Super Tickets in Pre-Trained Language Models: From Model Compression to Improving Generalization

Chen Liang[*1], **Simiao Zuo**[1], **Minshuo Chen**[1], **Haoming Jiang**[1],
Xiaodong Liu[2], **Pengcheng He**[3], **Tuo Zhao**[1], **Weizhu Chen**[3]
[1] Georgia Institute of Technology, [2] Microsoft Research, [3] Microsoft Azure AI
{cliang73, simiaozuo, mchen393, jianghm, tourzhao}@gatech.edu
{xiaodl,penhe,wzchen}@microsoft.com

Abstract

The Lottery Ticket Hypothesis suggests that an over-parametrized network consists of "lottery tickets", and training a certain collection of them (i.e., a subnetwork) can match the performance of the full model. In this paper, we study such a collection of tickets, which is referred to as "winning tickets", in extremely over-parametrized models, e.g., pre-trained language models. We observe that at certain compression ratios, the generalization performance of the winning tickets can not only match but also exceed that of the full model. In particular, we observe a phase transition phenomenon: As the compression ratio increases, generalization performance of the winning tickets first improves then deteriorates after a certain threshold. We refer to the tickets on the threshold as "super tickets". We further show that the phase transition is task and model dependent — as the model size becomes larger and the training data set becomes smaller, the transition becomes more pronounced. Our experiments on the GLUE benchmark show that the super tickets improve single task fine-tuning by 0.9 points on BERT-base and 1.0 points on BERT-large, in terms of task-average score. We also demonstrate that adaptively sharing the super tickets across tasks benefits multi-task learning[1].

1 Introduction

The Lottery Ticket Hypothesis (LTH, Frankle and Carbin (2018)) suggests that an over-parameterized network consists of "lottery tickets", and training a certain collection of them (i.e., a subnetwork) can 1) match the performance of the full model; and 2) outperform randomly sampled subnetworks of the same size (i.e., "random tickets"). The existence of such a collection of tickets, which is usually referred to as "winning tickets", indicates the potential of training a smaller network to achieve the full model's performance. LTH has been widely explored in across various fields of deep learning (Frankle et al., 2019; Zhou et al., 2019; You et al., 2019; Brix et al., 2020; Movva and Zhao, 2020; Girish et al., 2020).

Aside from training from scratch, such winning tickets have demonstrated their abilities to transfer across tasks and datasets (Morcos et al., 2019; Yu et al., 2019; Desai et al., 2019; Chen et al., 2020a). In natural language processing, Chen et al. (2020b); Prasanna et al. (2020) have shown existence of the winning tickets in pre-trained language models. These tickets can be identified when fine-tuning the pre-trained models on downstream tasks. As the pre-trained models are usually extremely over-parameterized (e.g., BERT Devlin et al. (2019), GPT-3 Brown et al. (2020), T5 Raffel et al. (2019)), previous works mainly focus on searching for a highly compressed subnetwork that matches the performance of the full model. However, behavior of the winning tickets in lightly compressed subnetworks is largely overlooked.

In this paper, we study the behavior of the winning tickets in pre-trained language models, with a particular focus on lightly compressed subnetworks. We observe that generalization performance of the winning tickets selected at appropriate compression ratios can not only match, but also exceed that of the full model. In particular, we observe a *phase transition* phenomenon (Figure 1): The test accuracy improves as the compression ratio grows until a certain threshold (Phase I); Passing the threshold, the accuracy deteriorates, yet is still better than that of the random tickets (Phase II). In Phase III, where the model is highly compressed,

[*] Work was done at Microsoft Azure AI.
[1] Our codes are available at https://github.com/cliang1453/super-structured-lottery-tickets.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6524–6538
August 1–6, 2021.

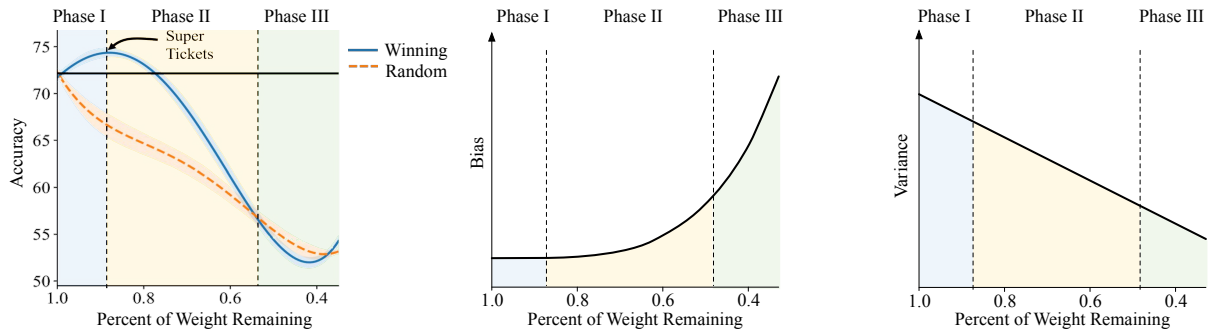

Figure 1: Illustrations of the phase transition phenomenon. *Left*: Generalization performance of the fine-tuned subnetworks (the same as Figure 4 in Section 5). *Middle and Right*: An interpretation of bias-variance trade-off.

training collapses. We refer to the set of winning tickets selected on that threshold as "super tickets".

We interpret the phase transition in the context of trade-offs between model bias and variance (Friedman et al., 2001, Chapter 7). It is well understood that an expressive model induces a small bias, and a large model induces a large variance. We classify the tickets into three categories: non-expressive tickets, lightly expressive tickets, and highly expressive tickets. The full model has a strong expressive power due to over-parameterization, so that its bias is small. Yet its variance is relatively large. In Phase I, by removing non-expressive tickets, variance of the selected subnetwork reduces, while model bias remains unchanged and the expressive power sustains. Accordingly, generalization performance improves. We enter Phase II by further increasing the compression ratio. Here lightly expressive tickets are pruned. Consequently, model variance continues to decrease. However, model bias increases and overturns the benefit of the reduced variance. Lastly for Phase III, in the highly compressed region, model bias becomes notoriously large and reduction of the variance pales. As a result, training breaks down and generalization performance drops significantly.

We conduct systematic experiments and analyses to understand the phase transition. Our experiments on multiple natural language understanding (NLU) tasks in the GLUE (Wang et al., 2018) benchmark show that the super tickets can be used to improve single task fine-tuning by 0.9 points over BERT-base (Devlin et al., 2019) and 1.0 points over BERT-large, in terms of task-average score. Moreover, our experiments show that the phase transition phenomenon is task and model dependent. It becomes more pronounced as a larger model is used to fit a task with less training data. In such a case, the set of super tickets forms a compressed network that exhibits a large performance gain.

