Appendix for "Graph Structure of Neural Networks"

Jiaxuan You [1] **Jure Leskovec** [1] **Kaiming He** [2] **Saining Xie** [2]

1. Details for Generating Relational Graphs

Here we provide more details for how we generate graphs in Section 3.2. For all generators, we fix the number of nodes $n = 64$, and constrain the graph sparsity within $[0.125, 1.0]$.

Watts-Strogatz (WS) graphs. WS graphs are characterized by: (1) number of nodes n, (2) initial node degree k (must be an integer), (3) edge rewiring probability (randomness) p. We search over:

- degree $k \in$ `np.arange(8,62)`
- randomness $p \in$ `np.linspace(0,1,300)**2`
- 30 random seeds

Since graph measures are more sensitive when p is small, we increase the sample density of small p value by squaring p. In total, we generate $54 \times 300 \times 30 = 486,000$ WS graphs.

Erdős-Rényi (ER) graphs. ER graphs are characterized by: (1) number of nodes n, (2) number of edges m. We search over:

- edge number $m \in$ `np.arange` $(64 \times 4, 64 \times 63/2)$
- 30 random seeds

In total, we generate $1760 \times 30 = 52,800$ ER graphs.

Barabási-Albert (BA) graphs. ER graphs are characterized by: (1) number of nodes n, (2) number of existing nodes m that a new node connects to. We search over:

- $m \in$ `np.arange(4,30)`
- 300 random seeds

In total, we generate $26 \times 300 = 7,800$ ER graphs.

Harary graphs. Harary graphs are determined by: (1) number of nodes n, (2) number of edges m. We search over:

- edge number $m \in$ `np.arange(64*4,64*63/2)`

In total, we generate 1760 Harary graphs.

Ring graphs. Ring graphs are characterized by: (1) number of nodes n, (2) node degree k (integer). We search over:

- degree $k \in$ `np.arange(8,62)`

In total, we generate 54 ring graphs.

WS-flex graphs. We describe the detailed procedures of getting 3942 WS-flex graphs that we used in the experiments. WS-flex graphs are characterized by: (1) number of nodes n, (2) average node degree k (real number), (3) edge rewiring probability (randomness) p. We search over:

- degree $k \in$ `np.linspace(8,62,300)`
- randomness $p \in$ `np.linspace(0,1,300)**2`
- 30 random seeds

In total, we generate $300 \times 300 \times 30 = 2,700,000$ WS-flex graphs. Generating these WS-flex graphs (and computing their average path length and clustering coefficient) only takes about 1 hour on a 80 CPU core machine.

Next, we sub-sample 3942 graphs from these 2.7M candidate graphs. We create 2-d bins over the graph structure measures: (1) for average path length, we create 15×9 bins whose bin edges are given by `np.linspace` $(1, 4.5, 15 \times 9 + 1)$; (2) for clustering coefficient, we create 15×9 bins whose bin edges are given by `np.linspace` $(0, 1, 15 \times 9 + 1)$. We sub-sample 1 graph, whose graph structure measures fall within a given 2-d bin, for each of the 2-d bins. After gathering bins that have graphs, we get 3942 graphs in total.

For ImageNet experiments, we further sub-sample 52 graphs from these 3942 graphs. Specifically, we collect graphs in the bins whose bin ID $(i \mod 9) = 5$, so that the sub-sampled graphs are roughly uniformly distributed in the graph measure space.

2. Details for Matching the Reference FLOPS

Here we provide more details on matching the reference FLOPS for a given model. As described in Section 3.3, we vary the layer width of a neural network to match the reference FLOPS.

[1]Department of Computer Science, Stanford University [2]Facebook AI Research. Correspondence to: Jiaxuan You <jiaxuan@cs.stanford.edu>, Saining Xie <s9xie@fb.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

