^2 & \dots & P^{2(l-1)} \\ \vdots & & & \\ I_q & P^{q-1} & \dots & P^{(q-1)(l-1)} \end{pmatrix}$ where I_q is the $q \times q$ identity matrix and P is as above. B is a $q^2 \times lq$ matrix and the bipartite graph is either $q^2 \times lq$ with bi-regularity (l, q) or symmetrically $lq \times q^2$ with bi-regularity (q, l). The graphs whose bi-adjacency matrices are represented as B (or B^T) are Ramanujan. These graphs are explicit realisations of the Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph on $k + l$ vertices as shown in (Hall et al., 2018, cor 2.2).

4.5 Construction of the convolution layers

For pruning the convolution layers, we utilise the bi-regular Ramanujan graphs. Let $q \geq l$. We analyse the size of the pruned network compared to the original fully connected network. The total number of edges in the $q^2 \times lq$ Ramanujan graph is lq^2 whereas the original network has lq^3 edges. Choosing the value of q to be as large as possible ensures that the pruned network has a small percentage of edges remaining while still being a Ramanujan network.

The condition $q \geq l$ is not a necessary requirement for implementing the technique outlined in Section 4.3. The reason behind this flexibility lies in the specific properties of the unbalanced bipartite graph B, which has dimensions q^2 by lq. The critical insight is that if the unbalanced bipartite graph B is Ramanujan, then its transpose, denoted as B^T (with dimensions lq by q^2), is also Ramanujan.

4.6 Time Complexity of Construction

A m by n bipartite graph connects two network layers with m and n nodes respectively. The technique for convolution layers to generate bi-regular bipartite graphs (of order m by n) has complexity $O(mn)$. This complexity is due to the creation of the pruning mask which is of size $m \times n$. The LPS technique to generate

$p+1$ regular Ramanujan graphs has complexity $O(q^5+p^4) = O(q^5) = O(m^{5/3})$. Here $m = n = O(q^3)$. This complexity is due to the creation of the PGL_2 group in which first we need to create the generator matrix and then find the equivalence classes which takes time $O(q^4q)$. The solution to the four square problem has complexity $O(p^4)$. Since the number of nodes are much less compared to the total parameters, the complexity is low.

4.7 General bipartite networks

In the case of bipartite networks with arbitrary sizes, one can achieve as sparse Ramanujan graphs as possible. It has recently been proven by Marcus, Spielman and Srivastava that for the regular case, for each degree $d \geq 3$, infinite families of bipartite Ramanujan graphs exist. This is also true for the bi-regular case, for each pair (d_1, d_2) with $d_1, d_2 \geq 3, d_2 = kd_1, k \geq 2$. Further they showed the existence of these types of graphs of all sizes. Their method of proof is probabilistic and existential in nature. It does not give explicit families of bipartite Ramanujan graphs. However there now exist polynomial time algorithms (Cohen, 2016) (for the regular case) (Gribinski & Marcus, 2021) (for the bi-regular case) with which we can extract explicit Ramanujan graphs. For the regular case, we fix an integer $n \geq 3$ and a degree $d \geq 3$ and in the output we shall obtain a d-regular $n \times n$ bipartite Ramanujan graph while for the bi-regular case, we fix three integers n, k, d with $n > 2, d > 2, k \geq 2$ and obtain (d, kd) bi-regular Ramanujan graph of size $kn \times n$.

5 Experimental Methodology and Results

The goal of our experiments is to study the effectiveness of deterministic Ramanujan graph based sparse network initialization.

5.1 Datasets and architectures

The datasets used for the experiments are Cifar-10 and Cifar-100 (Krizhevsky, 2009). The experiments are performed over a variety of architectures including VGG13, VGG16, VGG19 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), ResNet18 and ResNet34 (He et al., 2016) to show the robustness of our method. We proceed in two parts. In the first part, we prune the intermediate Fully Connected layers by replacing them with sparse Ramanujan Graph which is applicable for VGG13, VGG19 and AlexNet architectures. In the second part, we prune the whole network including the Convolution layers and the Fully Connected layers which is applicable for all the architectures considered in our experiment. The performance of the dense and the pruned networks are compared in each case. Finally, we compare the performance of our method against various state-of-the art PaI algorithms for VGG16 and the ResNet34 architectures. Training parameters for all of the architectures are same and are summarized in Table 1. We report accuracy on a randomly split 16% test set for all the experiments.

