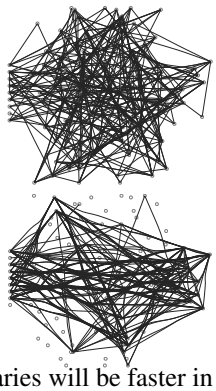
We are very grateful to the reviewers for their insightful, thorough, and thoughtful reviews.



• R1 & R2: **Code release.** We will release code, pretrained models and training config files so that the work is reproducible. We will also include additional low-level details in the paper.

• R1: **Visualization.** With thousands of nodes it is difficult to visualize connectivity in one figure though we are working on an interactive visualization in d3. See right for progress screen-shots of a smaller net shown early (top) and slightly later (bottom) in training (viewed best with zoom).

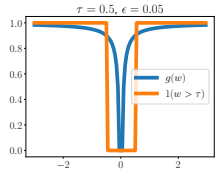
• R1: **Hardware.** The models are indeed inferior in terms of latency on current hardware. As per your suggestion we will update the paper to include this detail in addition to our response: Our key points are that **1)** We agree with the philosophy of the NeurIPS 2019 MicroNet Challenge – it may be important to explore models which might guide the direction of new hardware development (we are planning submission to the challenge). **2)** We will release a version of our model which performs inference via the Deep Graph Library (dgl) in Pytorch, we hope that dgl and similar libraries will be faster in the years to come. **3)** We also expect fast inference via sparse GPU kernels, which we anticipate will be developed.

• R2: **Sparsity:** Thank you for mentioning that *the unification of core parts of the sparse neural network literature combined with the neural architecture search problem is very nice*. As a useful experiment to include to showcase this, we apply Algorithm 1 separately to each filter of the version of ResNet50 used in [1] and obtain state of the art performance at the task of *training sparse networks from scratch* as described in [1]. With 10% and 20% of the parameters we achieve **74.6**$\%$ and **76.6**$\%$ top-1 accuracy respectively on ImageNet (keeping a dense first-layer – less than 10k params – we achieve **75.5**$\%$ and **76.8**$\%$). SOTA in [1] is 73.1% for 10% and 74.9% for 20%.

• R2: **Resolution & Stride:** As we discuss briefly in section 2.4, for the large scale experiments we consider a separate graph for each spatial resolution (the output of graph $\mathcal{G}_i$ is given as input to graph $\mathcal{G}_{i+1}$) and the convs performed at the input nodes are strided. The number of output nodes is the channels that MobileNetV1 has for that resolution. As we will elaborate on in the final revision, in the small scale setting either stride 1 may be used everywhere or some nodes may perform strided convs followed by zero-padding (as in Figure 2 of the paper).

• R2: **Theoretical Claims:** We agree that this work will benefit from theoretical analysis which extends beyond decreasing the loss on the mini-batch with the SGD update. In the final revision we believe that it will be useful to elaborate on this work from the perspective of the Lottery Ticket hypothesis and the myriad of recent work that seeks to explain it (wherein weights whose magnitude tends to zero are less important, as in line 5 of Algorithm 1) such as [2]. For now we invite you to reference the Extended Intuition section below for additional justification and in the final revision we will attempt to discuss or at least conjecture the extension of [2] to a student and teacher wiring.

• R3: **Extended Intuition.** We thank you for the excellent suggestion of offering the additional motivation of performing an approximation in the backwards pass. We are happy to see that we may arrive at the same objective in this way and we will include this exciting result. Indeed, we are calculating $\mathcal{I}_v = \sum_{u \in \mathcal{V}} g(w_{uv}) w_{uv} Z_u$ where $g(w_{uv}) = \mathbb{1}_{\{|w_{uv}| > \tau\}}$ in the forwards pass. Our method offers the following interpretation – we are using a soft*er* version of $g(w_{uv})$ in the backwards pass where g approaches 1 for large magnitude weights. Ignoring weights with $|w_{uv}| < \epsilon \ll \tau$ we may let $g(w_{uv}) = 1 - \epsilon |w_{uv}|^{-1} = 1 - \epsilon w_{uv}^{-1} \text{sign}(w_{uv})$ where $\text{sign}(w_{uv})$ is 1 if $w_{uv} > 0$ and -1 otherwise. Aligning with the update rule as needed, $\frac{\partial \mathcal{I}_v}{\partial w_{uv}} = \frac{\partial g(w_{uv})}{\partial w_{uv}} w_{uv} Z_u + g(w_{uv}) Z_u = \epsilon w_{uv}^{-2} \text{sign}(w_{uv}) w_{uv} Z_u + Z_u - \epsilon w_{uv}^{-1} \text{sign}(w_{uv}) Z_u = Z_u$.



• R3: **Related work.** We agree that this work would benefit from a better discussion of related material and less dynamic material. Thank you for the relevant papers which we will include.

• R3: **Clarity.** We will include more details concerning the small-scale experiment, wherein each node performs normalization followed by conv and relu and in all cases the graphs have 41,000 parameters. The experiments showcase that in the setting we consider, dynamic graphs are more expressive. We will also discuss the important relation between w and θ, including the effects of kernel scaling and batchnorm in which we believe theorem 3 from [2] may be crucial.

• R3: **500M FLOP Regime.** We explore the very low FLOP regime as it is an interesting application of the method – with enough wiring, a good wiring is likely less essential. During this response period we have completed a quick experiment to showcase that our method also performs well in larger FLOP settings. Since we are limited in time we only train for 100 epochs (250 epochs is standard for this setting). We take MobileNetV1 (width-mult 2) and restrict the number of edges to be the same as MobileNetV1 (width-mult 1) then apply our method to achieve **74.4**$\%$ accuracy on ImageNet with 582 MFLOPs and 4.2M params. NASNet-A, PNAS, DARTS, and vanilla MobileNetV1 achieve respectively 74.0, 74.2, 73.1, and 70.6 with 564, 588, 595, and 569 MFLOPs and 5.3M, 5.1M, 4.9M, and 4.2M params.

[1] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," 2019.

[2] Y. Tian, T. Jiang, Q. Gong, and A. Morcos, "Luck matters: Understanding training dynamics of deep relu networks," 2019.