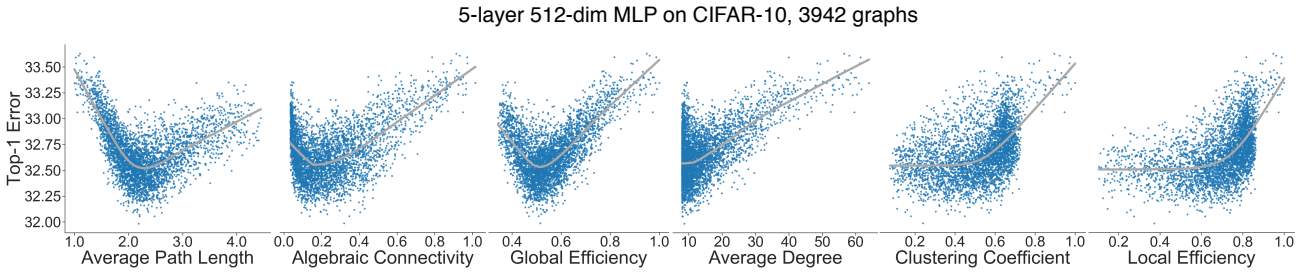

Figure 1: **More graph measures *vs.* neural network performance.** Global (**left 3**) and local (**right 3**) graph measures versus 5-layer 512-dim MLP performance on CIFAR-10.

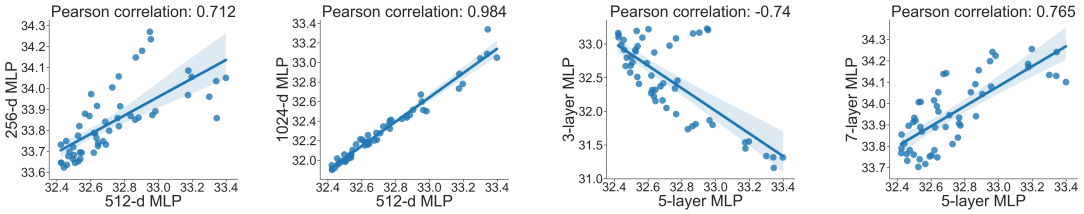

Figure 2: **Ablation study: varying width/depth of neural networks.** Pearson correlation between MLPs with different width/depth over the same set of 52 relational graphs. 5-layer 512-d MLP is the default architecture.

Specifically, to match the reference FLOPS, if all the layers have the same width, we incrementally vary the layer width by 1 for all the layers, then pick the layer width that has the fewest FLOPS above the reference FLOPS. In the scenarios where layer width varies in different stages, we fine-tune the network width in each of the stages via an iterative mechanism: (1) we incrementally vary the layer width of the narrowest stage by 1, while maintaining the ratio of layer width across all the stages, then fix the layer width for that stage; (2) we repeat (1) for the narrowest stage in the remaining stages. Using this technique, we can control the complexity of a model within 0.5% of the baseline FLOPS.

3. Details for Wall Clock Running Time

Training a baseline MLP (translated by a complete graph) on CIFAR-10 roughly takes 5 minutes on a NVIDIA V100 GPU, while training all baseline ResNets and EfficientNets approximately take a day on 8 NVIDIA V100 GPU. Due to the lack of mature support of sparse CUDA kernels, we implement relational graphs via applying sparse masks over dense weight matrices. The most sparse graph that we experiment with (sparsity = 0.125) is around 2x slower (in wall clock time) than the corresponding baseline graph.

4. Analysis with More Graph Measures

In the paper, we focus on 2 classic graph measures: clustering coefficient and average path length. Here we include the analysis using more graph measures. Specifically, we consider the following additional local and global graph measures.

Local graph measures. (1) Average degree: node degree averaged over all the nodes. (2) Local efficiency: a measure of how well information is exchanged by a node's neighbors when the node is removed, averaged over all the nodes.

Global graph measures. (1) Algebraic Connectivity: the second smallest eigenvalue of graph Laplacian. Graph Laplacian is defined as $A - D$, where A is the adjacency matrix and D is the degree matrix. (2) Global efficiency: a measure of how well information is exchanged across the whole network.

We plot the performance of 5-layer MLPs on CIFAR-10 dataset versus one of the graph structure measures, over the 3942 relational graphs that we experimented with. We use locally weighted linear regression to visualize the overall trend. From Figure 1, we can see that more graph measures exhibit the interesting U-shape correlation with respect to neural network predictive performance.

