# Diverse Lottery Tickets Boost Ensemble from a Single Pretrained Model

**Sosuke Kobayashi[1,2] Shun Kiyono[3,1] Jun Suzuki[1,3] Kentaro Inui[1,3]**
Tohoku University[1] Preferred Networks, Inc.[2] RIKEN[3]
sosk@preferred.jp shun.kiyono@riken.jp
jun.suzuki@tohoku.ac.jp inui@tohoku.ac.jp

## Abstract

Ensembling is a popular method used to improve performance as a last resort. However, ensembling multiple models finetuned from a single pretrained model has been not very effective; this could be due to the lack of diversity among ensemble members. This paper proposes *Multi-Ticket Ensemble*, which finetunes different subnetworks of a single pretrained model and ensembles them. We empirically demonstrated that winning-ticket subnetworks produced more diverse predictions than dense networks, and their ensemble outperformed the standard ensemble on some tasks.

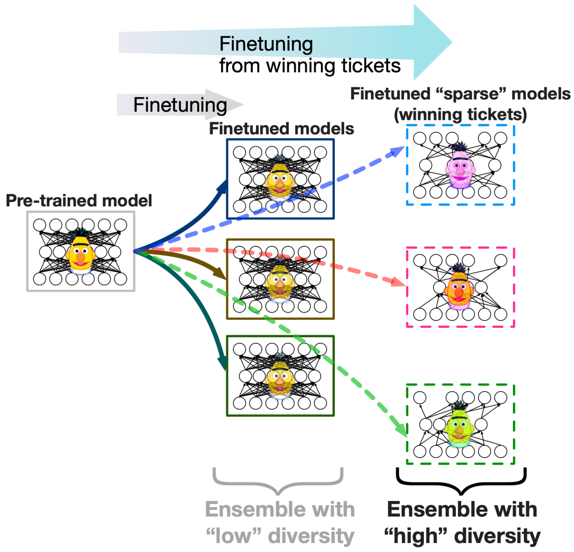


Figure 1: When finetuning from a single pretrained model (left), the models are less diverse (center). If we finetune different sparse subnetworks, they become more diverse and make the ensemble effective (right).

## 1 Introduction

*Ensembling* (Levin et al., 1989; Domingos, 1997) has long been an easy and effective approach to improve model performance by averaging the outputs of multiple comparable but independent models. Allen-Zhu and Li (2020) explain that different models obtain different views for judgments, and the ensemble uses complementary views to make more robust decisions. A good ensemble requires diverse member models. However, how to encourage diversity without sacrificing the accuracy of each model is non-trivial (Liu and Yao, 1999; Kirillov et al., 2016; Rame and Cord, 2021).

The *pretrain-then-finetune* paradigm has become another best practice for achieving state-of-the-art performance on NLP tasks (Devlin et al., 2019). The cost of large-scale pretraining, however, is enormously high (Sharir et al., 2020); This often makes it difficult to independently pretrain multiple models. Therefore, most researchers and practitioners only use a *single* pretrained model, which is distributed by resource-rich organizations.

This situation brings up a novel question to ensemble learning: Can we make an effective ensemble from only *a single pre-trained model*? Although ensembles can be combined with the pretrain-then-finetune paradigm, an ensemble of models finetuned from *a single pretrained model* is much less effective than that using *different pretrained models from scratch* in many tasks (Raffel et al., 2020). Naïve ensemble offers limited improvements, possibly due to the lack of diversity of finetuning from the same initial parameters.

In this paper, we propose a simple yet effective method called *Multi-Ticket Ensemble*, ensembling finetuned *winning-ticket subnetworks* (Frankle and Carbin, 2019) in a single pretrained model. We empirically demonstrate that pruning a single pretrained model can make diverse models, and their ensemble can outperform the naïve dense ensemble if winning-ticket subnetworks are found.

