# Finding Sparse Structures for Domain Specific Neural Machine Translation

**Jianze Liang,** [1,2]* **Chengqi Zhao,** [2] **Mingxuan Wang,** [2] **Xipeng Qiu,** [1] **Lei Li** [2]
[1] School of Computer Science, Fudan University, Shanghai, China
[2] ByteDance AI Lab, China
{jzliang18, xpqiu}@fudan.edu.cn, {zhaochengqi.d, wangmingxuan.89, lileilab}@bytedance.com

## Abstract

Neural machine translation often adopts the fine-tuning approach to adapt to specific domains. However, nonrestricted fine-tuning can easily degrade on the general domain and over-fit to the target domain. To mitigate the issue, we propose PRUNE-TUNE, a novel domain adaptation method via gradual pruning. It learns tiny domain-specific sub-networks during fine-tuning on new domains. PRUNE-TUNE alleviates the over-fitting and the degradation problem without model modification. Furthermore, PRUNE-TUNE is able to sequentially learn a single network with multiple disjoint domain-specific sub-networks for multiple domains. Empirical experiment results show that PRUNE-TUNE outperforms several strong competitors in the target domain test set without sacrificing the quality on the general domain in both single and multi-domain settings. The source code and data are available at https://github.com/ohlionel/Prune-Tune.

## Introduction

Neural Machine Translation (NMT) yields state-of-the-art translation performance when a large number of parallel sentences are available (Kalchbrenner and Blunsom 2013; Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2015; Vaswani et al. 2017). However, there are many language pairs lacking parallel corpora. It is also observed that NMT does not perform well in specific domains where the domain-specific corpora are limited, such as medical domain (Koehn and Schroeder 2007; Axelrod, He, and Gao 2011; Freitag and Al-Onaizan 2016; Chu and Wang 2018). There is huge need to produce high-quality domain-specific machine translation systems whereas general purpose MT has limited performance.

Domain adaptation for NMT has been studied extensively. These work can be grouped into two categories: data-centric and model fine-tuning (Chu and Wang 2018). Data-centric methods focus on selecting or generating target domain data from general domain corpora, which is effective and well explored (Axelrod, He, and Gao 2011; Chinea-Ríos, Peris, and Casacuberta 2017; Zeng et al. 2019). In this paper, we focus on the second thread: model fine-tuning. Fine-tuning is very common in domain adaptation, which first trains a base model on the general domain data and then fine-tunes it on each target domain (Luong and Manning 2015; Chen et al. 2017; Gu, Feng, and Liu 2019; Saunders et al. 2019). However, non-restricted fine-tuning requires very careful hyper-parameter tuning, and is prone to over-fitting on the target domain as well as forgetting on the general domain. To tackle these issues, researchers have proposed several constructive approaches, with the view to limiting the size or plasticity of parameters in the fine-tuning stage, which can be roughly divided into two categories: regularization and partial-tuning strategy. Regularization methods often integrate extra training objectives to prevent parameters from large deviations, such as model output regularization (Khayrallah et al. 2018), elastic weight consolidation (EWC) (Thompson et al. 2019). Regularization methods, which impose arbitrary global constraints on parameter updates, may further restrict the adaptive process of the network, especially when domain-specific corpora are scarce. Partial-tuning methods either freeze several sub-layers of the network and fine-tune the others (Thompson et al. 2018), or integrate domain-specific adapters into the network (Bapna and Firat 2019; Vilar 2018). By only fine-tuning the domain-specific part of the model, they can alleviate the over-fitting and forgetting problem in fine-tuning. However, the structure designed to adapting is usually hand-crafted, which relies on experienced experts and the adapter brings additional parameters. Therefore, a more adaptive, scalable, and parameter-efficient approach for domain adaptation is very valuable and worth well studying.

In this paper, we propose PRUNE-TUNE, a novel domain adaptation method via adaptive structure pruning. Our motivation is inspired from Continual Learning (Parisi et al. 2019; Kirkpatrick et al. 2017; Mallya and Lazebnik 2018; Mallya, Davis, and Lazebnik 2018; Hung et al. 2019; Lee, Cho, and Kang 2020) and *the lottery hypothesis* that a randomly-initialized, dense neural network contains a sub-network which can match the test accuracy of the original network after training for at most the same number of iterations (Frankle and Carbin 2019). We therefore suppose that multiple machine translation models for different domains can share different sparse sub-networks within a single neural network. Specifically, we first apply a standard pruning

*The work was done while JL was an intern at ByteDance AI Lab.

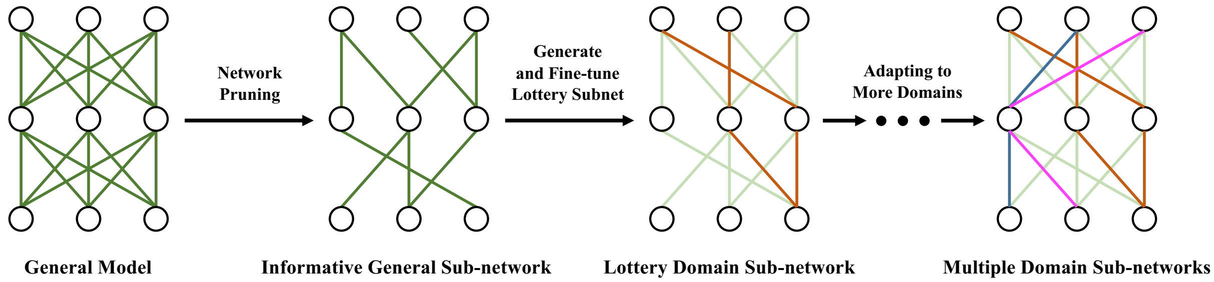


Figure 1: Illustration of domain adaptation from the general domain to the target domain with PRUNE-TUNE. c)→ d) demonstrates our proposed PRUNE-TUNE is capable of adapting to multiple domains.