The existence of super tickets suggests potential benefits to applications, such as Multi-task Learning (MTL). In MTL, different tasks require different capacities to achieve a balance between model bias and variance. However, existing methods do not specifically balance the bias and variance to accommodate each task. In fact, the fine-tuning performance on tasks with a small dataset is very sensitive to randomness. This suggests that model variance in these tasks are high due to over-parameterization. To reduce such variance, we propose a *tickets sharing* strategy. Specifically, for each task, we select a set of super tickets during single task fine-tuning. Then, we adaptively share these super tickets across tasks.

Our experiments show that tickets sharing improves MTL by 0.9 points over MT-DNN$_{\text{BASE}}$ (Liu et al., 2019) and 1.0 points over MT-DNN$_{\text{LARGE}}$, in terms of task-average score. Tickets sharing further benefits downstream fine-tuning of the multi-task model, and achieves a gain of 1.0 task-average score. In addition, the multi-task model obtained by such a sharing strategy exhibits lower sensitivity to randomness in downstream fine-tuning tasks, suggesting a reduction in variance.

We summarize our contributions as follows:

• Our result is the first to identify the phase transition phenomenon in pruning large neural language models.

• Our result is the first to show that pruning can improve the generalization when the models are lightly compressed, which has been overlooked by previous works. Our analysis paves the way for understanding the connection between model compression and generalization.

• Motivated by our observed phase transition,

we further propose a new pruning approach for multi-task fine-tuning of neural language models.

2 Background

We briefly introduce the Transformer architecture and the Lottery Ticket Hypothesis.

2.1 Transformer Architecture

The Transformer (Vaswani et al., 2017) encoder is composed of a stack of identical Transformer layers. Each layer consists of a multi-head attention module (MHA) followed by a feed-forward module (FFN), with a residual connection around each. The vanilla single-head attention operates as

$$\text{Att}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$

where $Q, K, V \in \mathbb{R}^{l\times d}$ are d-dimensional vector representations of l words in sequences of queries, keys and values. In MHA, the h-th attention head is parameterized by $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d\times d_h}$ as

$$\text{H}_h(\boldsymbol{q}, \boldsymbol{x}, W_h^{\{Q,K,V\}}) = \text{Att}(\boldsymbol{q}W_h^Q, \boldsymbol{x}W_h^K, \boldsymbol{x}W_h^V),$$

where $\boldsymbol{q} \in \mathbb{R}^{l\times d}$ and $\boldsymbol{x} \in \mathbb{R}^{l\times d}$ are the query and key/value vectors. In MHA, H independently parameterized attention heads are applied in parallel, and the outputs are aggregated by $W_h^O \in \mathbb{R}^{d_h\times d}$:

$$\text{MHA}(\boldsymbol{q}, \boldsymbol{x}) = \sum_h^H \text{H}_h(\boldsymbol{q}, \boldsymbol{x}, W_h^{\{Q,K,V\}})W_h^O.$$

Each FFN module contains a two-layer fully connected network. Given the input embedding $\boldsymbol{z}$, we let $\text{FFN}(\boldsymbol{z})$ denote the output of a FFN module.

2.2 Structured and Unstructured LTHs

LTH (Frankle and Carbin, 2018) has been widely explored in various applications of deep learning (Brix et al., 2020; Movva and Zhao, 2020; Girish et al., 2020). Most of existing results focus on finding unstructured winning tickets via iterative magnitude pruning and rewinding in randomly initialized networks (Frankle et al., 2019; Renda et al., 2020), where each ticket is a single parameter. Recent works further investigate learning dynamics of the tickets (Zhou et al., 2019; Frankle et al., 2020) and efficient methods to identify them (You et al., 2019; Savarese et al., 2020). Besides training from scratch, researchers also explore the existence of winning tickets under transfer learning regimes for over-parametrized pre-trained models across various tasks and datasets (Morcos et al., 2019; Yu et al., 2019; Desai et al., 2019; Chen et al., 2020a). For example, Chen et al. (2020b); Prasanna et al. (2020) have shown the existence of winning tickets when fine-tuning BERT on downstream tasks.

There is also a surge of research exploring whether certain structures, e.g., channels in convolutional layers and attention heads in Transformers, exhibit properties of the lottery tickets. Compared to unstructured tickets, training with structured tickets is memory efficient (Cao et al., 2019). Liu et al. (2018); Prasanna et al. (2020) suggest that there is no clear evidence that structured winning tickets exist in randomly initialized or pre-trained weights. Prasanna et al. (2020) observe that, in highly compressed BERT (e.g., the percent of weight remaining is around 50%), all tickets perform equally well. However, Prasanna et al. (2020) have not investigated the cases where the percent of weight remaining is over 50%.

3 Finding Super Tickets

We identify winning tickets in BERT through structured pruning of attention heads and feed-forward layers. Specifically, in each Transformer layer, we associate mask variables ξ_h to each attention head and ν to the FFN (Prasanna et al., 2020):

$$\text{MHA}(Q, \boldsymbol{x}) = \sum_h^H \xi_h \text{H}_h(Q, \boldsymbol{x}, W_h^{\{Q,K,V\}})W_h^O,$$
$$\text{FFN}(\boldsymbol{z}) = \nu\text{FFN}(\boldsymbol{z}).$$

Here, we set $\xi_h, \nu \in \{0, 1\}$, and a 0 value indicates that the corresponding structure is pruned.

We adopt importance score (Michel et al., 2019) as a gauge for pruning. In particular, the importance score is defined as the expected sensitivity of the model outputs with respect to the mask variables. Specifically, in each Transformer layer,

$$I_{\text{MHA}}^h = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_x}\left|\frac{\partial\mathcal{L}(\boldsymbol{x})}{\partial\xi_h}\right|,$$
$$I_{\text{FFN}} = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_x}\left|\frac{\partial\mathcal{L}(\boldsymbol{x})}{\partial\nu}\right|,$$

where $\mathcal{L}$ is a loss function and $\mathcal{D}_x$ is the data distribution. In practice, we compute the average over the training set. We apply a layer-wise ℓ_2 normalization on the importance scores of the attention heads (Molchanov et al., 2016; Michel et al., 2019).

The importance score is closely tied to expressive power. A low importance score indicates that the corresponding structure only has a small contribution towards the output. Such a structure has low expressive power. On the contrary, a large importance score implies high expressive power.

We compute the importance scores for all the mask variables in a single backward pass at the end of fine-tuning. We perform one-shot pruning of the same percent of heads and feed-forward layers with the lowest importance scores. We conduct pruning multiple times to obtain subnetworks, or winning tickets, at different compression ratios.

