We sincerely thank the reviewers for sharing their valuable feedback while pointing out weaknesses in our work and suggesting presentations improvements.

All - Report the model size as opposed to sparsity percentage / Claims are not quite fair since they are based on relative sparsity percentage instead of total non-zero parameter count. There was some confusion, which we will clarify, in R1/R2 about reporting sparsity percentages. All percentages are relative to BERT base, and correspond *exactly* to model size (even for MiniBERT and Layer Drop). To address we will include in the appendix the main results (Figures 2&3, Tables 2&3) to plot the performance against the number of non-zero parameters in the encoder. For instance, 3% corresponds to 2.6 millions (M) non-zero parameters in the encoder, 10% to 8.5M, 20% to 17M.

R1 - Is this distillation only on the training set, or is there data augmentation? We do not use data augmentation in any of our experiments. The model is trained solely on the training set. The distillation signal comes from the dense teacher of the same size plotted in Figures 2&3 in cyan. We follow the vanilla setup described in Hinton et al. [2014].

R1 - Can the authors comment how movement pruning might work for generative tasks? Interesting idea. For encoder-decoder setups, we can augment the Fully Connected layers in the Transformer block with score matrices, learn these scores during training and discard them after pruning. While we have not yet tested this extensively, initial small scale experiments on DistilBART for summarization (3 layers encoder and 3 layers decoder) give the following results at 90% sparsity (Rouge-2/Rouge-L on XSum): Dense=12.3/27.3, MaP=8.8/22.3, L_0=9.8/23.6, MvP=11.0/25.0, indicating that movement pruning is also promising in this setting.

R2 - As with most work on pruning, it is not yet possible to realize efficiency gains on GPU. We agree that inference speed for pruned models is still an open concern. However, we argue that our work (and other pruning studies) have direct applications in real-world settings. As highlighted in Han et al. [2016], most of the energy consumption for on-device deep learning comes from the loading of the weights. Reducing the memory size of the model is a crucial step towards enabling more on-device applications even without speed-ups. Moreover, chip manufacturers are making progress towards accelerating sparse models in many settings (for instance, A100 from Nvidia). For our models, working with new sparse inference frameworks, we are already able to get 3x speed gain on CPU using a sparse model.

R3 - The results presented seem correct, but I'm concerned about the lack of comparison to other approaches for compressing LMs during fine-tuning. The reviewer mentions several specific papers. We have compared against (1909.12486) in our submission: it is displayed as *RPP* in Figure 2&4. Works (2002.08307) and (2002.11794) apply unstructured magnitude pruning as a post-hoc operation whereas we use *automated gradual pruning* [Zhu and Gupta, 2018], a variant of magnitude pruning which improves on these methods by enabling masked weights to be updated. For instance, (2002.08307) obtains a score of 58.7 on MNLI compared to 78.4 at 90% sparsity with automated gradual pruning. Finally (1910.06360) compares multiple methods to compute structured masking (L_0 regularization and head importance as described in [Michel et al., 2019]) and found that structured L_0 regularization performs best. We did not find any implementation for this work, so to be fair, we presented a strong unstructured L_0 regularization baseline. We will also add a reference to the related NeurIPS2019 work (1907.04840).

R4 - The designed movement pruning approach is lacking of novelty, as various pruning heuristics have been proposed. As highlighted in Section 4, our method is indeed similar to previous general propositions, such as L_0 regularization. We frame our study in the context of *transfer learning* and how it differs from standard supervised learning. In this setting, the change paradigm (moving away from 0 instead of being far from 0) is crucial in high sparsity regimes. To the best of our knowledge, it is not a perspective that is commonly developed in other works since a significant part of these focus on pruning non-pre-trained models. Movement pruning shows strong performances in this context, out-performing L_0 regularization while being very simple (both to understand and implement).

R4 - Does the poor performance at low sparsity level mean that the proposed importance criterion is not suitable for low sparsity pruning? We have not found a convincing explanation for this phenomenon: movement-based pruning compare less favorably against magnitude pruning at low sparsity. We leave this interesting exploration for future work.

References

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS*, 2014.

Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark Horowitz, and William J. Dally. Eie: Efficient inference engine on compressed deep neural network. In *ISCA*, 2016.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *ICLR*, 2018.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *NeurIPS*, 2019.