technique to automatically uncover the sub-network from a well-trained NMT model in the general domain. The sub-network is capable of reducing the parameter without compromising accuracy. Therefore, it has the potential to keep as much general information as possible. Then we freeze this informative sparse network and leave the unnecessary parameters unfixed for the target domain, which enables our approach to be parameter efficient, and eases the scalability of the approach to more domains. The capacity of these non-fixed parameters can be tuned to match the requirements of the target domain, while keeping the parameters of the general domain. Our method successfully circumvents catastrophic forgetting problem (Kirkpatrick et al. 2017) and retains the quality on the general domain. As the benefits of the flexible design, PRUNE-TUNE can be easily extended to other transfer learning problems, such as multilingual machine translation.

We summarize our main contribution as follows:

- We propose PRUNE-TUNE, which enables generating domain-specific sub-networks via gradual pruning and potentially circumvents the notorious *catastrophic forgetting* problems in domain adaptation.
- We conduct extensive experiments to evaluate PRUNE-TUNE and demonstrate that PRUNE-TUNE outperforms the strong competitors both in the general and target domain with big margins. On domain adaptation benchmarks for EN→DE, PRUNE-TUNE outperforms several strong competitors including Fine-tuning, EWC, Model Distillation, Layer Freeze and Adapter in target domain test set without the loss of general domain performance.
- We extend PRUNE-TUNE to multi-domain experiments on EN→DE and ZH→EN, which shows the possibilities of training a single model to serve different domains without performance degradation.

## Background

### Neural Machine Translation

Given an source sentence $\mathbf{x} = \{x_1, \ldots, x_n\}$ and its translation $\mathbf{y} = \{y_1, \ldots, y_m\}$, Neural Machine Translation directly models the conditional probability of target sentence over source sentence:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{i=1}^{m} P(y_i|\mathbf{x}, y_{<i};\theta), \tag{1}$$

where $\theta$ denotes the parameters of the model. For a parallel training dataset $D = \{\mathbf{x}^j, \mathbf{y}^j\}_{j=1}^{N}$, $\theta$ is optimized to maximum the log-likelihood:

$$\arg\max_{\theta} \sum_{j=1}^{N} \log P(\mathbf{y}^j|\mathbf{x}^j;\theta). \tag{2}$$

### Fine-tuning for Domain Adaptation

Model fine-tuning on the target domain is the most natural approach for domain adaptation. Assume we have a well trained NMT model $\mathcal{F}(\cdot;\theta)$ and a dataset $D_I = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{N_I}$ of a new domain. We can simply apply fine-tuning to adapt the model to the new domain, that is, we continue training the model to optimize $\theta$ on $D_I$:

$$\arg\max_{\theta} \sum_{i=1}^{N_I} \log P(\mathbf{y}^i|\mathbf{x}^i;\theta). \tag{3}$$

As discussed in Introduction, fine-tuning on all model parameters $\theta$ often leads to over-fitting on the new domain as well as forgetting on the general domain. So apart from regularization approaches, it is effective to introduce domain-specific part of models to alleviate these problems. There are two typical kinds of methods: layer freeze and adapter.

**Layer freeze** approaches regard the top layer, denoted as $\theta_L$, of model as the domain-specific parameters while the rest parameters $\theta_{l<L}$ are kept fixed. The training object of layer freeze is:

$$\arg\max_{\theta_L} \sum_{i=1}^{N_I} \log P(\mathbf{y}^i|\boldsymbol{H}_{L-1};\theta_L), \tag{4}$$

where $\boldsymbol{H}_{L-1} = \mathcal{F}(\mathbf{x}^i;\theta_{l<L})$ indicates the output of the $(L-1)$-th layer of the model.

**Adapter** methods integrate an additional module $\theta_A$ into the network. The additional module can be a fully-connection layer, a self-attention layer or their combinations. Finally we fine-tune only on the domain-specific part $\theta_A$ and the training objective is as follow:

$$\arg\max_{\theta_A} \sum_{i=1}^{N_I} \log P(\mathbf{y}^i|\boldsymbol{H}_L;\theta_A), \tag{5}$$

where $\boldsymbol{H}_L = \mathcal{F}(\mathbf{x}^i;\theta)$.

As shown in equation (4) and (5), domain-specific parameters only interact with the output of general model, i.e. $\mathcal{F}(\cdot;\theta)$. We suppose interaction more with the general model would achieve much better performance.

## Approach

As many studies show, a great proportion of parameters in the network are redundant (Frankle and Carbin 2019; Zhu and Gupta 2018; Liu et al. 2019). Pruning such parameters causes minor or even no degradation in the task. Zhu and Gupta (2018) show that the dynamic and sparse sub-network after pruning is expressive and outperforms the dense network with the equivalent size of parameters. Therefore, it is possible to make use of such redundancy for domain adaptation.

