Generalizability of Mixture of Out-of-Domain Adapters from the Lens of Signed Weight Directions and its Application to Effective Model Pruning

Anonymous ACL submission

Abstract

Several parameter-efficient fine-tuning methods based on adapters have been introduced as a streamlined approach to incorporate not only a single specialized knowledge into existing Large Language Models (LLMs) but also multiple of them at once. However, understanding their generalizability across different out-of-domain tasks and their adversarial robustness remains unexplored. Thus, in this study, we conduct a comprehensive analysis to elucidate the workings of the Mixture of Out-of-Domain Adapters, offering insights across various facets, ranging from training data characteristics to the intricacies of adapter weights within the framework of the Mixture of Adapters. Specifically, we propose to analyze how the signed directions of adapters' weights during mixing correlate with their generalizability and how such analysis allows us to design a more effective model pruning algorithm that balances space, time, and predictive performance. The source code of the paper will be publicly available.

1 Introduction

Pre-trained Large Language Models (LLMs) are often complex and large in size. This makes the fine-tuning of a distinct model for every new specialized tasks very computationally expensive. Consequently, several *parameter-efficient fine-tuning methods that are based on adapters* have been introduced as a streamlined approach for incorporating new, specialized knowledge into existing LLMs. Several works have proposed to train a distinct adapter for each new domain (Houlsby et al., 2019; Pfeiffer et al., 2021; Hu et al., 2022). However, to further improve model robustness, a myriad of efforts now aim to utilize a blend of these adapters, intending to seamlessly merge knowledge from diverse knowledge sources. For instance, AdapterFusion (Pfeiffer et al., 2021) proposes to integrate multiple task adapters. However, it does not specify any particular architecture or training methodologies for assimilating external knowledge. A similar approach K-Adapter (Wang et al., 2021a) is also somewhat limited, training only on T-REx triples, hence missing the versatility to accommodate unstructured knowledge. On the other hand, more recent approaches such as MixDA (Diao et al., 2023) do not only train domain-specific adapters but also implement mechanisms for adapter routing, albeit with variations in routing techniques, foundational models, and training approaches.

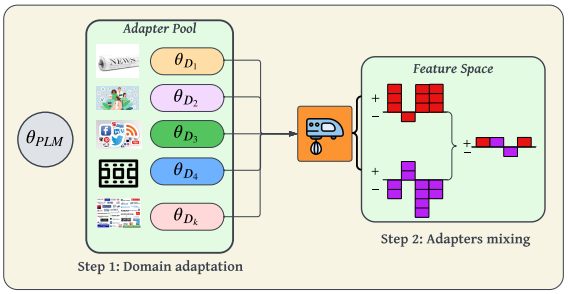

Figure 1: Mixing the adapter weights across various tasks may result in the importance weights of individual tasks nullifying each other, thereby yielding a merged adapter without maintaining any influential features after the fusion process. The evaluation method includes two steps: In Step 1, we train domain adapters with domain-specific knowledge. In Step 2, we mix adapters and evaluate with downstream tasks

Following this trend, existing works in this domain mostly focus on training multiple adapters for multiple tasks and continuously adding more adapters for incoming new tasks. This can be inefficient for the new domain tasks that have only a few examples, making the learning among the tasks unequal. Therefore, more recent works such as Matena and Raffel (2022); Wang et al. (2022, 2021b); Li et al. (2022); Chronopoulou et al. (2023) opt for weight-space averaging of model and/or adapters trained on different domains, resulting in *Mixture of Expert Adapters*, reporting superior predictive performance across out-of-domain tasks. However, several questions regarding the final mixtures of out-of-domain adapters remain

unanswered, especially those regarding their generalizability and their adversarial robustness when adapters from very different tasks are combined. Given the reported superiority in space, time, and predictive performance in these works, it is worth analyzing their performance trade-off in practice and investigating how such performance correlates with their defacto weight-space averaging mechanism, which is often the key component in classical ensemble Machine Learning literature.

Therefore, borrowing the pop-culture saying that *"mixed drinks and cocktails aren't actually the same thing"*, contrast from existing works, we hypothesize that *not all* Mixture of Expert Adapters are created equal and all have superior performance. Then, we attempt to explain and give answers to questions when and what to mix when it comes to domain-specific adapters. Specifically, we want to answer the following questions: (1) What will happen to model generalization, and robustness when one additional adapter is injected to LLMs? (2) How does each domain knowledge affect to model parameters? (3) In the case of several adapters that are very different from each other, is there any way to fuse these adapters more efficiently than the weight averaging method? To answer these questions, we will focus on analyzing the correlation between *signed directions of adapter weights during mixing* and task-specific predictive performance as our main hypothesis for performance gain analysis (Figure 1). Although simple, this intuitive and novel hypothesis also allows us to design a more effective model pruning as an application.

Our contributions are summarized as follows.

1. This is the first and most comprehensive analysis on the generalizability and adversarial robustness of mixture of domain-specific adapters with 3 different adapter methods on 13 diverse datasets,
2. It provides insights and analysis on when and what adapters to mix to achieve optimal performance via the lens of signed directions of adapters' weight matrices,
3. It demonstrates applications of such insights in more effective adapter-based model pruning.