Table 1: Training Parameters for the experiment

Hyperparmeters	
Epochs	200
Train Batch Size	256
Test Batch Size	128
Learning Rate	0.1
LR Decay, Epoch	10x, [100, 150]
Optimizer	SGD
Weight Decay	0.0005
Momentum	0.9
Weight Initialization	Kaiming Uniform

5.2 Methods compared

The performance of the pruned networks are compared with that of corresponding dense networks. We have also compared our method against various pruning at initialization techniques such as Random (Liu et al., 2022), ERK (Evci et al., 2020; Mocanu et al., 2018), GraSP (Wang et al., 2020), and SynFlow (Tanaka et al., 2020). The number of iterations used for SynFlow are 100 and for ERK and GraSP it is 1 keeping the rest of the hyperparameters same as in Table 1.

5.3 Network construction parameters

Experiments are conducted in two parts: 1) Pruning the Fully Connected layers, 2) Pruning the whole network including the Convolution and the Fully Connected layers.

For the first part, the construction of Ramanujan graphs as given in Section 4.3 is used. We have used a dedicated fully connected layer of size 4096×4096 for VGG13, VGG19 and AlexNet architectures and the values used for p and q (Section 4.2) are given in Table 2. This results in the fully connected layer becoming of size $q \times (q^2-1)/2$ with the effective number of connections between the layer being equal to $q \times (q^2-1)/2 \times (p+1)$.

Table 2: Values of p and q used to generate the sparse fully connected layer

Model	VGG13, VGG19, AlexNet
FC Layer Size	4096×4096
p	29, 109
q	17

The convolution layers are pruned according to the construction given in Section 4.5. The convolution layer can be thought of as a matrix of dimensions $|N_{out}| \times |N_{in}| \times |K_w| \times |K_h|$ where $|N_{out}|$ is the number of output channels, $|N_{in}|$ is the number of input channels, $|K_w|$ is the kernel width and $|K_h|$ is the kernel height. This is considered to be a bipartite graph with $|V_{left}| = |N_{in}| \times |K_w| \times |K_h|$ and $V_{right} = |N_{out}|$ where each vertex of V_{left} has an edge with each vertex of V_{right}. The size of convolution layer being pruned and the choice of l and q for VGG16 and ResNet34 architectures is given in Table 3 while for the rest of the architectures is given in Table 7.

Table 3: Values of q and l to generate Ramanujan Graphs for layers of VGG16 and ResNet34

VGG16			ResNet34		
Convolution Layer Size	q	l	Convolution Layer Size	q	l
$[128 \times 64 \times 3 \times 3] \times 1$	11	52	$[64 \times 64 \times 3 \times 3] \times 6$	7	82
$[128 \times 128 \times 3 \times 3] \times 1$	11	104	$[128 \times 64 \times 3 \times 3] \times 1$	11	52
$[256 \times 128 \times 3 \times 3] \times 1$	13	88	$[128 \times 128 \times 3 \times 3] \times 7$	11	104
$[256 \times 256 \times 3\times] \times 2$	13	177	$[256 \times 128 \times 3 \times 3] \times 1$	13	88
$[512 \times 256 \times 3 \times 3] \times 1$	19	121	$[256 \times 256 \times 3 \times 3] \times 11$	13	177
$[512 \times 512 \times 3 \times 3] \times 5$	19	242	$[512 \times 256 \times 3 \times 3] \times 1$	19	121
			$[512 \times 512 \times 3\times] \times 5$	19	242

The pruning mask thus obtained is a matrix of size $q^2 \times lq$ with the effective connections being equal to $q^2 \times l$. The original pruning mask of the convolution layer has size $|N_{out}| \times [|N_{in}| \times |K_w| \times |K_h|]$. By construction the obtained Ramanujan graph is actually a subgraph of the original pruning mask and thus the entries in the original mask not part of the constructed Ramanujan graph are set to 0.