Given a well trained general model, our approach consists of the following steps (see Figure 1):

1. Find and freeze the most informative parameters of the general domain and leave unnecessary parameters for the target domain
2. Uncover the lottery sub-networks from the free parameters for a specific domain
3. Tune the lottery sub-networks for the specific domain
4. Repeat the 2-3 steps for multi-domain adaptation

### Finding the Informative Parameters for General Domain

Pruning has proven to be effective for keeping the informative parameters and eliminating unnecessary ones for neural networks (LeCun, Denker, and Solla 1990; Li et al. 2017; Han et al. 2015; Zhu and Gupta 2018). Without loss of generality, we employ a simple and effective *Gradual Pruning* approach to find the most informative parameters for the general domain (Zhu and Gupta 2018). The method gradually prunes the model to reach the target sparsity by reducing low magnitude parameters every 100 training steps. Explicitly, we trim parameters to the target sparsity in each layer. Between pruning steps, the model is trained on the general dataset to recover its performance in the sub-network. Though NMT is one of the most complicated tasks in deep learning, our empirical study on pruning sparsity shows that up to 50% parameters in a Transformer big model are not necessary and can be pruned with a performance drop less than 0.6 BLEU (see Figure 2). In this way, we can keep the general NMT model intact as an informative sub-network of the original model. To keep consistent generalization ability provided from the original sub-network, we freeze the parameters of the informative sub-network during domain adaptation process.

The left unnecessary weights throughout the network provide the possibility of generating a sparse lottery sub-network that can exactly match the test accuracy of the domain-specific model. As the lottery sub-network keeps most of the general domain information, fine-tuning the unnecessary weights can potentially outperform the full fine-tuning approach. Particularly, the sparsity rate is very flexible which can be changed to meet the requirements of various scenarios. In general, a low sparsity rate is suitable for simple domain adaptation tasks, while high sparsity works better for complicated domain or multiple domain adaptation tasks.

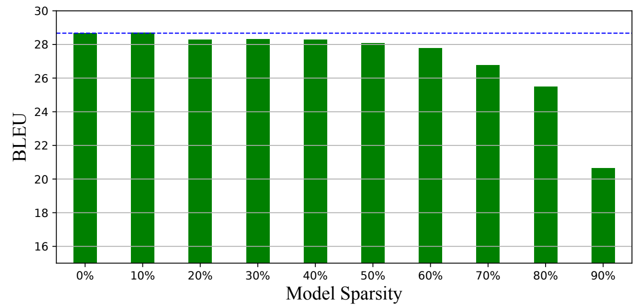


Figure 2: BLEU scores of pruned Transformer models with different sparsity (percentage of pruned parameters) on WMT14 En→De. Notice that even with 50% of the original parameters, the resulting model still achieves nearly the same translation performance as the original full Transformer.

### Lottery Sub-network Generation for Specific Domain

It is not necessarily needed to fine-tune all the free parameters for a specific domain, especially for multi-domain adaptation tasks that require parameter efficient sub-networks for different domains. As the extracted informative sub-network already has a strong capacity, we suppose that a few additional parameters may be enough for the target domain adaptation. The most challenging problem is to automatically uncover the best sparse structure for the specific domain within. And we call this sparse structure as *lottery sub-network*. The challenge is essentially a network architecture search problem (NAS) to learn domain-specific sub-network, which is very costly. For simplicity, we apply an iterative pruning method again as an effective way to learn the lottery sub-network.

Specifically, we fine-tune the free parameters on the target domain data for a few steps as warm-up training, then apply pruning to obtain the domain-specific structure. The generated structure is then fixed as the lottery domain-specific sub-network for further fine-tuning.

### Fine-tuning of Domain-Specific Sub-network

We introduce a mask matrix over all parameters in the network which indicates the sub-network for each domain with different domain identification. Each parameter of the network belongs to only one specific domain, and can not be updated by learning of other domains.

For single domain adaptation, we adapt the general domain to the target domain by training on the combined parameters of the general informative sub-network and the domain-specific lottery sub-network. For multiple domain adaptation, we iteratively repeat this process based on the general model. It is rather flexible as we do not require data from all domains simultaneously. Particularly, with the partition of parameters, we can adapt a new domain only from helpful domains. Supposes that we have successfully trained a multi-domain system supporting three different domains: news, law, biology. While our goal is to adapt to a new medical domain, it is capable of incorporating both the general and biology domain as source domains for the medical domain.

PRUNE-TUNE shares different domain sub-network in a single transform model with domain-specific masks. Given the source sentence and the corresponding domain identification, a binary domain mask will be applied to the unified model to support decoding with only the learned sparse sub-network. The mask matrix makes the system rather flexible for practical application or extends to a new domain.

| Direction | Corpus | Train | Dev. | Test |
|---|---|---|---|---|
| EN→DE | WMT14 | 3.9M | 3000 | 3003 |
| | IWSLT14 | 170k | 6750 | 1305 |
| | EMEA | 587k | 500 | 1000 |
| | Novel | 50k | 1015 | 1031 |
| ZH→EN | WMT19 | 20M | 3000 | 3981 |
| | Laws | 220k | 800 | 456 |
| | Thesis | 300k | 800 | 625 |
| | Subtitles | 300k | 800 | 598 |
| | Education | 449K | 800 | 791 |
| | News | 449K | 800 | 1500 |
| | Spoken | 219k | 800 | 456 |

Table 1: Datasets statistic for En→De and Zh→En tasks.