## 2 Diversity in a Single Pretrained Model

In this paper, we discuss the most standard way of ensemble, which averages the outputs of multiple neural networks; each has the same architecture but different parameters. That is, let $f(\boldsymbol{x}; \boldsymbol{\theta})$ be the output of a model with the parameter vector $\boldsymbol{\theta}$ given the input $\boldsymbol{x}$, the output of an ensemble is $f_{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\theta} \in \mathcal{M}} f(\boldsymbol{x}; \boldsymbol{\theta})/|\mathcal{M}|$, where $\mathcal{M} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{|\mathcal{M}|}\}$ is the member parameters.

### 2.1 Diversity from Finetuning

As discussed, when constructing an ensemble $f_{\mathcal{M}}$ by finetuning from a single pretrained model multiple times with different random seeds $\{s_1, ..., s_{|\mathcal{M}|}\}$, the boost in performance tends to be only marginal. In the case of BERT (Devlin et al., 2019) and its variants, three sources of diversities can be considered: random initialization of the task-specific layer, dataset shuffling for stochastic gradient descent (SGD), and dropout. However, empirically, such finetuned parameters tend not to be largely different from the initial parameters, and they do not lead to diverse models (Radiya-Dixit and Wang, 2020). Of course, if one adds significant noise to the parameters, it leads to diversity; however, it would also hurt accuracy.

### 2.2 Diversity from Pruning

To make models ensuring both accuracy and diversity, we focus on subnetworks in the pretrained model. Different subnetworks employ different subspaces of the pre-trained knowledge (Radiya-Dixit and Wang, 2020; Zhao et al., 2020; Cao et al., 2021); this would help the subnetworks to acquire different views, which can be a source of desired diversity[1]. Also, in terms of accuracy, recent studies on the *lottery ticket hypothesis* (Frankle and Carbin, 2019) suggest that a dense network at initialization contains a subnetwork, called the *winning ticket*, whose accuracy becomes comparable to that of the dense one after the same training. Interestingly, the pretrained BERT also has a winning ticket for finetuning on downstream tasks (Chen et al., 2020). Thus, if we can find *diverse winning tickets*, they can be good ensemble members with the two desirable properties: diversity and accuracy.

## 3 Subnetwork Exploration

We propose a simple yet effective method, *multi-ticket ensemble*, which finetunes different subnetworks instead of dense networks. Because it could be a key how to find subnetworks, we explore three variants based on iterative magnitude pruning.

[1]Some concurrent and recent studies also investigate subnetworks for effective ensemble (Durasov et al., 2021; Havasi et al., 2021) for training-from-scratch settings of image recognition.

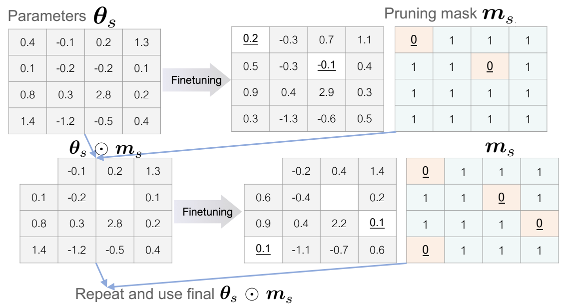


Figure 2: Overview of iterative magnitude pruning (Section 3.1). We can also use regularizers during finetuning to diversify pruning (Section 3.2).

### 3.1 Iterative Magnitude Pruning

We employ iterative magnitude pruning (Frankle and Carbin, 2019) to find winning tickets for simplicity. Other sophisticated options are left for future work. Here, we explain the algorithm (refer to the paper for details). The algorithm explores a good pruning mask via rehearsals of finetuning. First, it completes a finetuning procedure of an initialized dense network and identifies the parameters with the 10% lowest magnitudes as the targets of pruning. Then, it makes the pruned subnetwork and resets its parameters to the originally-initialized (sub-)parameters. This finetune-prune-reset process is repeated until reaching the desired pruning ratio. We used 30% as pruning ratio.