The network density (remaining edge percentage) reported in Section 5.4 is calculated by dividing the number of effective connections which is equal to the sum of $q \times (q^2-1)/2 \times (p+1)$ (effective number of connections in fully connected layer) and $q^2 \times l$ (effective number of connections in each of the convolution layers) divided by the total number of connections present in the unpruned network.

5.4 Results and discussion

We study the accuracy of sparse networks obtained by our technique for various architectures and datasets. The accuracy is compared with that of the corresponding unpruned network with similar number of nodes. Results for the first part of the experiment where only the intermediate fully connected layer is pruned, are summarized in Table 4. It can be observed that the Ramanujan graph construction allows us to extremely prune the fully connected layer upto **0.43%** while still retaining the accuracy as of the unpruned model.

Table 4: Accuracy of VGG and AlexNet when only the intermediate fully connected layer is pruned

Dataset: Cifar-10			
Model	FC layer Size (Remaining Edge Percentage)		
	4096×4096 (Unpruned)	$\mathbf{2448 \times 110}$ (**1.6**%)	$\mathbf{2448 \times 30}$ (**0.43**%)
VGG13	92%	**91%**	**91%**
VGG19	92%	**92%**	**92%**
AlexNet	86%	**84%**	**86%**

Dataset: Cifar-100			
Model	FC layer Size (Remaining Edge Percentage)		
	4096×4096 (Unpruned)	$\mathbf{2448 \times 110}$ (**1.6**%)	$\mathbf{2448 \times 30}$ (**0.43**%)
VGG13	66%	**66%**	**63%**
VGG19	66%	**67%**	**63%**
AlexNet	67%	**66%**	**66%**

For the second part of the experiment where we prune the complete network including the convolution layers and the fully connected layer, we could achieve an overall pruning percentage of ∼2% to ∼5% for VGG, ∼2.3% for AlexNet and ∼5% for the ResNet architectures. The accuracy of the models on the Cifar-10 and Cifar-100 datasets are summarized in Table 5. A small accuracy drop is observed as compared to the dense network.

Table 5: Accuracy of various architectures when the complete network is pruned including the Convolution and the FC layers

Dataset: Cifar-10			
Model	Unpruned accuracy	Pruned Accuracy	Network Density
VGG13	92%	90%	1.7%
VGG16	93%	91%	5.3%
VGG19	92%	89%	2.4%
AlexNet	86%	82%	2.3%
ResNet18	87%	86%	5.6%
ResNet34	88%	86%	5.2%

Dataset: Cifar-100			
Model	Unpruned accuracy	Pruned Accuracy	Network Density
VGG16	70%	66%	5.3%
ResNet18	55%	54%	5.6%
ResNet34	57%	56%	5.2%

Dataset: Tiny-ImageNet			
Model	Unpruned accuracy	Pruned Accuracy	Network Density
VGG16	43%	40%	5.3%
ResNet34	56%	48%	5.2%

Finally, we compare the performance of the proposed Ramanujan sparse network initialization with other state-of-art pruning at initialization (PaI) techniques. The comparison of accuracy between various pruning at initialization (PaI) techniques at network density ∼5% is shown for the VGG16 and ResNet34 architectures in Table 6.

We can observe that our zero-shot method can achieve comparable accuracy to other iterative pruning at initialization techniques. It also significantly outperforms the random mask initialization. The pruned networks still maintain their accuracy with a slight reduction compared to their unpruned counterparts even at such low remaining weight percentage.