## Experiment

We conducted experiments on both single domain adaptation and multiple domain adaptation to show the effectiveness and flexibility of PRUNE-TUNE.

### Dataset

To evaluate our model in single domain adaptation, we conducted experiments on English to German translation, where the training corpora for the general domain were from WMT14 news translation task. And we used newstest2013 and newstest2014 as our validation and test set respectively. The general domain model trained on WMT14 EN→DE was then individually adapted to three distinct target domains: TED talks, biomedicine, and novel. For TED talks, we used IWSLT14 as training corpus, dev2010, and tst2014 as the validation and test set respectively. For the biomedicine domain, we evaluated on EMEA News Crawl dataset[1]. As there were no official validation and test set for EMEA, we used Khresmoi Medical Summary Translation Test Data 2.0[2]. For novel domain, we used a book dataset from OPUS[3] (Tiedemann 2012). We randomly selected several chapters from *Jane Eyre* as our validation set and *The Metamorphosis* as the test set.

We extended PRUNE-TUNE to multi-domain adaptation on English to German and Chinese to English translation. For ZH→EN, we used the training corpora from WMT19 ZH→EN translation task as the general domain data. We selected 6 target domain datasets from from UM-Corpus[4] (Tian et al. 2014).

Table 1 lists the statistics of all datasets mentioned above.

### Setup

For EN→DE data preprocessing, we tokenized data using *sentencepiece* (Kudo and Richardson 2018), with a jointly learned vocabulary of size 32,768. For ZH→EN, we applied jieba and moses tokenizer to Chinese and English side respectively. Then we encoded sentences using byte pair encoding (BPE) (Sennrich, Haddow, and Birch 2016b) with 32k merge operations separately. We implemented our models on recently the state-of-the-are translation model, Transformer (Vaswani et al. 2017) and we followed the big setting, including 6 layers for both encoder and decoders. The embedding dimension was 1,024 and the size of ffn hidden units was 4,096. The attention head was set to 16 for both self-attention and cross-attention. We used Adam optimizer (Kingma and Ba 2015) with the same schedule algorithm as Vaswani et al. (2017). All models were trained with a global batch size of 32,768 on NVIDIA Tesla V100 GPUs. During inference, we used a beam width of 4 for both EN→DE and ZH→EN and we set the length penalty to 0.6 for EN→DE, 1.0 for ZH→EN.

The evaluation metric for all our experiments is tokenized BLEU (Papineni et al. 2002) using *multi-bleu.perl*[5].

### Domain Adaptation on Single Lottery Sub-network

We used a lottery sub-network with 10% sparsity and conducted domain adaptation experiments on EN→DE. The 10% free parameters were tuned to fit each target domain. During inference, our model can recover the capability of the general domain by simply masking these domain-specific parameters. We compared our model with several strong baselines and effective models:

- **General domain model**: The model was trained using only parallel data from the general domain.
- **Target domain model**: The model was trained using only the target domain data.
- **Mixed domain model**: All general domain and target domain data were mixed to train the model.
- **Fine-tuning** (Luong and Manning 2015): We continued to train the general domain model on target domain data with the training step unchanged. The empirical study shows it performs better than resetting the training step to 0.

[1] https://ufal.mff.cuni.cz/ufal_medical_corpus

[2] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2122

[3] http://opus.nlpl.eu/

[4] http://nlp2ct.cis.umac.mo/um-corpus

[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

| Model | IWSLT (190k) | | EMEA (587k) | | Novel (50k) | | #Tuning Params |
|---|---|---|---|---|---|---|---|
| | general | target | general | target | general | target | |
| Target Domain Model | 11.4 | 24.0 | 3.1 | 23.9 | 2.7 | 12.3 | 273M |
| Mixed Domain Model | 27.9 | 31.3 | 27.9 | **32.0** | 27.9 | 21.2 | 273M |
| General Domain Model | 28.7 | 28.5 | 28.7 | 28.4 | 28.7 | 14.5 | 273M |
| + Fine-tuning (Luong and Manning 2015) | 27.0 | 31.5 | 17.1 | 29.7 | 12.1 | 23.4 | 273M |
| + EWC-regularized (Thompson et al. 2019) | 28.0 | 31.5 | 27.1 | 30.5 | 23.5 | 23.1 | 273M |
| + Model Distillation (Khayrallah et al. 2018) | 26.3 | 31.5 | 16.3 | 30.0 | 11.6 | 23.1 | 273M |
| + Layer Freeze (Thompson et al. 2018) | 28.6 | 31.3 | 26.9 | 29.8 | 23.0 | 23.0 | 29M |
| + Adapter (Bapna and Firat 2019) | 27.0 | 31.6 | 26.7 | 30.1 | 19.8 | **24.3** | 13M |
| PRUNE-TUNE Model | **28.8** | **31.9** | **28.9** | 30.6 | **28.8** | **24.3** | 27M |

Table 2: BLEU scores of single domain adaptation on EN→DE. All models share the same Transformer-big setting. Notice that PRUNE-TUNE improves the translation performance on the specific domains while maintaining the general domain performance.

