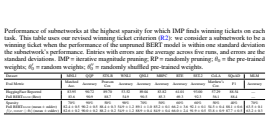
We thank the reviewers for their feedback, which we address point-by-point below. We include revised tables/figures at the bottom, including: Table 2 with an updated winning ticket criterion (R2), Table 3 with two metrics for some tasks (R2, R4), and Figure 2 split into two figures (R2). Due to limited space, reviewers must zoom in to see these items.

Shared Comments R2, R4: Some...tasks are intended to be evaluated on two metrics. Papers commonly report one or both of two metrics for MNLI, QQP, STS-B, and MRPC. Table 3 shows both of these metrics for those tasks. Besides STS-B (50% Pearson vs. 40% Spearman), winning ticket sparsities are the same on these tasks regardless of the metric.
R2, R4: Are the GLUE results from the test sets or dev sets? They are from the validation/dev sets.
R1: The claim of 'it may be possible to reduce the cost...of fine-tuning' might be incorrect. R4: It would be helpful to see speedup results. We acknowledge that the real-world speedup of a sparse network depends on the software libraries and the hardware. As R1 notes, the most direct way to do so is via structured sparsity. However, there is active work on accelerating unstructured sparsity via software (Elsen et al. for CNNs as cited by R4) and hardware (sparse support on the NVIDIA A100, GraphCore IPU, and Cerebras Wafer Scale Engine). These advances are a promising sign that future work will be able to exploit our unstructured sparsity, and our results serve as a strong baseline to guide this research.
R1: LTH...for structured pruning of BERT R3: Comparison to the sparsity of matching subnetworks [in Prasanna et al.].
As we discuss in Section 2, Prasanna et al. study the LTH for BERT with structured pruning of entire attention heads. We refer R1 to that section, where we discuss that work and other LTH results for structured pruning. Prasanna et al. look at "subnetworks that achieve 90% of full performance" (less accurate than our winning ticket criterion) and report how often each head survives pruning (rather than the overall sparsity), so we cannot directly compare to their results.

Reviewer 1, Reviewer 4: All technical comments addressed above. We acknowledge R4's presentation comments.

Reviewer 2: This paper overclaims on many things, so the claimed result should be taken with a grain of salt.
We have addressed all of the reviewer's concerns point-by-point below.
(A) This paper relaxes the definition [of a winning ticket] to be achieving performance within two points of the baseline.
Our motivation for the 2% threshold was to account for variation between runs (see (D)) rather than to introduce a more permissive criterion. To make this clear, we have revised our winning ticket criterion to account for this variation in a stricter, task-specific way. Specifically, we consider a subnetwork to be a winning ticket when the mean full BERT performance is within one standard deviation of the mean subnetwork performance (computed over five runs; see (B)). We updated Table 2 (below) accordingly; sparsities only change on STS-B (70% $\rightarrow$ 50%) and SQuaD (70% $\rightarrow$ 40%).
(B) Why the baseline compared against is the average of three runs, and why that average performance is used.
To clarify, we perform multiple runs with different random seeds for the data order; we report the average over these runs. (We always use the same HuggingFace BERT initialization.) For each baseline run, we fine-tune with a random data order. For each lottery ticket run, we train with a random data order, prune, rewind, and re-train with another random data order. The updated Table 2 below shows means and standard deviations across five such runs. We average over multiple runs in this way to show that our results are robust and are not cherry-picked. This is standard practice in lottery ticket work [13, 14, 15, 16, 17, 18, 19], and it is required by the Machine Learning Reproducibility Checklist.
(C) Figure 2 has coloring issues. The colors and numbers in Figure 2 described two separate comparisons; we acknowledge this was confusing, and we have split this into Figures 2a and 2b below. The numbers in Figure 2 showed $\textsc{Transfer}(\mathcal{S}, \mathcal{T})$ minus the performance of unpruned BERT on task $\mathcal{T}$; this determined whether the transferred subnetwork was a winning ticket. As we explain (L240), however, 70% sparsity is too sparse to find winning tickets on several tasks. As such, we also compare whether transfer performance $\textsc{Transfer}(\mathcal{S}, \mathcal{T})$ is at least as high as same-task performance $\textsc{Transfer}(\mathcal{T}, \mathcal{T})$ even if neither is a winning ticket. Cells in Figure 2 were blue when this was the case, and this could be computed manually by checking if a cell's value was at least as high as the cell in the same-column diagonal. We apologize for the confusion; we believe Figures 2a and 2b address this concern and improve clarity.
(D) BERT performance is below others' reported performance. We use the HuggingFace reference implementation of BERT Base; our best numbers are in line with those reported by HuggingFace (see Tables 2 and 3 below). Reported numbers can vary widely based on number of runs, metric (mean/median/best), and hyperparameter search [see *Show Your Work*, Dodge et al. EMNLP 2019]. STILT is not comparable to our numbers: (1) It uses BERT Large, not BERT Base. (2) It "perform[s] 20 random restarts...and report[s] the results...that performed best," while we report averages.

Reviewer 3: IMP...fails at finding matching subnetworks on tasks with relatively smaller training sets. IMP finds winning tickets (which are a form of matching subnetwork) on all tasks, and "there is no discernible relationship between the sparsities for each task and the properties of the task itself" (L164). Smaller training sets only seem to affect rewinding, which we find to be unnecessary for our goal of uncovering sparse, transferable subnetworks.


Revised Table 2

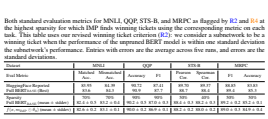


Table 3: Both Metrics


Figure 2a: Transfer Winning Tickets


Figure 2b: Transfer vs. Same-Task