6 Conclusion and Future Work

We presented a deterministic, data independent, zero-shot method for constructing sparse neural network structures which upon weight initialization can be trained to a high accuracy. The method is based on a

Table 6: Comparison of our method against other state-of-the art PaI methods

Dataset: Cifar-10	
VGG16 (Network Density ∼5.3%)	
Method	Accuracy
Unpruned	93%
Our Method	91%
Random	89%
ERK	91%
SynFlow	**92%**
ResNet34 (Network Density ∼5.2%)	
Method	Accuracy
Unpruned	88%
Our Method	**86%**
Random	81%
ERK	**86%**
GraSP	**86%**

Dataset: Cifar-100	
VGG16 (Network Density ∼5.3%)	
Method	Accuracy
Unpruned	70%
Our Method	**66%**
Random	60%
ERK	62%
SynFlow	65%
ResNet34 (Network Density ∼5.2%)	
Method	Accuracy
Unpruned	57%
Our Method	**56%**
Random	50%
ERK	**56%**
GraSP	**56%**

Dataset: Tiny-ImageNet	
VGG16 (Network Density ∼5.3%)	
Method	Accuracy
Unpruned	43%
Our Method	**40%**
Random	35%
ERK	39%
SynFlow	**40%**
ResNet34 (Network Density ∼5.2%)	
Method	Accuracy
Unpruned	56%
Our Method	**48%**
Random	45%
ERK	47%
GraSP	29%

Ramanujan graph construction technique using Cayley graphs and Ramanujan coverings. Unlike random graph generation, this always results in a structured, symmetric, and regular sparse network. The method is adapted for masking both the fully connected and convolution layers. Experimental results on popular architectures and datasets demonstrate that close to unpruned network accuracy can be achieved using a very sparse network structure.

With the success of sparse deterministic Ramanujan neural networks, our further direction of work is to implement these in the case of transformers and study sparse Ramanujan transformer networks. The proposed deterministic construction technique is expected to significantly reduce the number of parameters and training time while maintaining accuracy.

References

Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, Jun 1986. ISSN 1439-6912. doi: 10.1007/BF02579166. URL https://doi.org/10.1007/BF02579166.

Noga Alon and Vitali D Milman. $\lambda 1$, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

Suryam Arnav Kalra, Arindam Biswas, Pabitra Mitra, and Biswajit Basu. Graph expansion in pruned recurrent neural network layers preserve performance. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024,*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=hG5eu7ikDy.

Goulnara Arzhantseva and Arindam Biswas. Logarithmic girth expander graphs of $SL_n(F_p)$. *Journal of Algebraic Combinatorics*, 56(3):691–723, Nov 2022. ISSN 1572-9192. doi: 10.1007/s10801-022-01128-z. URL https://doi.org/10.1007/s10801-022-01128-z.

Arindam Biswas. On a cheeger type inequality in cayley graphs of finite groups. *European Journal of Combinatorics*, 81:298–308, October 2019. doi: 10.1016/j.ejc.2019.06.009. URL https://doi.org/10.1016/j.ejc.2019.06.009.

Arindam Biswas and Jyoti Prakash Saha. A Cheeger type inequality in finite Cayley sum graphs. *Algebraic Combinatorics*, 4(3):517–531, 2021. doi: 10.5802/alco.166. URL https://alco.centre-mersenne.org/articles/10.5802/alco.166/.

Arindam Biswas and Jyoti Prakash Saha. A spectral bound for vertex-transitive graphs and their spanning subgraphs. *Algebraic Combinatorics*, 6(3):689–706, 2023. doi: 10.5802/alco.278. URL https://alco.centre-mersenne.org/articles/10.5802/alco.278/.

Emmanuel Breuillard, Ben Green, Robert Guralnick, and Terence Tao. Expansion in finite simple groups of lie type. *Journal of the European Mathematical Society*, 17(6):1367–1434, 2015. doi: 10.4171/jems/533. URL https://doi.org/10.4171/jems/533.

Shantanu Prasad Burnwal, Kaneenika Sinha, and Mathukumalli Vidyasagar. New and explicit constructions of unbalanced ramanujan bipartite graphs. *The Ramanujan Journal*, 57(3):1043–1069, April 2021. URL https://doi.org/10.1007/s11139-021-00384-0.

Tianlong Chen, Xuxi Chen, Xiaolong Ma, Yanzhi Wang, and Zhangyang Wang. Coarsening the granularity: Towards structurally sparse lottery tickets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3025–3039. PMLR, 17–23 Jul 2022.