We adopt the weight rewinding technique in Renda et al. (2020): We reset the parameters of the winning tickets to their values in the pre-trained weights, and subsequently fine-tune the subnetwork with the original learning rate schedule. The super tickets are selected as the winning tickets with the best rewinding validation performance.

4 Multi-task Learning with Tickets Sharing

In multi-task learning, the shared model is highly over-parameterized to ensure a sufficient capacity for fitting individual tasks. Thus, the multi-task model inevitably exhibits task-dependent redundancy when being adapted to individual tasks. Such redundancy induces a large model variance.

We propose to mitigate the aforementioned model redundancy by identifying task-specific super tickets to accommodate each task's need. Specifically, when viewing an individual task in isolation, the super tickets can tailor the multi-task model to strike an appealing balance between the model bias and variance (recall from Section 3 that super tickets retain sufficient expressive power, yet keep the model variance low). Therefore, we expect that deploying super tickets can effectively tame the model redundancy for individual tasks.

Given the super tickets identified by each task, we exploit the multi-task information to reinforce fine-tuning. Specifically, we propose a *tickets sharing* algorithm to update the parameters of the multi-task model: For a certain network structure (e.g., an attention head), if it is identified as super tickets by multiple tasks, then its weights are jointly updated by these tasks; if it is only selected by one specific task, then its weights are updated by that task only; otherwise, its weights are completely pruned. See Figure 2 for an illustration.

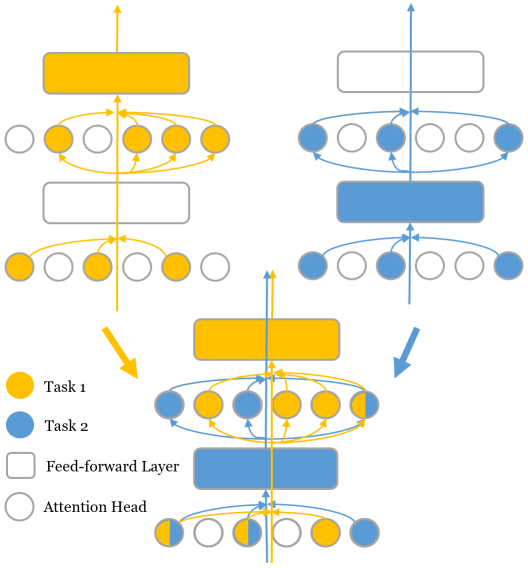

Figure 2: Illustration of tickets sharing.

In more detail, we denote the weight parameters in the multi-task model as θ. Suppose there are N tasks. For each task $i \in \{1, \ldots, N\}$, we denote $\Omega^i = \{\xi^i_{h,\ell}\}^{H,L}_{h=1,\ell=1} \bigcup \{\nu^i_\ell\}^{L}_{\ell=1}$ as the collection of the mask variables, where ℓ is the layer index and h is the head index. Then the parameters to be updated in task i are denoted as $\theta^i = M(\theta, \Omega^i)$, where $M(\cdot, \Omega^i)$ masks the pruned parameters according to Ω^i. We use stochastic gradient descent-type algorithms to update θ^i. Note that the task-shared and task-specific parameters are encoded by the mask variable Ω^i. The detailed algorithm is given in Algorithm 1.

Tickets sharing has two major difference compared to *Sparse Sharing* (Sun et al., 2020): 1) Sun et al. (2020) share winning tickets, while our strategy focuses on super tickets, which can better generalize and strike a sensible balance between model bias and variance. 2) In tickets sharing, tickets are structured and chosen from pre-trained weight parameters. It does not require *Multi-task Warmup*, which is indispensable in Sun et al. (2020) to stabilize the sharing among unstructured tickets selected from randomly initialized weight parameters.

5 Single Task Experiments

5.1 Data

General Language Understanding Evaluation (GLUE, Wang et al. (2018)) is a standard benchmark for evaluating model generalization performance. It contains nine NLU tasks, including question answering, sentiment analysis, text similarity

Algorithm 1 Tickets Sharing

Input: Pre-trained base model parameters θ. Number of tasks N. Mask variables $\{\Omega^i\}_{i=1}^N$. Loss functions $\{\mathcal{L}^i\}_{i=1}^N$. Dataset $D = \bigcup_{i=1}^N D_i$. Number of epochs $T_{\max}$.

for i in N **do**
  Initialize the super tickets for task i: $\theta^i = M(\theta, \Omega^i)$.
end for
for epoch in $1, \ldots, T_{\max}$ **do**
  Shuffle dataset D.
  for a minibatch b_i of task i in D **do**
    Compute Loss $\mathcal{L}^i(\theta^i)$.
    Compute gradient $\nabla_\theta \mathcal{L}^i(\theta^i)$.
    Update θ^i using SGD-type algorithm.
  end for
end for

and textual entailment. Details about the benchmark are deferred to Appendix A.1.1.

5.2 Models & Training

We fine-tune a pre-trained BERT model with task-specific data to obtain a single task model. We append a task-specific fully-connected layer to BERT as in Devlin et al. (2019).

• **ST-DNN$_{\text{BASE/LARGE}}$** is initialized with BERT-base/large followed by a task-specific layer.

• **SuperT$_{\text{BASE/LARGE}}$** is initialized with the chosen set of super tickets in BERT-base/large followed by a task-specific layer. Specifically, we prune BERT-base/large in unit of 10% heads and 10% feed-forward layers (FFN) at 8 different sparsity levels (10% heads and 10% FFN, 20% heads and 20% FFN, etc). Among them, the one with the best rewinding validation result is chosen as the set of super tickets. We randomly sample 10% GLUE development set for tickets selection.

Our implementation is based on the MT-DNN code base[3]. We use Adamax (Kingma and Ba, 2014) as our optimizer. We tune the learning rate in $\{5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$ and batch size in $\{8, 16, 32\}$. We train for a maximum of 6 epochs with early-stopping. All training details are summarized in Appendix A.1.2.

5.3 Generalization of the Super Tickets

We conduct 5 trails of pruning and rewinding experiments using different random seeds. Table 1 and 2 show the averaged evaluation results on the GLUE development and test sets, respectively. We remark that the gain of SuperT$_{\text{BASE/LARGE}}$ over ST-DNN$_{\text{BASE/LARGE}}$ is statistically significant. All the results[4] have passed a paired student t-test with p-values less than 0.05. More validation statistics are summarized in Appendix A.1.3.

Our results can be summarized as follows.

1) In all the tasks, SuperT consistently achieves better generalization than ST-DNN. The task-averaged improvement is around 0.9 over ST-DNN$_{\text{BASE}}$ and 1.0 over ST-DNN$_{\text{LARGE}}$.

