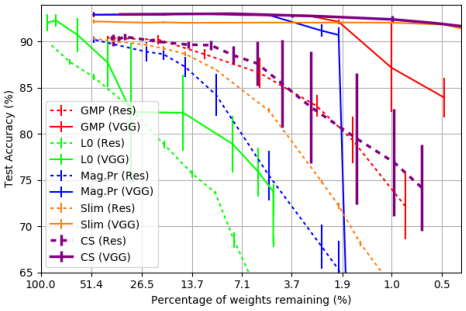
We thank the reviewers for the feedback and suggestions.

R1,R2,R3,R4 – Comparison to Louizos *et al.* [1]: Our approach to approximate the ℓ_0 objective is fundamentally different from prior work. Louizos *et al.* use tools from variational inference and stochastic optimization (see Appendix A of [1]), while we rely on continuation methods to construct a *deterministic* approximation. A stochastic approximation results in different sets of weights used in each forward pass: refer to Section 5.1 of Gale *et al.* [4], which reports that it failed to sparsify a Wide ResNet "across hundreds of experiments" without resulting in "test set performance akin to random guessing". Refer to the included plot for a comparison with the method of Louizos *et al.* on one-shot pruning a ResNet/VGG: note that we were only able to achieve these results with modifications to the original method, including fixing a stochastically-optimized mask prior to fine-tuning, without which the method fails to yield over 20% sparsity while maintaining performance that is not akin to random guessing. The need to choose a fixed, binary mask sample makes the method even less suited for ticket search. CS relies on a deterministic re-parameterization, avoiding gradient estimators altogether, while also having clear behavior at test time since a single mask is learned.



Method	ℓ_0	Mag.Prune	GMP	CS
VGG	32.0%	97.5%	98.0%	99.6%
ResNet	31.8%	75.5%	86.0%	89.3%

Highest sparsity achieved by each method while being within 2% (relative) performance of the dense model. Same protocol as in Section 4.3 (one-shot pruning on CIFAR-10).

R1,R2 – Comparisons: We will remove RigL, STR, and DNW from our Table 2 to avoid confusion, while still discussing the methods, pointing out differences in goals and methodology. L257 mentions that comparing methods with different methodologies is complicated, but we will remove entries to avoid confusion.

R1 – Matching pre-defined sparsities: Appendix F presents a variant of CS that is able to match exact sparsities by defining thresholds to binarize the mask values. For that experiment, we also tried having a fixed β=1 during training, which led to significantly worse results. **Motivation and LTH:** Ticket search is a strictly more general and harder task than pruning: every ticket search method can be used for pruning, but not every pruning method can perform ticket search (if the learned parameter-wise masks are not binary, then applying it to initial weights would change their magnitude, hence the resulting network would not be a 'ticket'). Moreover, ticket search provides insights into the behavior of neural networks, and is an invaluable tool to study how overparameterization affects optimization dynamics of deep networks. By making ticket search faster and parallelizable, we open doors to large-scale empirical studies on training dynamics of sparse networks. Finally, winning tickets have valuable applications other than just parameter efficiency – for example transfer learning (refer to [10,11,12] of our paper), where ticket search can be performed on small datasets and produced tickets can be successfully re-trained on larger tasks. **More pruning results:** IMP is actually a sensible pruning method: refer to Table 2. Also refer to Renda *et al.* [2] below, which proposes a method that is similar to IMP-Continued and is comparable to the state-of-the-art. The included plot presents comparison to additional methods, and we will extend it further in the camera-ready version.

R2 – Writing: We will use space from the extra page given to the camera-ready version to add a subsection introducing the reader to the nomenclature and precise definitions (in the same vein as how we define the Sparsest Matching Ticket and Best Performing Ticket in Section 4). We will also refer to sub-networks that are trained from early iterates as 'matching subnetworks' throughout the paper. **CS as pruning method:** We are currently collecting additional results with recent pruning methods like AMC, GMP (see plot above), and the method in Renda *et al.* [2]. We will add these to the ImageNet results in Table 2, and given space from the extra page for the camera-ready version, add pruning experiments with methods run for multiple rounds, where we can include iterative methods like Renda *et al.*'s approach. **Discussion on hyperparams:** We will use space from the extra page to add to the main text a more extensive discussion on the role of each hyperparameter and how it affects CS, while still keeping full details in the Appendix. Thanks for the detailed review and the list of suggestions; we commit to incorporate all the suggested changes to the camera-ready version of the paper, from using author names when referring to citations, to making statements throughout the paper more clear (following the points raised in part 8 of your review).

R3 – Refer to the plot above for comparison with Network Slimming, which we will add to the paper. We believe that characterizing what aspects of a pruning method yield better winning tickets is an exciting and valuable research question. We hope that our work further motivates research on this question by introducing a second methodology via which to perform ticket search - one that is fundamentally distinct from IMP and relies on different tools.

[1] Christos Louizos, Max Welling, Diederik P. Kingma. Learning Sparse Neural Networks through L_0 Regularization. ICLR, 2018.
[2] Alex Renda, Jonathan Frankle, Michael Carbin. Comparing Rewinding and Fine-tuning in Neural Network Pruning. ICLR, 2020.
[3] Suraj Srinivas, Akshayvarun Subramanya, R. Venkatesh Babu. Training Sparse Neural Networks. CVPR Workshops, 2017.
[4] Trevor Gale, Erich Elsen, Sara Hooker. The State of Sparsity in Deep Neural Networks. ICML Workshops, 2019.