Zhuangzhi Chen, Jingyang Xiang, Yao Lu, Qi Xuan, Zhen Wang, Guanrong Chen, and Xiaoniu Yang. Rgp: Neural network pruning through regular graph with edges swapping. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023. doi: 10.1109/TNNLS.2023.3280899.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations. *arXiv preprint arXiv:2308.06767*, 2023.

Patrick Chiu. Cubic ramanujan graphs. *Combinatorica*, 12:275–285, 1992.

Fan Chung. A generalized alon-boppana bound and weak ramanujan graphs. *The Electronic Journal of Combinatorics*, 23(3), July 2016. doi: 10.37236/5933. URL https://doi.org/10.37236/5933.

Michael B Cohen. Ramanujan graphs in polynomial time. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 276–281. IEEE, 2016.

Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284(2):787–794, 1984.

Kiara Esguerra, Muneeb Nasir, Tong Boon Tang, Afidalina Tumian, and Eric Tatt Wei Ho. Sparsity-aware orthogonal initialization of deep neural networks. *IEEE Access*, 2023.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, , and Erich Elsen. Rigging the lottery: Making all tickets winners. In *ArXiV*, 2020.

Jonas Fischer and Rebekka Burkholz. Plant'n'seek: Can you find the winning ticket? In *International Conference on Learning Representations*, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2020.

Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *Proceedings of International Conference on Machine Learning*, 2023.

Aurelien Gribinski and Adam W. Marcus. Existence and polynomial time construction of biregular, bipartite ramanujan graphs of all degrees, 2021.

The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.12.2*, 2022. URL https://www.gap-system.org.

Chris Hall, Doron Puder, and William F. Sawin. Ramanujan coverings of graphs. *Advances in Mathematics*, 323:367–410, 2018. ISSN 0001-8708. doi: https://doi.org/10.1016/j.aim.2017.10.042. URL https://www.sciencedirect.com/science/article/pii/S0001870817303146.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 2016.

Duc Hoang, Shiwei Liu, Radu Marculescu, and Zhangyang Wang. Revisiting pruning at initialization through the lens of ramanujan graph. In *International Conference on Learning Representations*, 2023.

Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

J. Kepner and R. Robinett. Radix-net: Structured sparse matrices for deep neural networks. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 268–274, Los Alamitos, CA, USA, may 2019. IEEE Computer Society. doi: 10.1109/IPDPSW.2019.00051. URL https://doi.ieeecomputersociety.org/10.1109/IPDPSW.2019.00051.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *NEURIPS*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NEURIPS*, 2012.

Steinar Laenen. One-shot neural network pruning via spectral graph sparsification. In *TAGML Workshop, ICML*, 2023.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. In *International Conference on Learning Representations*, 2019a.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations,*, 2019b.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, , and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *International Conference on Learning Representations*, 2022.

A Lubotzky, R Phillips, and P Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, and Yanzhi Wang. Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? In *Advances in Neural Information Processing Systems*, volume 34, pp. 12749–12760, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/6a130f1dc6f0c829f874e92e5458dced-Paper.pdf.

Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. *Annals. Math.*, 182:307 – 325, 2015.

Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families iv: Bipartite ramanujan graphs of all sizes. *SIAM Journal on Computing*, 47(6):2488–2509, 2018. doi: 10.1137/16M106176X. URL https://doi.org/10.1137/16M106176X.

G. A. Margulis. Explicit constructions of graphs without short cycles and low density codes. *Combinatorica*, 2(1):71–78, Mar 1982. ISSN 1439-6912. doi: 10.1007/BF02579283. URL https://doi.org/10.1007/BF02579283.

Grigorii Aleksandrovich Margulis. Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. *Problemy peredachi informatsii*, 24(1):51–60, 1988.

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. In *Nature communications*, 2018.

M. Morgenstern. Existence and explicit constructions of q + 1 regular ramanujan graphs for every prime power q. *Journal of Combinatorial Theory, Series B*, 62(1):44–62, 1994. ISSN 0095-8956. doi: https://doi.org/10.1006/jctb.1994.1054. URL https://www.sciencedirect.com/science/article/pii/S0095895684710549.