2 Related works

Adapter Fine-tuning. The primary method for adapting general-purpose LLMs to downstream tasks is via *full fine-tuning*, which requires adjusting all models' parameters (Peters et al., 2018; Devlin et al., 2019a). However, this results in redundant copies of fine-tuned models for each task, posing a significant space challenge for systems handling numerous tasks. To address this, various *parameter-efficient fine-tuning methods* have been proposed, including prompt-based tuning (Li and Liang, 2021) and adapter-based tuning (Houlsby et al., 2019; Pfeiffer et al., 2021; Hu et al., 2022). Additionally, approaches like K-Adapter (Wang et al., 2021a), AdapterFusion (Pfeiffer et al., 2021), MAD-X (Pfeiffer et al., 2020) and AdaMix (Wang et al., 2022) further optimize their adapters for various downstream tasks by maintaining a set of adapters and combine them together during inference. However, their adapters only focus on tasks that require additional in-domain and *not new or out-of-domain* knowledge. Thus, works such as MixDA (Diao et al., 2023) propose a promising way to adapt new domain knowledge while preserving existing one.

Mixture of Expert Adapters. Wang et al. (2022) fine-tunes so-called Mixture of Experts (MoEs) with adapters on a downstream task and averaging their weights during inference. Moreover, Wang et al. (2021b) enhances performance in an unseen target language by ensembling the source language adapters. Exploring performance in novel domains through weight averaging, (Li et al., 2022) focuses on entire language models. Similarly, AdapterSoup (Chronopoulou et al., 2023) opts for weight-space averaging of adapters trained on different domains. Weight averaging is identified as a viable solution in this context, as it allows for the preservation of LLMs performance on new domains while ensuring robust in-domain results, as evidenced by studies such as (Jin et al., 2023a) and (Chronopoulou et al., 2023). These diverse strategies contribute to the evolving landscape of model merging and ensemble techniques in the realm of fine-tuned LLMs. However, none of them comprehensively evaluates and analyzes the generalizability and adversarial robustness of the mixed model under various conditions of *out-of-domains* knowledge.

3 Mixture of Adapters Benchmark

Our benchmark includes two steps. First, we train several adapters with domain-specific knowledge. Second, we mix those adapters in different combinations and evaluate each of them on different downstream tasks on two aspects: (1) generalizability and (ii) adversarial robustness under adversarial text attacks.

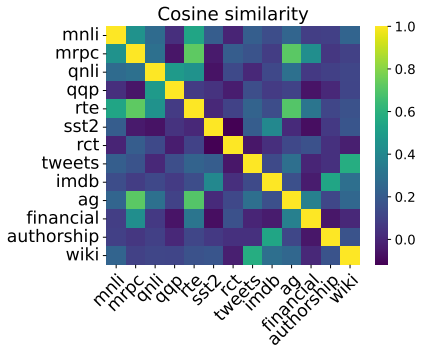

Figure 2: Cosine similarity among datasets. Universal Sentence Encoder (USE) (Cer et al., 2018) is used to generate embeddings of 1K randomly sampled documents from each dataset. Cosine similarities are then calculated using the centroid embeddings of each dataset.

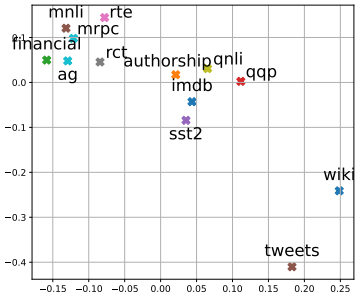

Figure 3: Visualization of Topic distribution over all datasets. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is used to extract 10 most important words that represent the topics of each domain dataset. We then use FastText [2] to extract their embeddings and average them with PCA for 2D visualization.

3.1 Dataset

Diverse Knowledge Datasets. To simulate knowledge diversity, we gather a total of 13 distinct and diverse *out-of-domain* datasets or classification tasks for evaluation. They are MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Bentivogli et al., 2009), MRPC (Dolan and Brockett, 2005), QQP (Iyer et al., 2017) and SST2 (Socher et al., 2013) from the *GLUE domain* corpus. PubMed-20K RCT dataset (Dernoncourt and Lee, 2017) from *Biology domain* for sentence classification. IMDB dataset from a *Movie Review domain*. Ag News, Financial (Malo et al., 2014) and Guardian Authorship (Altakrori et al., 2021) are *News domain* datasets across World, Sports, Business, Science/Technology, and Financial topics. Wiki Toxic [1] and Tweets Hate Speech are two *Informal text domain* for toxicity detection. We refer the readers to Appendix A.1 for detailed statistics such as number of documents, average document length, and sentence lengths.

Semantic and Topic Distributions. Figure 2, Figure 3 illustrates the cosine similarity among datasets and the topic distribution across different domain datasets, respectively. The figures reveal the intricate relationships within our diverse selected datasets. Notably, SST2 and IMDB, both originating from the same movie corpus, exhibit proximity in topic embedding spaces. On the contrary, non-formal datasets such as Wiki and Tweets are distinctly distant from other datasets in this regard. For a more detailed exploration of the topics within the training data, please refer to Sec. A.2 of Appendix.