### 3.2 Pruning with Regularizer

We discussed that finetuning with different random seeds did not lead to diverse parameters in Section 2.1. Therefore, iterative magnitude pruning with different seeds could also produce less diverse subnetworks. Thus, we also explore means of diversifying pruning patterns by enforcing different parameters to have lower magnitudes. Motivated by this, we experiment with a simple approach, applying an $L_1$ regularizer (i.e., magnitude decay) to different parameters selectively depending on the random seeds. Specifically, we explore two policies to determine which parameters are decayed and how strongly they are, i.e., the element-wise coefficients of the $L_1$ regularizer, $\boldsymbol{l}_s \in \mathbb{R}_{\geq 0}^{|\theta|}$. During finetuning (for pruning), we add a regularization term $\tau ||\boldsymbol{\theta}_s \odot \boldsymbol{l}_s||_1$ with a positive scalar coefficient $\tau$ into the loss of the task (e.g., cross entropy for classification), where $\odot$ is element-wise product. This softly enforces various parameters to have a lower magnitude among a set of random seeds and could lead various parameters to be pruned.

**Active Masking** To maximize the diversity of the surviving parameters of member models, it is necessary to prune the surviving parameters of the random seed $s_1$ when building a model with the next random seed $s_2$. Thus, during finetuning with seed $s_2$, we apply the $L_1$ regularizer on the first surviving parameters. Likewise, with the following seeds $s_3, s_4, ..., s_i, ..., s_{|\mathcal{M}|}$, we cumulatively use the average of the surviving masks as the regularizer coefficient mask. Let $\boldsymbol{m}_{s_j} \in \{0,1\}^{|\boldsymbol{\theta}|}$ be the pruning mask indicating surviving parameters from seed $s_j$, the coefficient mask with seed $s_i$ is $\boldsymbol{l}_{s_i} = \sum_{j<i} \boldsymbol{m}_{s_j}/(i-1)$. We call this affirmative policy as *active masking*.

**Random Masking** In active masking, each coefficient mask has a sequential dependence on the preceding random seeds. Thus, the training of ensemble members cannot be parallelized. Therefore, we also experiment with a simpler and parallelizable variant, *random masking*, where a mask is independently and randomly generated from a random seed. With a random seed $s_i$, we generate the seed-dependent random binary mask, i.e., $\boldsymbol{l}_s = \boldsymbol{m}_{s_i}^{\text{rand}} \in \{0,1\}^{|\boldsymbol{\theta}|}$, where each element is sampled from Bernoulli distribution and 0's probability equals to the target pruning ratio.

## 4 Experiments

We evaluate the performance of ensembles using four finetuning schemes: (1) finetuning without pruning (BASELINE), (2) finetuning of lottery-ticket subnetworks found with the naïve iterative magnitude pruning (BASE-LT), and (3) with $L_1$ regularizer by the active masking (ACTIVE-LT) or (4) random masking (RANDOM-LT). We also compare with (5) BAGGING-based ensemble, which trains dense models on different random 90% training subsets. We use the GLUE benchmark (Wang et al., 2018) as tasks. The implementation and settings follow Chen et al. (2020)[2] using the Transformers library (Wolf et al., 2020) and its bert-base-uncased pretrained model. We report the average performance using twenty different random seeds. Ensembles are evaluated using exhaustive combinations of five members. We also perform Student's t-test for validating statistical significance[3]. Note that, while the experiments focus on using BERT, we believe that the insights would be helpful to other pretrain-then-finetune settings in general[4].

| | MRPC | | | STS-B | | |
|---|---|---|---|---|---|---|
| | single | ens. | diff. | single | ens. | diff. |
| BASELINE | 83.48 | 84.34 | +0.86 | 88.35 | 89.04 | +0.69 |
| (BAGGING) | 82.87 | 84.19 | +1.32 | 88.17 | 88.84 | +0.68 |
| BASE-LT | 83.84 | <u>*84.98*</u> | *+1.14* | 88.37 | *89.16* | *+0.79* |
| ACTIVE-LT | 83.22 | *84.60* | ***+1.38*** | 88.39 | ***89.32*** | <u>***+0.94***</u> |
| RANDOM-LT | 83.53 | <u>*85.05*</u> | <u>***+1.52***</u> | 88.49 | <u>***89.35***</u> | ***+0.86*** |

Table 1: The performances (single, ens.) and the improvements by ensembling (diff.). *Italic* indicates that the value is significantly larger than that of BASELINE. ***Bold-italic*** indicates significantly larger than that of both BASELINE and BASE-LT. <u>Underline</u> indicates the best.