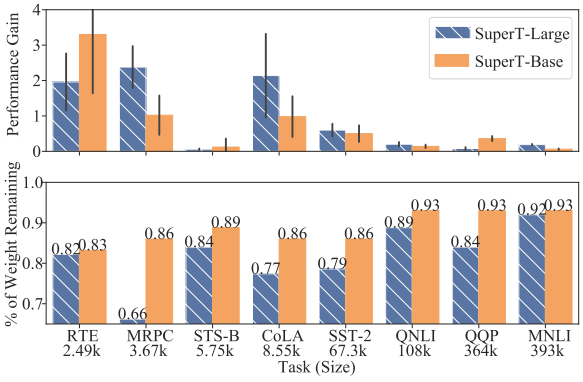

Figure 3: Single task fine-tuning validation results in different GLUE tasks. *Upper*: Performance Gain. *Lower*: Percent of weight remaining.

2) Performance gain of the super tickets is more significant in small tasks. For example, in Table 1, we obtain 3.3 points gain on RTE (2.5k data), but only 0.4/0.3 on QQP (364k data) in the SuperT$_{\text{BASE}}$ experiments. Furthermore, from Figure 3, note that the super tickets are more heavily compressed in small tasks, e.g., for SuperT$_{\text{BASE}}$, 83% weights remaining for RTE, but 93% for QQP. These observations suggest that for small tasks, model variance is large, and removing non-expressive tickets reduces variance and improves generalization. For large tasks, model variance is low, and all tickets are expressive to some extent.

3) Performance of the super tickets is related to model size. Switching from SuperT$_{\text{BASE}}$ to SuperT$_{\text{LARGE}}$, the percent of weights remaining shrinks uniformly across tasks, yet the generalization gains persist (Figure 3). This suggests that in large models, more non-expressive tickets can be pruned without performance degradation.

[3] https://github.com/namisan/mt-dnn

[4] Except for STS-B (SuperT$_{\text{BASE}}$, Table 1), where the p-value is 0.37.

	RTE Acc	MRPC Acc/F1	CoLA Mcc	SST Acc	STS-B P/S Corr	QNLI Acc	QQP Acc/F1	MNLI-m/mm Acc	Average Score	Average Compression
ST-DNN$_{\text{BASE}}$	69.2	86.2/90.4	57.8	92.9	89.7/89.2	91.2	90.9/88.0	**84.5**/84.4	82.8	100%
SuperT$_{\text{BASE}}$	**72.5**	**87.5/91.1**	**58.8**	**93.4**	**89.8/89.4**	**91.3**	**91.3/88.3**	**84.5/84.5**	**83.7**	86.8%
ST-DNN$_{\text{LARGE}}$	72.1	85.2/89.5	62.1	93.3	89.9/89.6	92.2	91.3/88.4	86.2/86.1	84.1	100%
SuperT$_{\text{LARGE}}$	**74.1**	**88.0/91.4**	**64.3**	**93.9**	**89.9/89.7**	**92.4**	**91.4/88.5**	**86.5/86.2**	**85.1**	81.7%

Table 1: Single task fine-tuning evaluation results on the GLUE development set. *ST-DNN* and *SuperT* results are the averaged score over 5 trails with different random seeds.

	RTE Acc	MRPC F1	CoLA Mcc	SST Acc	STS-B S Corr	QNLI Acc	QQP F1	MNLI-m/mm Acc	Average Score	Average Compression
ST-DNN$_{\text{BASE}}$	66.4	88.9	52.1	93.5	85.8	**90.5**	71.2	**84.6**/83.4	79.6	100%
SuperT$_{\text{BASE}}$	**69.6**	**89.4**	**54.3**	**94.1**	**86.2**	**90.5**	**71.3**	**84.6/83.8**	**80.4**	86.8%

Table 2: Single task fine-tuning test set results scored using the GLUE evaluation server[2]. Results of *ST-DNN$_{BASE}$* are from Devlin et al. (2019).

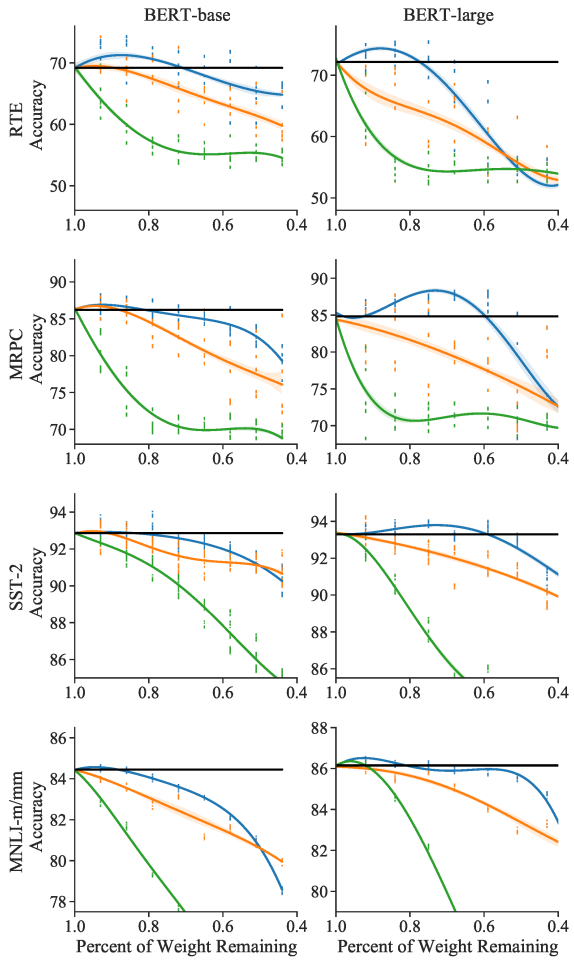

Figure 4: Single task fine-tuning evaluation results of the winning (blue), the random (orange), and the losing (green) tickets on the GLUE development set under various sparsity levels.

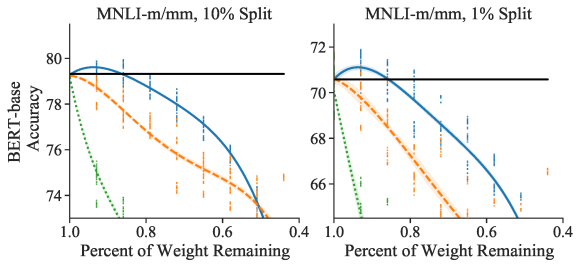

Figure 5: Phase transition under different randomly sampled training subsets. Note that the settings are the same as Figure 4 (bottom left), except the data size.

5.4 Phase Transition

Phase transitions are shown in Figure 4. We plot the evaluation results of the winning, the random, and the losing tickets under 8 sparsity levels using BERT-base and BERT-large. The winning tickets contain structures with the highest importance scores. The losing tickets are selected reversely, i.e., the structures with the lowest importance scores are selected, and high-importance structures are pruned. The random tickets are sampled uniformly across the network. We plot the averaged scores over 5 trails using different random seeds[5]. Phase transitions of all the GLUE tasks are in Appendix A.5.