Antoine Musitelli and Pierre de la Harpe. Expanding graphs, ramanujan graphs, and 1-factor perturbations. *Bulletin of the Belgian Mathematical Society-Simon Stevin*, 13(4):673–680, 2006.

A. Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991. ISSN 0012-365X. doi: https://doi.org/10.1016/0012-365X(91)90112-F. URL https://www.sciencedirect.com/science/article/pii/0012365X9190112F.

Bithika Pal, Arindam Biswas, Sudeshna Kolay, Pabitra Mitra, and Biswajit Basu. A study on the ramanujan graph property of winning lottery tickets. In *International Conference on Machine Learning*, pp. 17186–17201, 2022.

Ameya Prabhu, Girish Varma, and Anoop Namboodiri. Deep expander networks: Efficient deep networks from graph theory. In *Computer Vision – ECCV 2018*, pp. 20–36. Springer International Publishing, 2018a. doi: 10.1007/978-3-030-01261-8_2. URL https://doi.org/10.1007%2F978-3-030-01261-8_2.

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, 2018b.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gSj0NKvB.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014.

Kartik Sreenivasan, Jy-yong Sohn, Liu Yang, Matthew Grinde, Alliot Nagle, Hongyi Wang, Eric Xing, Kangwook Lee, and Dimitris Papailiopoulos. Rare gems: Finding lottery tickets at initialization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14529–14540, 2022.

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.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Enzo Tartaglione. The rise of the lottery heroes: Why zero-shot pruning is hard. In *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, October 2022. doi: 10.1109/icip46576.2022.9897223. URL http://dx.doi.org/10.1109/ICIP46576.2022.9897223.

Artem Vysogorets and Julia Kempe. Connectivity matters: Neural network pruning through the lens of effective sparsity. *J. Mach. Learn. Res.*, 24, 2023.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations,*, 2020.

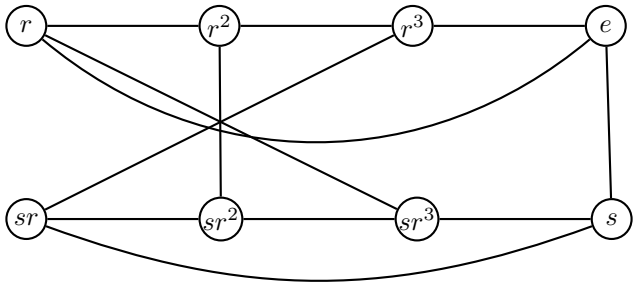
Huan Wang, Can Qin, Yue Bai, Yulun Zhang, and Yun Fu. Recent advances on neural network pruning at initialization. *arXiv preprint arXiv:2103.06460*, 2021.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: zeros, signs, and the supermask. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

A Appendix

A.1 Cayley graph

Let G be a group and let S be a subset of G that is closed under inversion i.e., $S = S^{-1}$. The corresponding Cayley graph $C(G, S)$ is a graph with vertex set the elements of G and edge set $\{(x, xs) : x \in G,\ s \in S\}$. As an example of a Cayley graph of a non-abelian group, one can take the group $G = D_4$, the dihedral group of order 8 with elements $\{e, r, r^2, r^3, s, sr, sr^2, sr^3\}$ and generating set $S = \{r, s, r^{-1}\}$. Here the r denotes rotation by $\frac{\pi}{2}$ and s is reflection. So $r^4 = 1$, $s^2 = 1$ and $sr = r^{-1}s$.



The Cayley Graph $C(D_4, \{s, r, r^3\})$

A.2 Expander graphs and Ramanujan graphs

An expander graph is a structurally sparse graph that has strong connectivity properties. The connectivity can be quantified in different ways which give rise to different notions of expanders such as vertex expanders, edge expanders and spectral expanders. These notions are actually interrelated. In the following, a graph $\Gamma = (V, E)$ is a tuple consisting of a vertex set V and an edge set E which is a subset of $V \times V$.

