Appendix A Optimization of task-specific last layers alone fails to fine-tune

Optimization of only task-specific layers does not lead to successful fine-tuning. For instance, for the MRPC task, freezing parameter weights in the pre-trained model and optimizing the task-specific last layer alone yields a non-performing model. Across 10 independent runs, the model consistently predicts all 1's for the paraphrase classification task, yielding an F1 score of 81.2. This is a significant degradation compared to the baseline performance of 89.4 ± 0.7 across multiple runs (Table 3). Thus, it is critical to fine-tune layers in the pre-trained model and not just the task-specific layers alone.

Appendix B Learning rate of supermask training

Supermask training requires a much larger learning rate compared to typical training (Zhang et al., 2019). While a learning rate of 2×10^{-5} is used for optimizing weights, a learning rate of 2×10^{-1} is used for optimizing masks. We notice a degradation in performance at smaller learning rates for supermask training (Table 5). This pattern holds true across GLUE tasks.

Table 5: MRPC low-sparsity supermask performance at learning rates from 2×10^{-5} and 2×10^{-1}.

Learning-rate	F1 score
2×10^{-1}	91.3 ± 0.4
2×10^{-2}	82.0 ± 0.2
2×10^{-3}	0.0
2×10^{-4}	0.0
2×10^{-5}	0.0

Appendix C Correlation between initial and final sparsities of supermasks

There is no straightforward control of the amount of weights pruned in previous reports of supermask training (Zhang et al., 2019; Mallya et al., 2018). We find that setting the initial sparsity through a soft magnitude-based pruning mask controls the final sparsity level, which we use to produce supermasks of varied sparsity levels. Figure 7 shows this correlation between initial and final sparsities of supermasks for different GLUE tasks. We note that, at lower initial sparsity levels, the supermask is pushed to a greater sparsity level, whereas at higher sparsity levels, the supermask is pushed to a lower sparsity level. This pattern is similar across GLUE tasks but is most prominent in the MNLI task, scaling with the number of fine-tuning steps (Table 1).

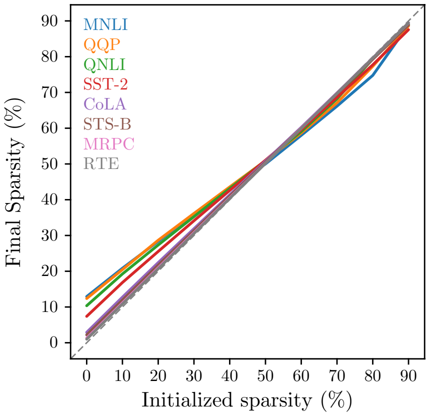

Figure 7: Initial versus final sparsity levels of supermasks.

Appendix D Correlation of parameter distance with fine-tuning steps

In order to understand how distance in parameter space increases as a function of fine-tuning steps, we study this relationship across GLUE tasks. We find that parameter distance scales with the number of fine-tuning steps by a power law with exponent close to 0.5 (Figure 8).

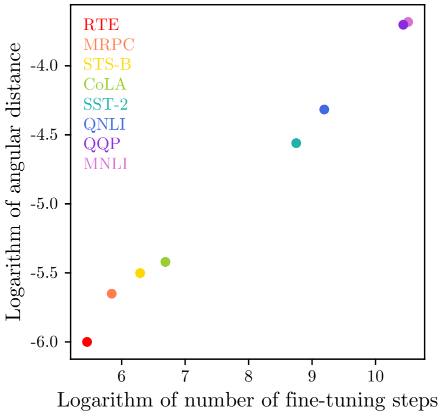

Figure 8: Correlation of parameter distance with the number of fine-tuning iterations. Shown are angular distances. Each data point corresponds to a different GLUE task.

Appendix E Fine-tuning with iterative pruning

We also use iterative pruning (Zhu and Gupta, 2017) during fine-tuning to produce sparse models. Pruning is based on weight magnitudes in each layer and is performed periodically during fine-tuning with sparsity gradually increasing from 0% to a final level according to a cubic schedule.