We summarize our observations:

1) The winning tickets are indeed the "winners". In Phase I and early Phase II, the winning tickets perform better than the full model and the random tickets. This demonstrates the existence of struc-

[5]Except for MNLI, where we plot 3 trails as the there are less variance among trails.

	RTE Acc	MRPC Acc/F1	CoLA Mcc	SST Acc	STS-B P/S Corr	QNLI Acc	QQP Acc/F1	MNLI-m/mm Acc	Average Score	Average Compression
MT-DNN$_{\text{BASE}}$	79.0	80.6/86.2	54.0	92.2	86.2/86.4	90.5	90.6/87.4	84.6/84.2	82.4	100%
+ ST Fine-tuning	79.1	86.8/89.2	59.5	**93.6**	90.6/90.4	91.0	**91.6**/88.6	**85.3/85.0**	84.6	100%
Ticket-Share$_{\text{BASE}}$	81.2	87.0/90.5	52.0	92.7	87.7/87.5	91.0	90.7/87.5	84.5/84.1	83.3	92.9%
+ ST Fine-tuning	**83.0**	**89.2/91.6**	**59.7**	93.5	**91.1/91.0**	**91.9**	**91.6/88.7**	85.0/**85.0**	**85.6**	92.9%
MT-DNN$_{\text{LARGE}}$	83.0	85.2/89.4	56.2	93.5	87.2/86.9	92.2	91.2/88.1	86.5/86.0	84.4	100%
+ ST Fine-tuning	83.4	87.5/91.0	63.5	**94.3**	90.7/90.6	92.9	**91.9/89.2**	**87.1**/86.7	86.4	100%
Ticket-Share$_{\text{LARGE}}$	80.5	88.4/91.5	61.8	93.2	89.2/89.1	92.1	91.3/88.4	86.7/86.0	85.4	83.3%
+ ST Fine-tuning	**84.5**	**90.2/92.9**	**65.0**	94.1	**91.3/91.1**	**93.0**	**91.9**/89.1	87.0/**86.8**	**87.1**	83.3%

Table 3: Multi-task Learning evaluation results on the GLUE development set. Results of *MT-DNN$_{BASE/LARGE}$* with and without ST Fine-tuning are from Liu et al. (2020).

tured winning tickets in lightly compressed BERT models, which Prasanna et al. (2020) overlook.

2) Phase transition is pronounced over different tasks and models. Accuracy of the winning tickets increases up till a certain compression ratio (Phase I); Passing the threshold, the accuracy decreases (Phase II), until its value intersects with that of the random tickets (Phase III). Note that Phase III agrees with the observations in Prasanna et al. (2020). Accuracy of the random tickets decreases in each phase. This suggests that model bias increases steadily, since tickets with both low and high expressive power are discarded. Accuracy of the losing tickets drops significantly even in Phase I, suggesting that model bias increases drastically as highly expressive tickets are pruned.

3) Phase transition is more pronounced in large models and small tasks. For example, in Figure 4, the phase transition is more noticeable in BERT-large than in BERT-base, and is more pronounced in RTE (2.5k) and MRPC (3.7k) than in SST (67k) and MNLI (393k). The phenomenon becomes more significant for the same task when we only use a part of the data, e.g., Figure 5 vs. Figure 4 (bottom left).

6 Multi-task Learning Experiments

6.1 Model & Training

We adopt the MT-DNN architecture proposed in Liu et al. (2020). The MT-DNN model consists of a set of task-shared layers followed by a set of task-specific layers. The task-shared layers take in the input sequence embedding, and generate shared semantic representations by optimizing multi-task objectives. Our implementation is based on the MT-DNN code base. We follow the same training settings in Liu et al. (2020) for multi-task learning, and in Section 5.2 for downstream fine-tuning. More details are summarized in Appendix A.2.

- **MT-DNN$_{\text{BASE/LARGE}}$.** An MT-DNN model refined through multi-task learning, with task-shared layers initialized by pre-trained BERT-base/large.
- **MT-DNN$_{\text{BASE/LARGE}}$ + ST Fine-tuning.** A single task model obtained by further fine-tuning MT-DNN on an individual downstream task.
- **Ticket-Share$_{\text{BASE/LARGE}}$.** An MT-DNN model refined through the ticket sharing strategy, with task-shared layers initialized by the union of the super tickets in pre-trained BERT-base/large.
- **Ticket-Share$_{\text{BASE/LARGE}}$ + ST Fine-tuning.** A fine-tuned single-task **Ticket-Share** model.

6.2 Experimental Results

Table 3 summarizes experimental results. The fine-tuning results are averaged over 5 trails using different random seeds. We have several observations:

1) Ticket-Share$_{\text{BASE}}$ and Ticket-Share$_{\text{LARGE}}$ achieve 0.9 and 1.0 gain in task-average score over MT-DNN$_{\text{BASE}}$ and MT-DNN$_{\text{LARGE}}$, respectively. In some small tasks (RTE, MRPC), Ticket-Share achieves better or on par results compared to MT-DNN+Fine-tuning. This suggests that by balancing the bias and variance for different tasks, the multi-task model's variance is reduced. In large tasks (QQP, QNLI and MNLI), Ticket-Share behaves equally well with the full model. This is because task-shared information is kept during pruning and still benefits multi-task learning.

2) Ticket-Share$_{\text{BASE}}$+Fine-tuning and Ticket-Share$_{\text{LARGE}}$+Fine-tuning achieve 1.0 and 0.7 gains in task-average score over MT-DNN$_{\text{BASE}}$+Fine-tuning and MT-DNN$_{\text{LARGE}}$+Fine-tuning, respectively. This suggests that reducing the variance in the multi-task model benefits fine-tuning downstream tasks.