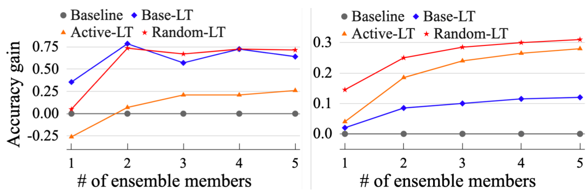


Figure 3: Comparison of the performances and the number of ensemble members on MRPC (left) and STS-B (right). They are represented as the relative gain compared with BASELINE's accuracy.

### 4.1 Accuracy

We show the results on MRPC (Dolan and Brockett, 2005) and STS-B (Cer et al., 2017) in Table 1. Multi-ticket ensembles (*-LT) outperform BASELINE and BAGGING significantly ($p < 0.001$). This result supports the effectiveness of multi-ticket ensemble. Note that the improvements of *-LT are attributable to ensembling (diff.) rather than to any performance gains of the individual models (single). We also plot the improvements (ens. values relative to BASELINE) as a function of the number of ensemble members on MRPC and STS-B in Figure 3. This also clearly shows that while the single models of *-LT have accuracy similar to BASELINE, the gains appear when ensembling them. While multi-ticket ensemble works well even with the naive pruning method (BASE-LT), RANDOM-LT and ACTIVE-LT achieve the better ensembling effect on average; this suggests the effectiveness of regularizers. Interestingly, RANDOM-LT is simpler but more effective than ACTIVE-LT.

[2]We found a bug in Chen et al. (2020)'s implementation on GitHub, so we fixed it and experimented with the correct version.

[3]Note that not all evaluation samples satisfy independence assumption.

[4]Raffel et al. (2020) reported that the same problem happened on almost all tasks (GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), SQuAD (Rajpurkar et al., 2016), summarization, and machine translation) using the T5 model.

**When Winning Tickets are Less Accurate** Does multi-ticket ensemble work well on any tasks? The answer is no. To enjoy the benefit from multi-ticket ensemble, we have to find diverse winning-ticket subnetworks sufficiently comparable to their dense network. When winning tickets are less accurate than the baseline, their ensembles often fail to outperform the baseline's ensemble. It happened to CoLA (Warstadt et al., 2019), QNLI (Rajpurkar et al., 2016), SST-2 (Socher et al., 2013), MNLI (Williams et al., 2018); the naive iterative magnitude pruning did not find comparable winning-ticket subnetworks (with or sometimes even without regularizers)[5,6,7]. Note that, even in such a case, RANDOM-LT often yielded a higher effect of ensembling (diff.), while the degradation of single models canceled out the effect in total, and BAGGING also failed to improve. More sophisticated pruning methods (Blalock et al., 2020; Sanh et al., 2020) or tuning will find better winning-ticket subnetworks and maximize the opportunities for multi-ticket ensemble in future work.

## 4.2 Diversity of Predictions

As an auxiliary analysis of behaviors, we show that each subnetwork produces diverse predictions. Because any existing diversity scores do not completely explain or justify the ensemble performance[8], we discuss only rough trends in five popular metrics of classification diversity; Q statistic (Yule, 1900), ratio errors (Aksela, 2003), negative double fault (Giacinto and Roli, 2001), disagreement measure (Skalak, 1996), and correlation coefficient (Kuncheva and Whitaker, 2003). See Kuncheva and Whitaker (2003); Cruz et al. (2020) for their summarized definitions. As shown in Table 2, in all the metrics, winning-ticket subnetworks (*-LT) produced more diverse predictions than the baseline using the dense networks (BASELINE).

| | Q↓ | R↑ | ND↑ | D↑ | C↓ |
|---|---|---|---|---|---|
| BASELINE | 0.96 | 0.72 | -0.12 | 0.09 | 0.69 |
| BASE-LT | 0.93 | 1.00 | -0.11 | 0.10 | 0.62 |
| ACTIVE-LT | 0.94 | 0.94 | -0.11 | 0.11 | 0.62 |
| RANDOM-LT | 0.94 | 0.94 | -0.11 | 0.10 | 0.63 |

Table 2: Diversity metrics on MRPC. The signs, ↓ and ↑, indicate that the metric gets lower and higher when the predictions are diverse. Q = Q statistic, R = ratio errors, ND = negative double fault, D = disagreement measure, C = correlation coefficient.