Iterative pruning during fine-tuning (Figure 9) outperforms supermask training (Figure 3) at higher sparsity levels. While supermask training remains successful up to 40% sparsity, iterative pruning produces binary masks up to 50%

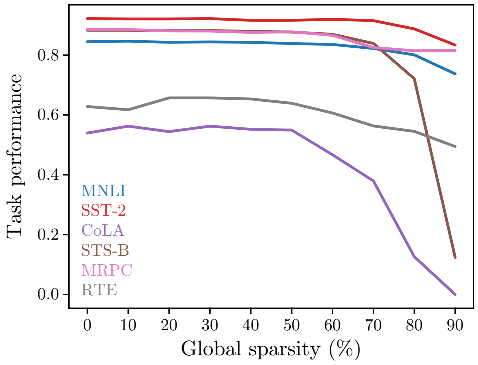

Figure 9: Iterative pruning during fine-tuning. We plot the evaluation performance at sparsity levels from 10% to 90% across GLUE tasks. Note the baseline performance for each task marked by the leftmost end of each curve (0% sparsity).

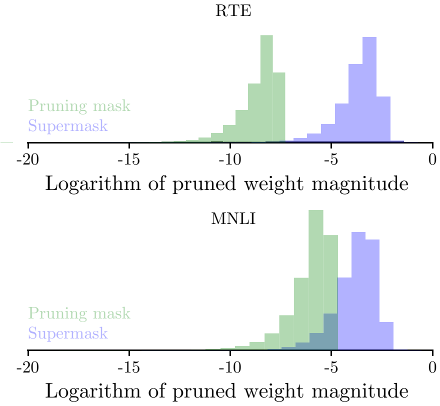

Figure 10: Pruned weight distributions, compared between supermask and magnitude-based pruning. Shown for the RTE and MNLI fine-tuning tasks.

sparse and for some tasks even sparser without significant performance degradation. Though iterative pruning produces sparse models, the fine-tuned models do not share parameters–one still needs to store all parameters for each task. Fine-tuned supermasks, on the other hand, store only a binary mask of certain layers for each task, with all tasks sharing a same set of underlying pre-trained weights.

Appendix F Fine-tuned supermasks are not trivial

How does the learning of a supermask actually work? Does a supermask simply learn to prune away the weights with smallest magnitudes? Since pure magnitude-based pruning of pre-trained weights does not perform any task-specific learning, we reason that the weight entries being set to zero by the supermask must have significant values. Here, we inspect the magnitudes of the pre-trained weights zeroed by the supermasks (Figure 10, Table 6). These weights turn out to have remarkably higher magnitudes than the smallest entries, suggesting the learning of supermasks is not trivial magnitude-based pruning.

Appendix G Learning curves of low-sparsity supermask fine-tuning

Our results suggest that supermask fine-tuning, if initialized at 0% sparsity, gradually increases sparsity during optimization, reaching a final sparsity level that correlates with the number of fine-tuning steps (Table 4). For MNLI, the GLUE task with the most fine-tuning steps, the sparsity level reaches 12.9%. We ask how prediction accuracy grows with sparsity during fine-tuning. As shown in Figure 11, like model performance, sparsity rapidly grows during the initial phase of fine-tuning. This makes model performance increase roughly linearly with sparsity.

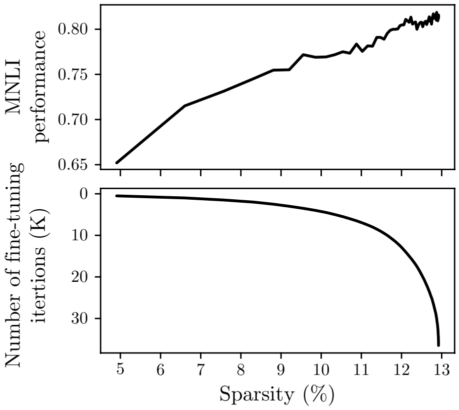

Figure 11: Learning curves of MNLI low-sparsity supermask fine-tuning.

Table 6: Comparison between weights pruned with low-sparsity supermasks (initialized at 0% sparsity) and weights pruned with magnitude-based pruning at the same final sparsity. We report the maximum and mean magnitude of the pruned weights. The last row shows percentages of the overlap between the supermask and the magnitude-based pruning mask, *i.e.* the percentages of weights zeroed by the supermask that are also the smallest weights.

GLUE task	MNLI	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE
Pruned max	0.0093	0.0093	0.0075	0.0059	0.0022	0.0018	0.0009	0.0007
Supermask max	1.7	6.4	2.5	1.7	1.1	2.8	1.8	2.8
Pruned mean	0.0033	0.0032	0.0026	0.0020	0.0008	0.0006	0.0003	0.0002
Supermask mean	0.032	0.033	0.033	0.035	0.037	0.036	0.038	0.036
Overlap	11.1%	10.0%	6.7%	3.6%	0.7%	0.7%	0.7%	0.7%