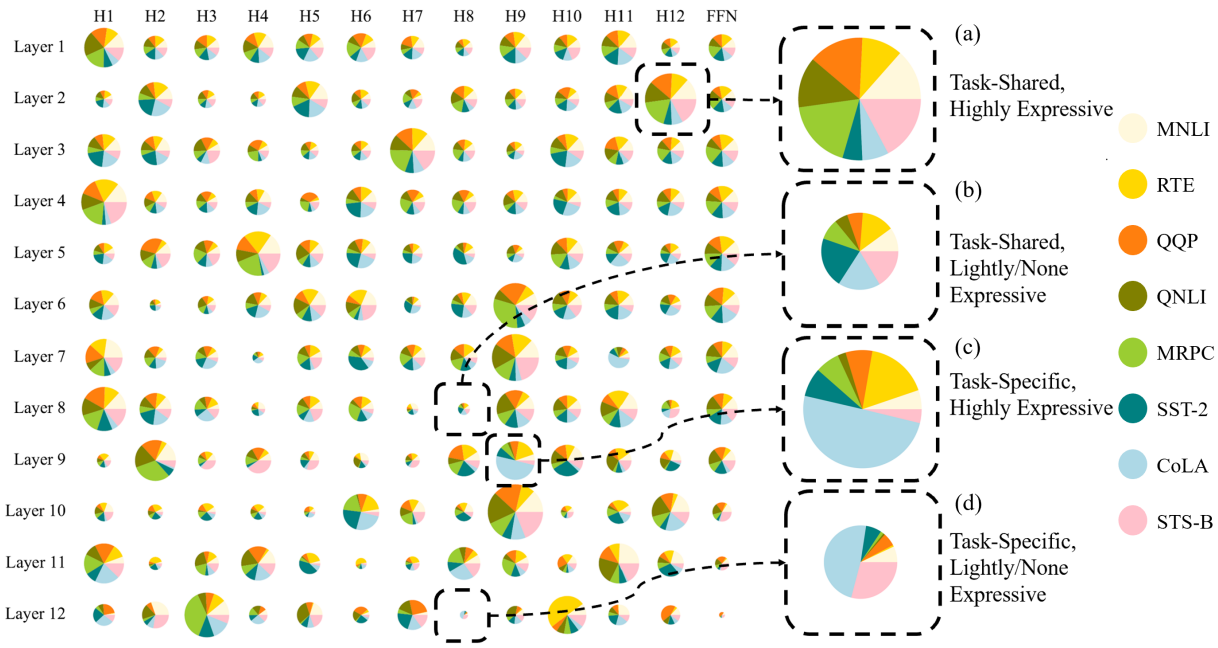

Figure 6: Illustration of tickets importance across tasks. Each ticket is represented by a pie chart. The size of a pie indicates the *Ticket Importance*, where a larger pie suggests the ticket exhibits higher expressivity. Each task is represented by a color. The share of a color indicates the *Task Share*, where a even share suggests the ticket exhibits equal expressivity in all tasks.

Model	0.1%	1%	10%	100%
SNLI (Dev Acc%)				
# Training Data	549	5493	54k	549k
MNLI-ST-DNN$_{\text{BASE}}$	82.1	85.1	88.4	90.7
MNLI-SuperT$_{\text{BASE}}$	**82.9**	**85.5**	**88.8**	**91.4**
MT-DNN$_{\text{BASE}}$	82.1	85.2	88.4	91.1
Ticket-Share$_{\text{BASE}}$	**83.3**	**85.8**	**88.9**	**91.5**
SciTail (Dev Acc%)				
# Training Data	23	235	23k	235k
MNLI-ST-DNN$_{\text{BASE}}$	80.6	88.8	92.0	95.7
MNLI-SuperT$_{\text{BASE}}$	**82.9**	**89.8**	**92.8**	**96.2**
MT-DNN$_{\text{BASE}}$	81.9	88.3	91.1	95.7
Ticket-Share$_{\text{BASE}}$	**83.1**	**90.1**	**93.5**	**96.5**

Table 4: Domain adaptation evaluation results on SNLI and SciTail development set. Results of *MT-DNN$_{BASE}$* are from Liu et al. (2020).

7 Domain Adaptation

To demonstrate that super tickets can quickly generalize to new tasks/domains, we conduct few-shot domain adaptation on out-of-domain NLI datasets.

7.1 Data & Training

We briefly introduce the target domain datasets. The data and training details are summarized in Appendix A.3.1 and A.3.2, respectively.

SNLI. The Stanford Natural Language Inference dataset (Bowman et al., 2015) is one of the most widely used entailment dataset for NLI. It contains 570k sentence pairs, where the premises are drawn from the captions of the Flickr30 corpus and hypotheses are manually annotated.

SciTail is a textual entailment dataset derived from a science question answering (SciQ) dataset (Khot et al., 2018). The hypotheses are created from science questions, rendering SciTail challenging.

7.2 Experimental Results

We consider domain adaptation on both single task and multi-task super tickets. Specifically, we adapt SuperT$_{\text{BASE}}$ and ST-DNN$_{\text{BASE}}$ from MNLI to SNLI/SciTail, and adapt the shared embeddings generated by Ticket-Share$_{\text{BASE}}$ and by MT-DNN$_{\text{BASE}}$ to SNLI/SciTail. We adapt these models to 0.1%, 1%, 10% and 100% SNLI/SciTail training sets[6], and evaluate the transferred models on SNLI/SciTail development sets. Table 4 shows the domain adaptation evaluation results. As we can see, SuperT and Ticket-Share can better adapt to SNLI/SciTail than ST-DNN and MT-DNN, especially under the few shot setting.

[6]We use the subsets released in MT-DNN code base.

8 Analysis

Sensitivity to Random Seed. To better demonstrate that training with super tickets effectively reduces model variance, we evaluate models' sensitivity to changes in random seeds during single task fine-tuning and multi-task downstream fine-tuning. In particular, we investigate fitting small tasks with highly over-parametrized models (variance is often large in these models, see Section 5 and 6). As shown in Table 5, $\text{SuperT}_{\text{LARGE}}$ and $\text{Ticket-Share}_{\text{LARGE}}$ induce much smaller standard deviation in validation results. Experimental details and further analyses are deferred to Appendix A.4.

	RTE	MRPC	CoLA	STS-B	SST-2
$\text{ST-DNN}_{\text{LARGE}}$	1.17	0.61	1.32	0.16	0.17
$\text{SuperT}_{\text{LARGE}}$	0.72	0.20	0.97	0.07	0.16
$\text{MT-DNN}_{\text{LARGE}}$	1.43	0.78	1.14	0.15	0.18
$\text{Ticket Share}_{\text{LARGE}}$	0.99	0.67	0.81	0.08	0.16

Table 5: Standard deviation of tasks in GLUE (dev) over 5 different random seeds.

Tickets Importance Across Tasks. We analyze the importance score of each ticket computed in different GLUE tasks. For each ticket, we compute the importance score averaged over tasks as the *Ticket Importance*, and the proportion of the task-specific importance score out of the sum of all tasks' scores as the *Task Share*, as illustrated in Figure 6.

We observe that many tickets exhibit almost equal *Task Share*s for over 5 out of 8 tasks (Figure 6(a)(b)). While these tickets contribute to the knowledge sharing in the majority of tasks, they are considered non-expressive for tasks such as SST-2 (see Figure 6(a)(c)(d)). This explains why SST-2 benefits little from tickets sharing. Furthermore, a small number of tickets are dominated by a single task, e.g., CoLA (Figure 6(c)), or dominated jointly by two tasks, e.g., CoLA and STS-B (Figure 6(d)). This suggests that some tickets only learn task-specific knowledge, and the two tasks may share certain task-specific knowledge.