- **EWC-regularized model** (Thompson et al. 2019): EWC (Kirkpatrick et al. 2017) is a popular algorithm in Continual Learning (Parisi et al. 2019), which applies elastic consolidation to each parameter during gradient updates. The EWC-regularized model prevents the parameters from large deviations.
- **Model Distillation** (Khayrallah et al. 2018): We employed an auxiliary loss during fine-tuning to prevent the target domain model's output from differing too much from the original general domain model's output.
- **Layer Freeze** (Thompson et al. 2018): We froze all model layers except for the top layers of both the encoder and decoder, then fine-tuned the top layers on the target domain data.
- **Adapter** (Bapna and Firat 2019): We stacked adapters on each transformer block of both encoder and decoder as proposed by Bapna and Firat (2019), and fine-tuned the adapters only.

Our proposed PRUNE-TUNE outperforms fine-tuning and other baselines as shown in Table 2. In three distinct domains with varying corpus size, our approach achieves competitive performance with less training parameters. Moreover, our model is able to serve both general or target domain machine translation without any performance compromise in a unified model, simply via a domain mask. Figure 3 demonstrates that our approach effectively alleviates the serious over-fitting problem that fine-tuning often suffers from. To conclude, PRUNE-TUNE enjoys the following advantages:

- PRUNE-TUNE is very effective for the target domain adaptation. We attribute this to the adaptive pruning of the lottery sub-network. With little modification of a sub-network, PRUNE-TUNE significantly outperforms *Layer Freeze* and *adapter* with pre-defined sub-network fine-tuning, which shows the benefits of dynamic structure finding.
- Clearly, PRUNE-TUNE is firmly capable of keeping the translation performance in the general domain. After fine-tuning on the novel domain, PRUNE-TUNE even surpasses the second fine-tuning competitor by 5 BLEU score in the general domain.
- PRUNE-TUNE is robust when compared to the fine-tuning baseline, which suffers from the over-fitting challenges and requires very careful checkpoints choices.

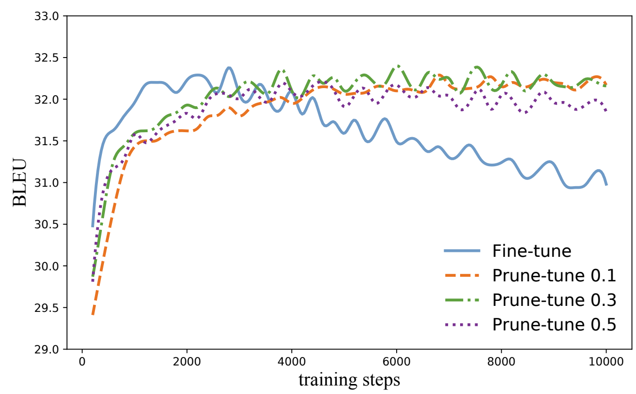


Figure 3: BLEU scores of fine-tuning and our proposed PRUNE-TUNE with 10%, 30%, 50% sparsity when adapting to IWSLT14 EN→DE. Notice the plain fine-tuning will degrade in the end on the target domain, while PRUNE-TUNE steadily improves.

## Sequential Domain Adaptation

We conducted multi-domain adaptation experiments on EN→DE and ZH→EN to demonstrate the unique sequential learning ability of our approach.

We first trained general models on EN→DE and ZH→EN, and then gradually pruned them to reach 50% sparsity. We find it empirically that 50% is a sparsity with no significant performance drop and enough redundant parameters. Different from single domain adaptation, we fixed embedding layers and layer normalization parameters to avoid sharing parameters across multiple domains. In these experiments, we adopt the general models to target domains sequentially. For each target domain:

| Model | Input domain | #M | WMT14 (**W**) | IWSLT (**I**) | EMEA (**E**) | Novel (**N**) |
|---|---|---|---|---|---|---|
| Mixed Domain Model | **W**, **I**, **E**, **N** | 1 | 27.9 | 31.3 | 32.0 | 21.2 |
| General Domain Model | **W** | 1 | 28.7 | 28.5 | 28.4 | 14.5 |
| + Fine-tuning | **I, E, N** | 3 | N/A | 31.5 | 29.7 | 23.4 |
| Single PRUNE-TUNE Model | **W, I, E, N** | 3 | N/A | 31.9 | 30.6 | 24.3 |
| Sequential PRUNE-TUNE Model | #1 **W** | 1 | 28.4 | N/A | N/A | N/A |
| | #2 + **I** | | 28.4 | 31.9 | N/A | N/A |
| | #3 + **E** | | 28.4 | 31.9 | 30.1 | N/A |
| | #4 + **N** | | 28.4 | 31.9 | 30.1 | 23.6 |

Table 3: BLEU scores of sequential domain adaptation on EN→DE. #M denotes the number of required models. W, I, E, N refer to dataset WMT14, IWSLT, EMEA, Novel, respectively. In our sequential PRUNE-TUNE Model, general domain occupied 50% parameters, and each target domain occupied 10%. Notice that sequential PRUNE-TUNE obtains a single model with best performance on all domains except EMEA.