5. Ablation Study

5.1. Varying Width/Depth of Neural Networks

Here we investigate the effect of network width/depth on the performance of neural networks translated by the same set of relational graphs. Specifically, we study 5-layer MLPs with $[256, 512, 1024]$ dimension hidden layers, and 512-dim MLP with $[3, 5, 7]$ layers. We can see that the performance

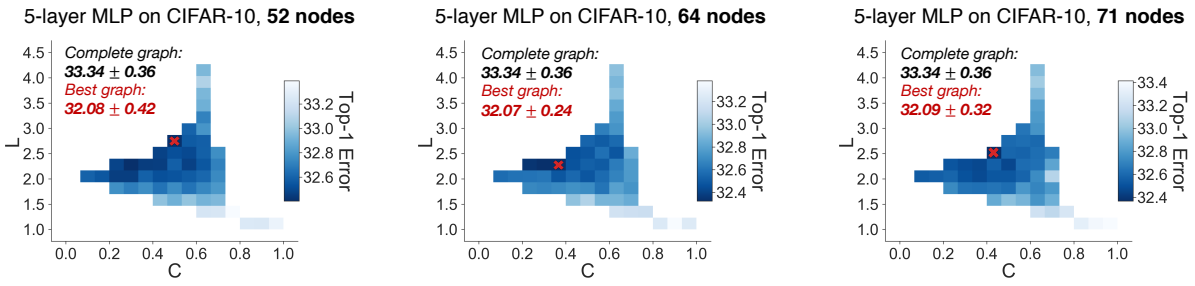

Figure 3: **Ablation study: varying number of nodes in relational graphs.** We average results from 482 relational graphs for 52-node graphs, 449 graphs for 64-node graphs, and 422 graphs for 71-node graphs.

of relational graphs with certain structural measures highly correlates across networks with different width.

The behavior is different when varying the network depth: while increasing the depth of MLP to 7 layers maintains a high correlation with 5-layer MLP, decreasing the depth of MLP to 3 layers completely reverse the correlation[1]. One possible explanation is that while sparse relational graphs may represent a more efficient neuron connectivity pattern, more rounds of message exchange are necessary for these neurons to fully communicate. Understanding how many rounds of message exchange are required by a given relational graph to reach optimal performance is an interesting direction left for future work.

5.2. Varying Number of Nodes in Relational Graphs

In Section 2.3, we show that an m-dim neural network layer can be flexibly represented by an n-node relational graph, as long as $n \leq m$. Here we show that varying the number of nodes in a relational graph has little effect on our findings.

In Figure 3, we show the results of 5-layer MLP on CIFAR-10, where we consider using 52-node (number of nodes of the cat cortex graph) and 71-node (number of nodes of the macaque whole cortex graph) relational graphs in addition to 64-node graphs used in the main paper. To cover the space of clustering C and path length L, we generate 482 graphs for 52-node graphs, 449 graphs for 64-node graphs, and 422 graphs for 71-node graphs. To save computational cost, we use fewer graphs than the 3942 graphs in the main paper. From the results, we can see the performance of the best graph is almost identical across these varied number of nodes, which justifies our claimed flexibility of selecting the number of nodes in a relational graph.

[1]Recall that when translating a relational graph, we leave the input and output layer unchanged; therefore, a 3-layer MLP only has 1 round of message passing over a given relational graph.

6. Discussion of Failure Cases

There are some special cases for convolutional neural networks on CIFAR-10, where the consistent sweet spot pattern ($C \in [0.43, 0.50]$, $L \in [1.82, 2.28]$) breaks and the best results are obtained with approximately fully-connected graphs. We visualize and discuss this phenomenon in Figure 4. We start from analyzing the results of 8-layer 64-dim CNN on ImageNet, where consistent sweep spot region emerges (Figure 4(a)). We then hold the model fixed, and switch the dataset to CIFAR-10. On CIFAR-10, we find that the complete graph performs better than most sparse graphs (Figure 4(b)).

Without a theoretical discussion on network overparameterization (Zhang et al., 2017) and intrinsic task difficulty (Li et al., 2018), we provide some empirical intuitions for those cases. As we have shown in main paper Figure 7, fully-connected neural networks can automatically learn graph structures during training. We hypothesize that the structural prior matters less when the task is simple, relative to the representation capacity and efficiency of the neural network and the learned graph structure is sufficient in those settings.

To verify this hypothesis, we reduce the model capacity by reducing the width of CNN from 64 to 16 dimensions (Figure 4(c)), and train it again on CIFAR-10. In this setting, we show that sparse graph structure significantly outperforms the complete graph again. We provide more examples that compares two scenarios in Figure 5. Note that with ImageNet, sparse relational graphs consistently yield better performance in the sweet spot regions, even when the model capacity is increased.