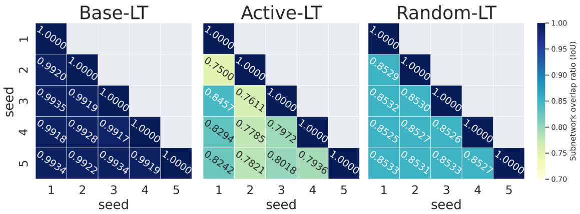


Figure 4: Overlap ratio of pruning masks $\boldsymbol{m}_{s_i}$ between different seeds on MRPC. The lower (yellower) the value is, the more dissimilar the two masks are.

## 4.3 Diversity of Subnetwork Structures

We finally revealed the diversity of the subnetwork structures on MRPC. We calculated the overlap ratio of two pruning masks, which is defined as intersection over union, IoU $= \frac{|\boldsymbol{m}_i \cap \boldsymbol{m}_j|}{|\boldsymbol{m}_i \cup \boldsymbol{m}_j|}$ (Chen et al., 2020). In Figure 4, we show the overlap ratio between the pruning masks for the five random seeds, i.e., $\{\boldsymbol{m}_{s_1}, ..., \boldsymbol{m}_{s_5}\}$. At first, we can see that ACTIVE-LT and RANDOM-LT using the regularizers resulted in diverse pruning. This higher diversity could lead to the best improvements by ensembling, as discussed in Section 4.1. Secondly, BASE-LT produced surprisingly similar (99%) pruning masks with different random seeds. However, recall that even BASE-LT using the naïve iterative magnitude pruning performed better than BASELINE. This result shows that even seemingly small changes in structure can improve the diversity of predictions and the performance of the ensemble.

# 5 Conclusion

We raised a question on difficulty of ensembling large-scale pretrained models. As an efficient remedy, we explored methods to use subnetworks in a single model. We empirically demonstrated that ensembling winning-ticket subnetworks could outperform the dense ensembles via diversification and indicated a limitation too.

[5] Although some studies (Prasanna et al., 2020; Chen et al., 2020; Liang et al., 2021) reported that they found winning-ticket subnetworks on these tasks, our finding did not contradict it. Their subnetworks were often actually a little worse than their dense networks, as well as we found. Chen et al. (2020) defined winning tickets as subnetworks with performances within one standard deviation from the dense networks. Prasanna et al. (2020) considered subnetworks with even 90% performance as winning tickets.

[6] For example, comparing BASELINE with RANDOM-LT of pruning ratio 20%, their average values of single/ensemble/difference are 91.38/91.93/+0.55 vs. 91.09/91.90/+0.81 on SST-2.

[7] This also happens to experiments with roberta-base while multi-ticket ensemble still works well on MRPC.

[8] Finding such a convenient diversity metric itself is still a challenge in the research community (Wu et al., 2021).

## Acknowledgments

We appreciate the helpful comments from the anonymous reviewers. This work was supported by JSPS KAKENHI Grant Number JP19H04162.

## References

Matti Aksela. 2003. Comparison of classifier selection methods for improving committee performance. In *Proceedings of the 4th International Conference on Multiple Classifier Systems*, MCS'03, page 84–93, Berlin, Heidelberg. Springer-Verlag.

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *CoRR*, abs/2012.09816.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? In *Proceedings of Second Machine Learning and Systems (MLSys 2020)*, pages 129–146.

Steven Cao, Victor Sanh, and Alexander M. Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pages 15834–15846. Curran Associates, Inc.