A.2.1 Combinatorial expansion

Definition A.1 (vertex Cheeger constant)**.** The infimum of the quantity $\frac{|\delta(X)|}{|X|}$ where $\delta(X)$ denotes the outer vertex boundary of X i.e., the set of vertices in Γ which are connected to a vertex in X but do not lie in X as X runs over all non-empty subsets of V satisfying the condition with $|X| \leq \frac{|V|}{2}$ is known as the vertex Cheeger constant and is denoted by $\mathfrak{h}(\Gamma)$.

Definition A.2 (edge-Cheeger constant)**.** The edge boundary of a set S, denoted δS, is $\delta S =$ the set of edges going out from S to its complement. The edge Cheeger constant of Γ, denoted by $\mathbf{h}(\Gamma)$, is defined as: $\mathbf{h}(\Gamma) = \min \frac{|\delta S|}{D|S|}$ as S satisfies the following: $\{S \neq \text{empty set}, |S| \leqslant \frac{n}{2}\}$ and D is the maximum degree of the graph Γ

The vertex Cheeger constant $\mathfrak{h}(\Gamma)$ and the edge Cheeger constant $\mathbf{h}(\Gamma)$ are related by the following equivalence

$$\frac{\mathfrak{h}(\Gamma)}{D} \leqslant \mathbf{h}(\Gamma) \leqslant \mathfrak{h}(\Gamma),$$

where D denotes the maximum degree of the graph (the degree of each vertex is the number of edges going out from the vertex). This allows one to speak about vertex expansion and edge expansion interchangeably. Having high combinatorial expansion means having high Cheeger constant, a desirable property for our case.

A.2.2 Spectral expansion

Given a finite undirected graph Γ the eigenvalues $\lambda_n \leqslant \cdots \leqslant \lambda_1$ of its adjacency matrix are all real and $\lambda_1 \leqslant D$ with equality iff the graph is D-regular. The spectra, i.e., the distribution of the eigenvalues convey

a lot of information about the structure of the graphs. For instance, the quantity $\lambda_1 - \lambda_2$ (also known in the literature as the one sided spectral gap) quantifies the connectivity and the combinatorial expansion of the graph via the discrete Cheeger-Buser inequality, discovered independently by Dodziuk (1984) and by Alon & Milman (1985). A graph $\Gamma = (V, E)$ is said to be a spectral expander if the quantities $\{|\lambda_1| - |\lambda_2|, |\lambda_1| - |\lambda_k|\}$ are both bounded away from zero, where $k = n - 1$ if the graph is bipartite and $k = n$ otherwise.

A.2.3 Discrete Cheeger–Buser inequality

The discrete Cheeger–Buser inequality discovered independently by (Dodziuk, 1984) and by (Alon & Milman, 1985) allows one to pass from spectral expansion to combinatorial expansion. The inequality states that

$$\frac{\mathbf{h}(\Gamma)^2}{2} \leqslant \alpha_2 \leqslant 2\mathbf{h}(\Gamma),$$

where α_2 denotes the second smallest eigenvalue of the normalised Laplacian matrix of Γ and is related to the eigenvalues of the adjacency matrix via

$$\frac{\lambda_i}{D} \leqslant 1 - \alpha_i \leqslant \frac{\lambda_i}{d} \ \forall i = 1, 2, \ldots, n.$$

See (Chung, 2016) for details. From the above, it is easy to check that a high $|\lambda_1| - |\lambda_2|$ ensures a high $\mathbf{h}(\Gamma)$ and vice-versa. Thus, the two notions of expansion are inter-connected and every spectral expander remains a combinatorial expander. They are actually equivalent for some classes of graphs, for instance bipartite graphs (as the adjacency spectrum is symmetric about the origin), variants of algebraic graphs e.g., see (Breuillard et al., 2015; Biswas, 2019; Biswas & Saha, 2021; 2023) etc.

A.2.4 Ramanujan graph bounds, Alon-Bopanna Theorem

A d-regular graph is said to be a Ramanujan graph if $\max\{|\lambda_2|, |\lambda_k|\} \leqslant 2\sqrt{d-1}$. In the case of bipartite graphs, $\lambda_n = \lambda_1$ and $\lambda_{n-1} = \lambda_2$, hence the previous expression reduces to $|\lambda_2| \leqslant 2\sqrt{d-1}$. For fixed degree, with the sizes of the graphs growing larger and larger, these are the best possible expanders, as given by the Alon-Bopanna bound (Alon, 1986; Nilli, 1991).