9 Discussion

Structured Lottery Tickets. LTH hypothesizes that a subset of unstructured parameters can be trained to match the full model's performance. Instead, we question whether a subset of structured weight matrices, e.g., FFN layers and attention heads, can also be trained to match the full model's performance. This question is more practically important than the unstructured one: training and inference on structured matrices are better optimized for hardware acceleration. Our results give a positive answer to this question, while previous works show that the structured tickets do not exist in highly compressed models (Prasanna et al., 2020).

Searching Better Generalized Super Tickets. We select winning tickets according to the sensitivity of the model outputs with respect to the mask variables of each structure (Michel et al., 2019; Prasanna et al., 2020), as this measure is closely tied to the structure's expressive power (Section 3). In addition, we conduct an one-shot pruning for computational simplicity. We leave other importance measures and pruning schedules, which may help identifying better generalized super tickets, for future works (Voita et al., 2019; Behnke and Heafield, 2020; Wang et al., 2019; Fan et al., 2019; Zhou et al., 2020; Sajjad et al., 2020).

Searching Super Tickets Efficiently. Determining the compression ratio of the super tickets requires rewinding models at multiple sparsity levels. To leverage super tickets in practice, a potential direction of research is to find heuristics to determine this ratio prior or early-on in training. We leave this for future works.

10 Conclusion

We study the behaviors of the structured lottery tickets in pre-trained BERT. We observe that the generalization performance of the winning tickets exhibits a phase transition phenomenon, suggesting pruning can improve generalization when models are lightly compressed. Based on the observation, we further propose a tickets sharing strategy to improve multi-task fine-tuning. Our analysis paves the way for understanding the connection between model compression and generalization.

Broader Impact

This paper studies the behavior of the structured lottery tickets in pre-trained language models. Our investigation neither introduces any social/ethical bias to the model nor amplifies any bias in the data. We do not foresee any direct social consequences or ethical issues. Furthermore, our proposed method improves performance through model compression, rendering it energy efficient.

References

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online. Association for Computational Linguistics.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09)*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Christopher Brix, Parnia Bahar, and Hermann Ney. 2020. Successfully applying the stabilized lottery ticket hypothesis to the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3909–3915.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Shijie Cao, Chen Zhang, Zhuliang Yao, Wencong Xiao, Lanshun Nie, Dechen Zhan, Yunxin Liu, Ming Wu, and Lintao Zhang. 2019. Efficient and effective sparse lstm on fpga with bank-balanced sparsity. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 63–72.

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.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. 2020a. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. *arXiv preprint arXiv:2012.06908*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020b. The lottery ticket hypothesis for pre-trained bert networks. *arXiv preprint arXiv:2007.12223*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg. Springer-Verlag.

Shrey Desai, Hongyuan Zhan, and Ahmed Aly. 2019. Evaluating lottery tickets under distributional shifts. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 153–162.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. 2019. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*.

Jonathan Frankle, David J Schwab, and Ari S Morcos. 2020. The early phase of neural network training. *arXiv preprint arXiv:2002.10365*.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. 2001. *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Sharath Girish, Shishira R Maiya, Kamal Gupta, Hao Chen, Larry Davis, and Abhinav Shrivastava. 2020. The lottery ticket hypothesis for object recognition. *arXiv preprint arXiv:2012.04643*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, et al. 2020. The microsoft toolkit of multi-task deep neural networks for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 118–126.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*.

Ari S Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *arXiv preprint arXiv:1906.02773*.

Rajiv Movva and Jason Y Zhao. 2020. Dissecting lottery ticket transformers: Structural and behavioral study of sparse neural machine translation. *arXiv preprint arXiv:2009.13270*.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. *arXiv preprint arXiv:2003.02389*.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.

Pedro Savarese, Hugo Silva, and Michael Maire. 2020. Winning the lottery with continuous sparsification. *Advances in Neural Information Processing Systems*, 33.

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*, pages 1631–1642.

Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8936–8943.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*.

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. Association for Computational Linguistics.

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G Baraniuk, Zhangyang Wang, and Yingyan Lin. 2019. Drawing early-bird tickets: Towards more efficient training of deep networks. *arXiv preprint arXiv:1909.11957*.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S Morcos. 2019. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. 2019. Deconstructing lottery tickets: Zeros, signs, and the supermask. *arXiv preprint arXiv:1905.01067*.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2020. Scheduled drophead: A regularization method for transformer models. *arXiv preprint arXiv:2004.13342*.

A Appendix

A.1 Single Task Experiments

A.1.1 Data

GLUE. GLUE is a collection of nine NLU tasks. The benchmark includes question answering (Rajpurkar et al., 2016), linguistic acceptability (CoLA, Warstadt et al. 2019), sentiment analysis (SST, Socher et al. 2013), text similarity (STS-B, Cer et al. 2017), paraphrase detection (MRPC, Dolan and Brockett 2005), and natural language inference (RTE & MNLI, Dagan et al. 2006; Bar-Haim et al. 2006; Giampiccolo et al. 2007; Bentivogli et al. 2009; Williams et al. 2018) tasks. Details of the GLUE benchmark, including tasks, statistics, and evaluation metrics, are summarized in Table 9.

A.1.2 Training

We use Adamax as the optimizer. A linear learning rate decay schedule with warm-up over 0.1 is used. We apply a gradient norm clipping of 1. We set the dropout rate of all task specific layers as 0.1, except 0.3 for MNLI and 0.05 for CoLA. All the texts were tokenized using wordpieces, and were chopped to spans no longer than 512 tokens. All experiments are conducted on Nvidia V100 GPUs.

A.1.3 Evaluation Results Statistics

We conduct 5 sets of experiments on different random seeds. Each set of experiment consists of fine-tuning, pruning, and rewinding at 8 sparsity levels. For results on GLUE dev set (Table 1), we report the average score of super tickets rewinding results over 5 sets of experiments. The standard deviation of the results is shown in Table 6. The statistics of the percent of weight remaining in the selected super tickets are shown in Table 7.

For results on GLUE test set (Table 2), as the evaluation server sets an limit on submission times, we only evaluate the test prediction under a single random seed that gives the best task-average validation results.

A.2 Multi-task Learning Experiments

A.2.1 Multi-task Model Training

We adopt the MT-DNN code base and adopt the exact optimization settings in Liu et al. (2020). We use Adamax as our optimizer with a learning rate of 5×10^{-5} and a batch size of 32. We train for a maximum number of epochs of 5 with early stopping. A linear learning rate decay schedule with warm-up over 0.1 was used. The dropout rate of all the task specific layers is set to be 0.1, except 0.3 for MNLI and 0.05 for CoLa. We clipped the gradient norm within 1. All the texts were tokenized using wordpieces, and were chopped to spans no longer than 512 tokens.