| Model | #M | Laws | Thesis | Subtitles | Education | News | Spoken | Avg. |
|---|---|---|---|---|---|---|---|---|
| Mixed Domain Model | 1 | 47.4 | 15.6 | 17 | 31.4 | 21.2 | 16.7 | 24.9 |
| General Domain Model | 1 | 44.9 | 13.8 | 16.1 | 30.8 | 21.4 | 16.7 | 23.9 |
| + Fine-tuning | 6 | 55.9 | 17.9 | 20.8 | 29.2 | 22.1 | 14.8 | 26.7 |
| Sequential PRUNE-TUNE Model | 1 | 50.3 | 16.2 | 17.2 | 31.2 | 21.3 | 14.6 | 25.1 |

Table 4: BLEU scores of sequential domain adaptation on ZH→EN. #M denotes the number of required models. In our Sequential PRUNE-TUNE Model, general domain occupied 50% parameters, and each target domain occupied 5%. Notice that Sequential PRUNE-TUNE is the best performing single model for all domains.

1. Firstly, we applied warm-up training and *Gradual Pruning* to generate a suitable lottery domain sub-network.
2. Secondly, We simply adapt the general domain learned before to the current domain by including the general sub-network as frozen parameters.
3. Finally, we fine-tune the lottery sub-network of the domain.

We adapted 3 target domains on EN→DE, and 6 target domains on ZH→EN.

**Result** In Table 3, we report performance in EN→DE experiment. Within a single model, our approach can learn new domains in sequence and outperforms several baselines. Specifically, it outperforms Mixed Domain Model which requires all domain data simultaneously, and Fine-tuning which requires multiple models. Moreover, PRUNE-TUNE can learn the current target domain while retaining the performance in previously learned domains, because lottery domain sub-networks are separate.

ZH→EN experiment result in Table 4 also demonstrates that our approach is effective and flexible for more domains.

## Analysis

In this section, we revisit our approach to reveal more details and explain the effectiveness of the proposed PRUNE-TUNE.

| Pruning Rate | WMT | IWSLT | EMEA | Novel |
|---|---|---|---|---|
| 10% | **28.7** | 32.3 | **30.6** | **24.3** |
| 30% | 28.3 | **32.4** | 30.3 | 23.9 |
| 50% | 28.1 | 32.2 | 29.5 | 23.6 |
| 70% | 26.8 | 31.8 | 28.9 | 23.1 |

Table 5: BLEU scores of different pruning rate for PRUNE-TUNE. Only 10% of parameters for fine-tuning is able to achieve the best performance.

### Robustness of PRUNE-TUNE

We are convinced that the over-fitting problem seriously affects the robustness of fine-tuning. As shown in figure 3, fine-tuning reaches the best performance at the early step, and then starts to decline, while our method yields stable performance. When the target data is scarce, domain adaptation by unrestricted fine-tuning will rapidly over-fit to the target domain, forgetting the generalization ability from the general model. Our proposed PRUNE-TUNE is a more robust method as we integrate a frozen informative sub-network within the model, which provides generalized information consistently.

### Less Pruning Improves Performance

Since we can prune the model to different sparsity, we evaluate the single domain adaptation performance on general models with different sparsity. As shown in Table 5, do-

| Adaptation Order | EMEA |
|---|---|
| 1 | 30.3 |
| 2 | 30.1 |
| 3 | 30 |

Table 6: BLEU scores of different adaptation order for sequential domain adaptation.

| Target Domain Params(%) | IWSLT | EMEA | Novel |
|---|---|---|---|
| 1% | 31.7 | 29.4 | 22.3 |
| 5% | 31.8 | 30.1 | 23 |
| 10% | 31.9 | 30.1 | 23.6 |

Table 7: BLEU scores of different scale of target domain-specific parameters for sequential domain adaptation.

main adaptation on low sparsity achieves better performance mainly due to better knowledge preservation of the general domain. It also indicates that a few parameters are enough for single domain adaptation. As the pruning goes further, the high sparsity model is doomed to degrade on the general domain, which affects the subsequent domain adaptation. However, the performance gap between low sparsity PRUNE-TUNE models is relatively small.

## PRUNE-TUNE is Very Effective for Low-resource Domain Adaptation

To evaluate the performance of our approach on varying amounts of target domain data, we experimented on the EMEA dataset with different fractions of training data. We extract 1%, 3%, 10%, 30% and 100% of the original EMEA training set. We compare with full fine-tuning using 10% sparsity PRUNE-TUNE model on different fractions of EMEA dataset. As the results are shown in Figure 4, our approach significantly outperforms fine-tuning for each fraction. Especially for extremely small 1% fraction, which consists of 5.7K sentences, our proposed approach improves the performance over the general model by 0.7 BLEU, while fine-tuning leads to a 3.3 BLEU drop. With fractions less than 30%, fine-tuning can not improve the target bio domain, but brings damage to the general model. In the contrast, our approach does not harm the general domain ability, and can make the most of the few training data to improve the target domain. It indicates that our proposed approach is suitable for low resource domain adaptation, which is common and valuable in practice.