Theorem A.3 (Alon-Boppana)**.** *For every d regular graph on n vertices,*

$$\lambda \geq 2\sqrt{d-1} - o_n(1).$$

The $o_n(1)$ term is a quantity that tends to zero for every fixed d as $n \to \infty$.

The bound requires a bit of technical details. However, if one relaxes a bit the right hand side of the above bound then one can easily show the following,

$$\lambda \geq \sqrt{d} \cdot (1 - o_n(1)).$$

The proof goes as follows: Let A be the adjacency matrix of G, then $\text{trace}(A^k)$ is the number of all walks of length k in G that start and end in the same vertex. In particular, all the diagonal entries in A^2 are $\geq d$. Thus, $\text{trace}(A^2) \geq nd$. On the other hand,

$$\text{trace}(A^2) = \sum_i \lambda_i^2 \leq d^2 + (n-1)\lambda^2.$$

Thus, $(n-1)\lambda^2 \geq dn - d^2$, which implies that $\lambda^2 \geq d \cdot \frac{n-d}{n-1}$.
See (Hoory et al., 2006) for details.

A.2.5 Expanders and Ramanujan graphs from finite simple groups

The existence and construction of expanders are a deep question and that of Ramanujan graphs are even deeper. Most of the constructions of expanders are based on Cayley graphs of finite simple groups of Lie type. The first construction is due to Margulis (Margulis, 1982). Later Lubotzky–Phillips–Sarnak (Lubotzky et al.,

1988) constructed Ramanujan graphs from $SL_2(\mathbb{F}_p)$. Till 2014 these graphs and variants thereof by (Chiu, 1992) and (Morgenstern, 1994) were the only known construction of Ramanujan graphs. Recent works of Markus–Spielmann–Srivastava (Marcus et al., 2015) have shown the existence of bipartite Ramanujan graphs of all degrees and sizes. Recently, there has also been new research directions on the topic of construction of expanders satisfying other desirable properties such as the diameter-by-girth ratio are bounded, for instance (Arzhantseva & Biswas, 2022). It will be interesting to see if these special expander networks play important roles as architectures for neural networks or not.

A.3 Experimental methodology

The q and l values used by the Ramanujan Graph construction for the convolution layers as mentioned in Section 4.5 for various architectures is provided in Table 7.

Table 7: Values of q and l to generate Ramanujan graphs for layers of VGG, AlexNet and ResNet

VGG13			VGG19		
Conv Size	q	l	Conv Size	q	l
$[256 \times 256 \times 3 \times 3] \times 1$	13	177	$[256 \times 256 \times 3 \times 3] \times 3$	13	177
$[512 \times 256 \times 3 \times 3] \times 1$	19	121	$[512 \times 256 \times 3 \times 3] \times 1$	19	121
$[512 \times 512 \times 3 \times 3] \times 3$	19	242	$[512 \times 512 \times 3 \times 3] \times 7$	19	242
Conv to Linear Size	q	l	Conv to Linear Size	q	l
2448×25088	47	533	2448×25088	47	533
AlexNet			ResNet18		
Conv Size	q	l	Conv Size	q	l
$[384 \times 256 \times 3 \times 3] \times 1$	19	121	$[64 \times 64 \times 3 \times 3] \times 4$	7	82
$[384 \times 384 \times 3 \times 3] \times 1$	19	181	$[128 \times 64 \times 3 \times 3] \times 1$	11	52
$[256 \times 384 \times 3 \times 3] \times 1$	13	265	$[128 \times 128 \times 3 \times 3] \times 3$	11	104
			$[256 \times 128 \times 3 \times 3] \times 1$	13	88
			$[256 \times 256 \times 3 \times 3] \times 3$	13	177
			$[512 \times 256 \times 3 \times 3] \times 1$	19	121
			$[512 \times 512 \times 3 \times 3] \times 3$	19	242
Conv to Linear Size	q	l			
2448×25088	47	533			