Worth mentioning, the task-specific super tickets used in Ticket Share are all selected during the case where a matched learning rate (i.e., 5×10^{-5}) is used in single task fine-tuning. We empirically find that, rewinding the super tickets selected under a matched optimization settings usually outperforms those selected under a mismatched settings (i.e. using two different learning rates in single-task fine-tuning and rewinding/multi-task learning). This agrees with previous observation in literature of Lottery Ticket Hypothesis, which shows that unstructured winning tickets are not only related to its weight initialization, but also model optimization path.

A.2.2 Multi-task Model Downstream Fine-tuning

We follow the exact optimization setting as in Section 5.2 and in Section A.1.2, except we choose learning rate in $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$, and choose the dropout rate of all task specific layers in $\{0.05, 0.1, 0.2, 0.3\}$.

A.3 Domain Adaptation Experiments

A.3.1 Data

SNLI. is one of the most widely used entailment dataset for NLI.

SciTail involves assessing whether a given premise entails a given hypothesis. In contrast to other entailment datasets, the hypotheses in SciTail is created from science questions. These sentences are linguistically challenging. The corresponding answer candidates and premises come from relevant web sentences. The lexical similarity of premise and hypothesis is often high, making SciTail particularly challenging.

Details of the SNLI and SciTail, including tasks, statistics, and evaluation metrics, are summarized in Table 9.

A.3.2 Training

For single task model domain adaptation from MNLI to SNLI/SciTail, we follow the exact optimization setting as in Section 5.2 and in Section A.1.2, except we choose the learning rate in $\{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}\}$.

	RTE	MRPC	CoLA	STS-B	SST-2	QNLI	QQP	MNLI
SuperT$_{\text{BASE}}$	0.91	0.74	1.51	0.49	0.50	0.10	0.08	0.04
SuperT$_{\text{LARGE}}$	0.72	0.20	0.97	0.07	0.16	0.07	0.11	0.02

Table 6: Standard deviation of the evaluation results on GLUE development set over 5 different random seeds.

	RTE	MRPC	CoLA	STS-B	SST-2	QNLI	QQP	MNLI
SuperT$_{\text{BASE}}$ (Mean)	0.83	0.86	0.89	0.86	0.93	0.93	0.93	0.93
SuperT$_{\text{BASE}}$ (Std Dev)	0.07	0.08	0.04	0.06	0.07	0.00	0.00	0.00
SuperT$_{\text{LARGE}}$ (Mean)	0.82	0.66	0.84	0.77	0.79	0.90	0.84	0.92
SuperT$_{\text{LARGE}}$ (Std Dev)	0.04	0.04	0.00	0.10	0.05	0.03	0.00	0.00

Table 7: Statistics of the percent of weight remaining of the selected super tickets over 5 different random seeds.

A.4 Sensitivity Analysis

A.4.1 Randomness Analysis

For single task experiments in Table 5, we vary the random seeds only and keep all other hyper-parameters fixed. We present the standard deviation of the validation results over 5 trails rewinding experiments. For multi-task downstream fine-tuning experiments, we present the standard deviation of the validation results over 5 trails, each result averaged over learning rates in $\{5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$. This is because the downstream fine-tuning performance is more sensitive to hyper-parameters.

A.4.2 Hyper-parameter Analysis

We further analyze the sensitivity of Ticket Share$_{\text{LARGE}}$ model to changes in hyper-parameters in downstream fine-tuning in some GLUE tasks. We vary the learning rate in $\{5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$ and keep all other hyper-parameter fixed. Table 8 shows the standard deviation of the validation results over different learning rates, each result averaged over 5 different random seeds. As can be seen, Task Share$_{\text{LARGE}}$ exhibits stronger robustness to changes in learning rate in downstream fine-tuning.

	RTE	MRPC	CoLA	STS-B	SST-2
MT-DNN$_{\text{LARGE}}$	1.26	0.86	1.05	0.42	0.26
Ticket Share$_{\text{LARGE}}$	0.44	0.58	0.61	0.36	0.25

Table 8: Standard deviation of some tasks in GLUE (dev) over 3 different learning rates.

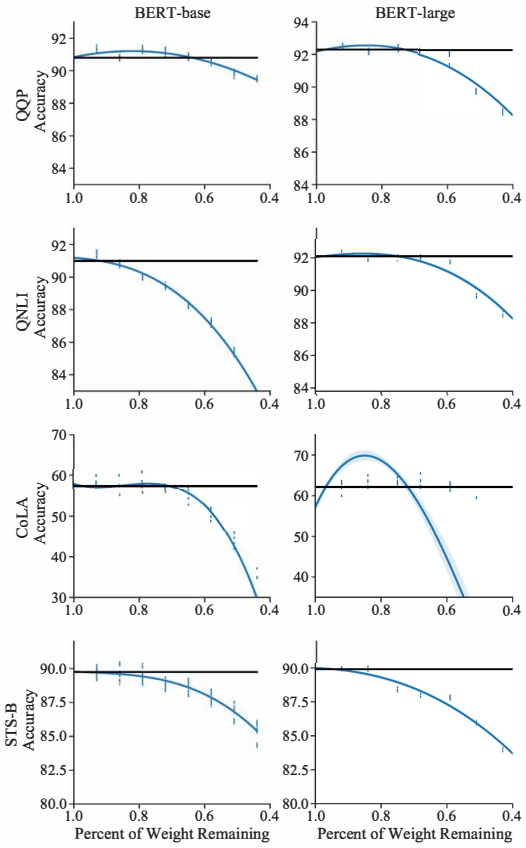

Figure 7: Single task fine-tuning evaluation results of the winning tickets on the GLUE development set under various sparsity levels.

A.5 Phase Transition on GLUE Tasks

Figure 7 shows the phase transition plots on winning tickets on GLUE tasks absent from Figure 4. All experimental settings conform to Figure 4.

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
Single-Sentence Classification (GLUE)						
CoLA	Acceptability	8.5k	1k	1k	2	Matthews corr
SST	Sentiment	67k	872	1.8k	2	Accuracy
Pairwise Text Classification (GLUE)						
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy
Text Similarity (GLUE)						
STS-B	Similarity	7k	1.5k	1.4k	1	Pearson/Spearman corr
Pairwise Text Classification						
SNLI	NLI	549k	9.8k	9.8k	3	Accuracy
SciTail	NLI	23.5k	1.3k	2.1k	2	Accuracy

Table 9: Summary of the GLUE benchmark, SNLI and SciTail.