## Sequential PRUNE-TUNE is Capable for Numerous Domains

We conducted experiments to explore the limit of sequential multi-domain adaptation with PRUNE-TUNE. We first evaluated the influence of the learning order of the EMEA dataset. As shown in Table 6, there is only a minor gap of BLEU score between different learning order. We also conducted experiment on EN→DE with different scale of target

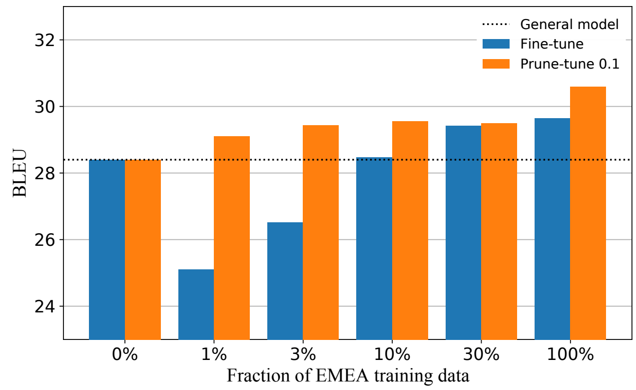


Figure 4: Fine-tuning with different domain-specific corpus. PRUNE-TUNE improves the baseline at different scales, while full fine-tuning suffers from over-fitting.

domain-specific parameters. As shown in Table 7, 5% of parameters is sufficient for most domains, and even 1% of parameters yields comparable performance. Actually, PRUNE-TUNE has the potential to adapt to dozens of domains.

# Related Work

## Domain Adaptation

Domain adaptation has been widely investigated in recent years. In Machine Translation, the fine-tuning based approach is the most relevant to our work. Fine-tune is the conventional way for domain adaptation (Luong and Manning 2015; Sennrich, Haddow, and Birch 2016a; Freitag and Al-Onaizan 2016; Chu, Dabre, and Kurohashi 2017). Many studies try to address the shortcoming of Fine-tune. Thompson et al. (2018) freeze selected modules of the general network. Adapters is introduced for parameter efficiency (Bapna and Firat 2019; Vilar 2018). Khayrallah et al. (2018) explore regularization techniques to avoid overfitting. Thompson et al. (2019) employ EWC (Kirkpatrick et al. 2017) to alleviate the catastrophic forgetting problem in domain adaptation. Zhang et al. (2020) re-initialize parameters from some layer for few-sample BERT fine-tuning. Wuebker, Simianer, and DeNero (2018) introduce sparse offset from the general model parameters for every domain, sharing the similar idea of our proposed method. The key difference is that PRUNE-TUNE provides a dynamic parameter adaptation method, which is parameter efficient and potentially makes the most of general domain information for the target domain.

Another research line for domain adaptation is data selection and data mixing, both being concerned with how to sample examples to train an MT model with a strong focus on a specific domain (Axelrod, He, and Gao 2011; Chinea-Ríos, Peris, and Casacuberta 2017; Zeng et al. 2019; Wang et al. 2020), while PRUNE-TUNE focused on the training model which can complement with the data-driven methods perfectly.

## Continual Learning

The main idea of our approach is originated from the Continual Learning community (Parisi et al. 2019; Kirkpatrick et al. 2017; Mallya and Lazebnik 2018; Mallya, Davis, and Lazebnik 2018; Hung et al. 2019; Lee, Cho, and Kang 2020), as they all try to alleviate the *catastrophic forgetting* problem. Mallya and Lazebnik (2018); Mallya, Davis, and Lazebnik (2018); Hung et al. (2019) learn separate sub-networks for multiple tasks in computer vision, which inspires us with PRUNE-TUNE for machine translation domain adaptation.

## Model Pruning

Our approach is also inspired by many studies of sparse networks (Frankle and Carbin 2019; Zhu and Gupta 2018; Liu et al. 2019; Masana et al. 2017). Frankle and Carbin (2019); Liu et al. (2019) reevaluate unstructured network pruning to highlight the importance of sparse network structure. Zhu and Gupta (2018) introduce advanced pruning technique to compress the model. Sun et al. (2020) learn sparse sharing architecture for multi-task learning. Hung et al. (2019) introduce compact parameter sub-network for continual learning. Different from these work, PRUNE-TUNE aims at finding the best sparse structure for a specific domain based on an NMT model trained on large scale general domain data. Model pruning is an effective method for our approach.

# Conclusion and Future Work

In this work, we propose PRUNE-TUNE, an effective way for adapting neural machine translation models which first generates an informative sub-network for the general domain via gradual pruning and then fine-tunes the unnecessary parameters for the target domain. By doing so, PRUNE-TUNE is able to retain as much general information as possible and alleviate the *catastrophic forgetting* problems. Experiments show that the proposed PRUNE-TUNE outperforms fine-tuning and several strong baselines and it is shown to be much more robust compared to fine-tuning due to the complete retainment of the general information. Beyond that, PRUNE-TUNE can be extended to adapting multiple domains by iteratively pruning and tuning, which is naturally suitable for multi-lingual scenario. We leave the multi-lingual problem as our future work.

# Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. We would also like to thank Zehui Lin, Yang Wei, Danqing Wang, Zewei Sun, and Kexin Huang for their useful suggestions and help with experiments. This work was supported by the National Key Research and Development Program of China (No. 2017YFB1002104).

# References

Axelrod, A.; He, X.; and Gao, J. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 355–362.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bapna, A.; and Firat, O. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1538–1548. Hong Kong, China: Association for Computational Linguistics.

Chen, B.; Cherry, C.; Foster, G.; and Larkin, S. 2017. Cost Weighting for Neural Machine Translation Domain Adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, 40–46. Vancouver: Association for Computational Linguistics.