Rafael M. O. Cruz, Luiz G. Hafemann, Robert Sabourin, and George D. C. Cavalcanti. 2020. Deslib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research*, 21(8):1–5.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*.

Pedro Domingos. 1997. Why does bagging work? a bayesian account and its implications. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD 1997)*, page 155–158. AAAI Press.

Nikita Durasov, Timur Bagautdinov, Pierre Baque, and Pascal Fua. 2021. Masksembles for uncertainty estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13539–13548.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Giorgio Giacinto and Fabio Roli. 2001. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9):699–707.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew Mingbo Dai, and Dustin Tran. 2021. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*.

Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Stefan Lee, and Dhruv Batra. 2016. CVPR tutorial: Diversity meets deep networks - inference, ensemble learning, and applications.

Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207.

Esther Levin, Naftali Tishby, and Sara A. Solla. 1989. A statistical approach to learning and generalization in layered neural networks. In *Proceedings of the Second Annual Workshop on Computational Learning Theory (COLT 1989)*, page 245–260, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2021. Super tickets in pre-trained language models: From model compression to improving generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

*Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6524–6538, Online. Association for Computational Linguistics.

Y. Liu and X. Yao. 1999. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4970–4979. PMLR.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 3208–3229, Online. Association for Computational Linguistics.

Evani Radiya-Dixit and Xin Wang. 2020. How fine can fine-tuning be? learning efficient language models. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (ICML 2020)*, volume 108 of *Proceedings of Machine Learning Research*, pages 2435–2443. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alexandre Rame and Matthieu Cord. 2021. {DICE}: Diversity in deep ensembles via conditional redundancy adversarial estimation. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. 2022. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations (ICLR 2022)*.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training NLP models: A concise overview. *CoRR*, abs/2004.08900.

David B. Skalak. 1996. The sources of increased accuracy for two proposed boosting algorithms. In *In Proc. American Association for Arti Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*, pages 120–125.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, pages 38–45, Online. Association for Computational Linguistics.

Yanzhao Wu, Ling Liu, Zhongwei Xie, Ka-Ho Chow, and Wenqi Wei. 2021. Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16469–16477.

G. Udny Yule. 1900. On the association of attributes in statistics: With illustrations from the material of the childhood society, &c. *Philosophical Transactions of the Royal Society of London*, 194:257–319.

Zhilu Zhang, Vianne R. Gao, and Mert R. Sabuncu. 2021. Ex uno plures: Splitting one model into an ensemble of subnetworks.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an efficient alternative to finetuning for pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 2226–2241, Online. Association for Computational Linguistics.

## A The Setting of Fine-tuning

We follow the setting of Chen et al. (2020)'s implementation; epoch: 3, initial learning rate: 2e-5 with linear decay, maximum sequence length: 128, batch size: 32, dropout probability: 0.1. This is one of the most-used settings for finetuning a BERT; e.g., the example of finetuning in the Transformers library (Wolf et al., 2020) uses the setting[9].

We did not prune the embedding layer, following Chen et al. (2020); Prasanna et al. (2020). The coefficient of $L_1$ regularizer, $\tau$, is decayed using the same scheduler as the learning rate. We tuned it on MRPC and used it for other tasks.

## B The Learning Rate Scheduler of Chen et al. (2020)

Our implementation used in the experiments are derived from Chen et al. (2020)'s implementation[10]. However, we found a bug in Chen et al. (2020)'s implementation on GitHub. Thus, we fixed it and experimented with the correct version. In their implementation, the learning rate schedule did not follow the common setting and the description mentioned in the paper; *'We use standard implementations and hyperparameters [49]. Learning rate decays linearly from initial value to zero'*. Specifically, the learning rate with linear decay did not reach zero but was at significant levels even at the end of the finetuning. Our implementation corrected it so that it did reach zero as specified in their paper and in the common setting.