References

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *ICLR*, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

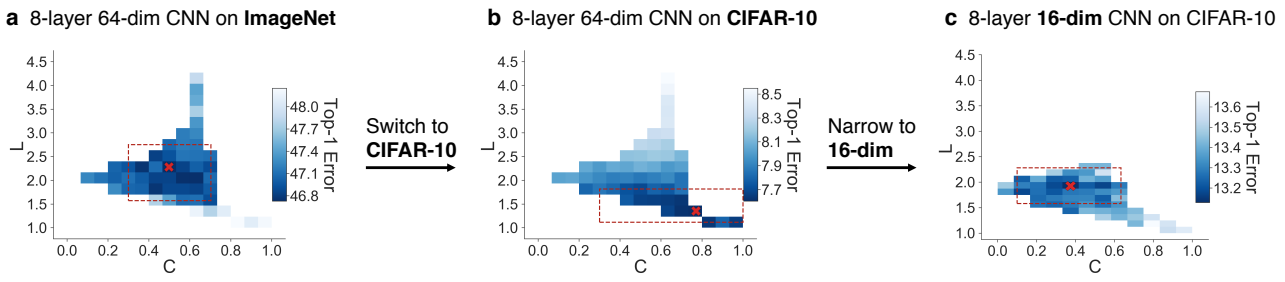

Figure 4: **(a)** We recap the results of 8-layer 64-dim CNN on ImageNet. **(b)** We apply the same architecture to CIFAR-10. Results from 449 relational graphs are averaged to 52 bins, using the technique in Section 5.2. We find that the complete graph performs better than most sparse graphs in this setting. **(c)** We hypothesize that since CIFAR-10 is an intrinsically much easier task than ImageNet, the graph structural priors might not matter when the representation capacity is abundant. To verify the hypothesis,we reduce the model capacity by reducing the width of CNN from 64 to 16 dimensions. Due to the narrowed dimensions, we explore relational graphs with 16 nodes instead of 64. Results of 326 relational graphs are averaged to 48 bins. In this setting, sparse graph structure significantly outperforms the complete graph again.

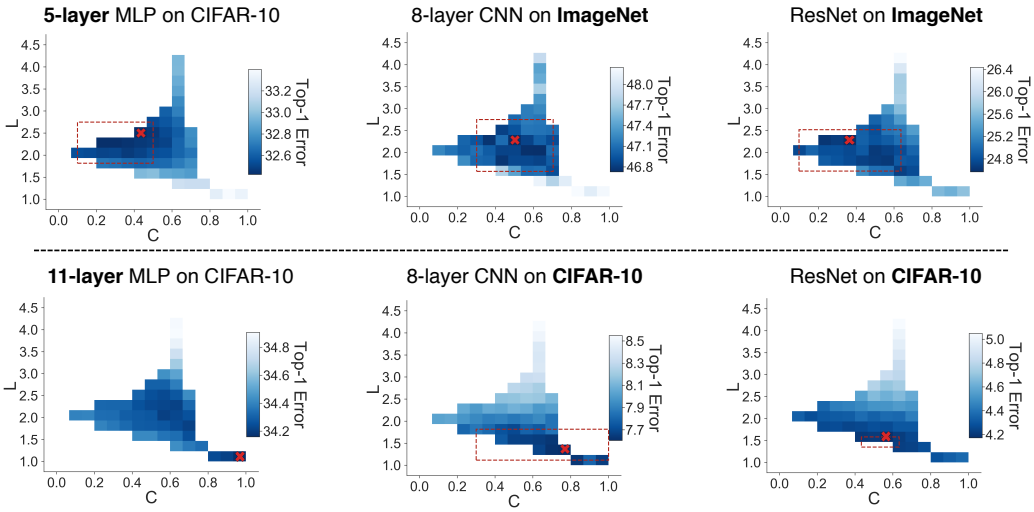

Figure 5: There are some special cases where the consistent sweet spots disappear, when training CNN models on CIFAR-10 (bottom row). For a simple task, we hypothesize that the graph structural priors might not matter when the representation capacity is abundant. However, for challenging tasks like ImageNet, sweet spots are consistent and sparse relational graphs always produce better results than the fully-connected/complete graph counterpart. (For 11-layer MLP on CIFAR-10, no sweet spot can be identified.)