Chinea-Ríos, M.; Peris, Á.; and Casacuberta, F. 2017. Adapting Neural Machine Translation with Parallel Synthetic Data. In *Proceedings of the Second Conference on Machine Translation*, 138–147. Copenhagen, Denmark: Association for Computational Linguistics.

Chu, C.; Dabre, R.; and Kurohashi, S. 2017. An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 385–391. Vancouver, Canada: Association for Computational Linguistics.

Chu, C.; and Wang, R. 2018. A Survey of Domain Adaptation for Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1304–1319. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Freitag, M.; and Al-Onaizan, Y. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897* .

Gu, S.; Feng, Y.; and Liu, Q. 2019. Improving Domain Adaptation Translation with Domain Invariant and Specific Information. 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)*, 3081–3091. Minneapolis, Minnesota: Association for Computational Linguistics.

Han, S.; Pool, J.; Tran, J.; and Dally, W. J. 2015. Learning both Weights and Connections for Efficient Neural Network. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 1135–1143.

Hung, S. C. Y.; Tu, C.; Wu, C.; Chen, C.; Chan, Y.; and Chen, C. 2019. Compacting, Picking and Growing for Unforgetting Continual Learning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 13647–13657.

Kalchbrenner, N.; and Blunsom, P. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1700–1709. Seattle, Washington, USA: Association for Computational Linguistics.

Khayrallah, H.; Thompson, B.; Duh, K.; and Koehn, P. 2018. Regularized Training Objective for Continued Training for Domain Adaptation in Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 36–44. Melbourne, Australia: Association for Computational Linguistics.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114(13): 3521–3526.

Koehn, P.; and Schroeder, J. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 224–227. Prague, Czech Republic: Association for Computational Linguistics.

Kudo, T.; and Richardson, J. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. Brussels, Belgium: Association for Computational Linguistics.

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1990. Optimal brain damage. In *Advances in neural information processing systems*, 598–605.

Lee, C.; Cho, K.; and Kang, W. 2020. Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2017. Pruning Filters for Efficient ConvNets. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; and Darrell, T. 2019. Rethinking the Value of Network Pruning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Luong, M.-T.; and Manning, C. D. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, 76–79.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 67–82.

Mallya, A.; and Lazebnik, S. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 7765–7773. IEEE Computer Society.

Masana, M.; van de Weijer, J.; Herranz, L.; Bagdanov, A. D.; and Álvarez, J. M. 2017. Domain-Adaptive Deep Network Compression. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 4299–4307. IEEE Computer Society.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113: 54–71.

Saunders, D.; Stahlberg, F.; de Gispert, A.; and Byrne, B. 2019. Domain Adaptive Inference for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 222–228. Florence, Italy: Association for Computational Linguistics.

Sennrich, R.; Haddow, B.; and Birch, A. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 86–96. Berlin, Germany: Association for Computational Linguistics.

Sennrich, R.; Haddow, B.; and Birch, A. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725. Berlin, Germany: Association for Computational Linguistics.

Sun, T.; Shao, Y.; Li, X.; Liu, P.; Yan, H.; Qiu, X.; and Huang, X. 2020. Learning Sparse Sharing Architectures for Multiple Tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 8936–8943. AAAI Press.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112.

Thompson, B.; Gwinnup, J.; Khayrallah, H.; Duh, K.; and Koehn, P. 2019. Overcoming Catastrophic Forgetting During Domain Adaptation of Neural Machine Translation. 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)*, 2062–2068. Minneapolis, Minnesota: Association for Computational Linguistics.

Thompson, B.; Khayrallah, H.; Anastasopoulos, A.; McCarthy, A. D.; Duh, K.; Marvin, R.; McNamee, P.; Gwinnup, J.; Anderson, T.; and Koehn, P. 2018. Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, 124–132. Brussels, Belgium: Association for Computational Linguistics.

Tian, L.; Wong, D. F.; Chao, L. S.; Quaresma, P.; Oliveira, F.; Lu, Y.; Li, S.; Wang, Y.; and Wang, L. 2014. UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 1837–1842. Reykjavik, Iceland: European Language Resources Association (ELRA).

Tiedemann, J. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2214–2218. Istanbul, Turkey: European Language Resources Association (ELRA).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Vilar, D. 2018. Learning Hidden Unit Contribution for Adapting Neural Machine Translation Models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 500–505. New Orleans, Louisiana: Association for Computational Linguistics.

Wang, W.; Tian, Y.; Ngiam, J.; Yang, Y.; Caswell, I.; and Parekh, Z. 2020. Learning a Multi-Domain Curriculum for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7711–7723. Online: Association for Computational Linguistics.

Wuebker, J.; Simianer, P.; and DeNero, J. 2018. Compact Personalized Models for Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 881–886. Brussels, Belgium: Association for Computational Linguistics.

Zeng, J.; Liu, Y.; Su, J.; Ge, Y.; Lu, Y.; Yin, Y.; and Luo, J. 2019. Iterative Dual Domain Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 845–855. Hong Kong, China: Association for Computational Linguistics.

Zhang, T.; Wu, F.; Katiyar, A.; Weinberger, K. Q.; and Artzi, Y. 2020. Revisiting Few-sample BERT Fine-tuning. *arXiv preprint arXiv:2006.05987* .

Zhu, M.; and Gupta, S. 2018. To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*.