## C The Combinations of Ensembles

In the experiments, we first prepared twenty random seeds and split them into two groups, each of which trained ten models. For stabilizing the measurement of the result, we exhaustively evaluated all the possible combinations of ensembles (i.e., depending on the number of members, ${}_{10}\mathrm{C}_2$, ${}_{10}\mathrm{C}_3$, ${}_{10}\mathrm{C}_4$, ${}_{10}\mathrm{C}_5$ patterns, respectively) among the ten models for each group, and averaged the results with the two groups. The performance of the members is also averaged over all the seeds.

[9] https://github.com/huggingface/transformers/blob/7e406f4a65727baf8e22ae922f410224cde99ed6/examples/pytorch/text-classification/README.md#glue-tasks

[10] https://github.com/VITA-Group/BERT-Tickets

| | MRPC | | | STS-B | | |
|---|---|---|---|---|---|---|
| | single | ens. | diff. | single | ens. | diff. |
| BASELINE | 87.77 | 88.47 | +0.70 | 89.52 | 90.00 | +0.48 |
| (BAGGING) | 87.64 | 88.12 | +0.49 | 89.34 | 89.91 | +0.54 |
| BASE-LT | 87.72 | 88.25 | +0.53 | 89.71 | 90.07 | +0.36 |
| ACTIVE-LT | 87.39 | 88.51 | +1.12 | 88.46 | 89.50 | +1.04 |
| RANDOM-LT | 87.86 | 89.26 | +1.40 | 88.41 | 89.39 | +0.98 |

Table 3: The performances (single, ens.) and the improvements by ensembling (diff.) of RoBERTa-base models.

## D The Results with RoBERTa

We simply conducted supplementary experiments with RoBERTa (Liu et al., 2019) (robeta-base model), although optimal hyperparameters were not searched well. The results were similar to the cases of base-base-uncased. The patterns can be categorized into the three. First, multi-ticket ensembles worked well with roberta on MRPC, as shown in Table 3. Secondly, accurate winning-ticket subnetworks were not found on CoLA and QNLI. Although the effect of ensembleing was improved after pruning, each single model got worse and the final ensemble accuracy did not outperform the dense baseline. Thirdly, although accurate winning-ticket subnetworks were found on STS-B and SST-2, regularizations worsened single-model performances. While this case also improved the effect of ensembling, the final accuracy did not outperform the baseline. These experiments further emphasized the importance of development of more sophisticated pruning methods without sacrifice of model performances in the context of the lottery ticket hypothesis.

## E Related Work

Some concurrent studies also investigate the usage of subnetworks for ensembles. Gal and Ghahramani (2016) is a pioneer to use subnetwork ensemble. A trained neural network with dropout can infer with many different subnetworks, and their ensemble can be used for uncertainty estimation, which is called MC-dropout. Durasov et al. (2021) improved the efficiency of MC-dropout by exploring subnetworks. Zhang et al. (2021) (unpublished) experimented with an ensemble of subnetworks of different structures and initialization when trained from scratch, while the improvements possibly could be due to regularization of each single model. Havasi et al. (2021) is a similar but more elegant approach, which does not explicitly identify subnetworks. Instead, it trains a single dense model with training using multi-input multi-output inference; the optimization can implicitly find multiple disentangled subnetworks in the dense model during optimization from random initialization. These studies support our assumption that different subnetworks can improve ensemble by diversity.

Some other directions for introducing diversity exist, while most are unstable. Promising directions are to use entropy (Pang et al., 2019) or adversarial training (Rame and Cord, 2021). Although they required complex optimization processes, they improved the robustness or ensemble performance on small image recognition datasets.

Recently, concurrent work (Sellam et al., 2022; Tay et al., 2022) provide multiple BERT or T5 models pretrained from different seeds or configurations for investigation of seed or configuration dependency using large-scale computational resources. Further research with the models and such computational resources will be helpful for more solid comparison and analysis.

Note that no prior work tackled the problem of ensembles from a pre-trained model. Framing the problem is one of the contributions of this paper. Secondly, our multi-ticket ensemble based on random masking enables an independently parallelizable training while existing methods require a sequential processing or a grouped training procedure. Finally, multi-ticket ensemble can be combined with other methods, which can improve the total performance together.