[1] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/

3.2 Mixing Fine-Tuned Adapters

Models and Individual Adapters. We design our evaluation on two transformer-based models, namely **BERT** (Devlin et al., 2019b) and **RoBERTa** (Liu et al., 2019) with a total of 3 diverse and well-known adapter methods. They are Houlsby (Houlsby et al., 2019), Pfeifer (Pfeiffer et al., 2021) and LoRA (Hu et al., 2022). This adapter-based method introduces variations in the adapter architecture and parameterization, contributing to the exploration of efficient and effective ways to adapt pre-trained models for specific downstream tasks. In particular, Houlsby introduces two adapter blocks with bottleneck networks in each Transformer block, augmenting the RoBERTa model for downstream tasks. Similarly, the Pfeiffer adapter differs in architecture, incorporating only one adapter layer in each Transformer block, in contrast to the two layers introduced by Houlsby. Pfeiffer makes minor adjustments to include layer normalization. LoRA takes a distinctive approach by freezing the MLP modules of transformers and representing updates to attention weights with two low-rank matrices to optimize space while effectively retaining model performance.

Mixing Adapters. From a pre-trained large model (PLM) such as BERT and RoBERTa, denoted as θ_{PLM}, we proceed to train a suite of optimal adapters tailored for diverse domains, specifically $\theta_{D_1}, \theta_{D_2}, \ldots, \theta_{D_k}$. In scenarios involving tasks that have only a few examples, the fusion process amalgamates the weights of all adapters, facilitating predictions without the need for further fine-tuning. Following (Chronopoulou et al., 2023), the final inference become:

$$f(x, \theta_{PLM} + \frac{1}{k}\sum_{i=1}^{i=k} \theta_{D_i}) \quad (1)$$

3.3 Adversarial Text Generation

Textual adversarial attacks are popular in AI robustness research. Technically speaking, given a dataset $D=\{(x_i, y_i)\}_{i\in[N]}$, where x represents the sample and y denotes the ground truth label, a textual adversarial attack aims to attack an LLM f_θ with a classification loss function L by perturbing each sample x with δ given a certain budget C:

$$\arg\max_{\delta\in C} L[f_\theta(x+\delta), y], \quad (2)$$

Toward evaluating the robustness of mixture of adapters, we then modify the existing *black-box and white-box* textual attacks to implement Equation 2. We utilize the popular *TextFooler* (Jin et al., 2020) as the *black-box attack*, which aims to replace words with synonyms or contextually similar words to deceive LLMs. We utilize the well-known *FGDS* (Goodfellow et al., 2015) as the *white-box attack*, which can efficiently craft adversarial examples by perturbing embedding of text data in the direction of the sign of the gradient of the loss function to the input, thereby exposing vulnerabilities in model robustness.

3.4 Combinatory Evaluation

We first train a single adapter for each of the 13 tasks, resulting in a total of 13 adapters upon completion of training. Subsequently, for each target task, we generate combinations from the set of 13 tasks. To illustrate, for MNLI, when combining two adapters, we have the flexibility to choose 1 adapter out of the remaining 12, resulting in 12 possible combinations. For a set of 3 adapters, including MNLI, we select 2 task adapters out of the 12 to generate C_{12}^2 combinations. This process continues similarly for sets ranging from 4 to 13 adapters, where, in the case of 13 adapters, all adapters are combined. In summary, for each target task, we have $\sum_{i=1}^{12} C_{12}^i$ combinations. Subsequently, for each combination in the set of k adapters, we evaluate task performance after merging, computing the mean and variance over all combinations.

4 Experiments

4.1 Implementation Details

We employ training and evaluation datasets to gauge the accuracy of datasets in the GLUE corpus. On the Ag News, Authorship, Financial, IMDB, Tweets, and Wiki-Toxic, we partition the training set into three segments with an 8:1:1. For Black-Box (*TextFooler* (Jin et al., 2020)), we set the minimum embedding cosine similarity between a word and its synonyms as 0.8, and the minimum USE similarity is 0.84. With WhiteBox (FGDS (Goodfellow et al., 2015)), we choose the magnitude of the perturbation in embedding space as 0.01. We use adapters with a dimension of 64 and 256 using RoBERTa-large and BERT-base encoders following the setup of (Houlsby et al., 2019), (Pfeiffer et al., 2021). With LoRA, we use rank r=4 following the setup of (Hu et al., 2022). Detailed training, evaluation dataset, and hyper-parameter configuration for different tasks are presented in Sec. A.3 in the Appendix.

4.2 Results

We present the performance of RoBERTa with Houlsby adapter with different numbers of additional mixing domains in Figure 4. Table 1 shows how much the predictive performance drops without and with adversarial black-box and white-box attacks when mixing all adapters. Overall, the average performance drops over all tasks on the clean test set from the original performance to a mix of all adapters is 11.3%, and that for black-box and white-box attacks are 12.1% and 10.2 %, respectively. For further results on different model performances in other adapter methods, we refer the readers to Figures 10, 11, 12, 13 in Sec. A.4 of the Appendix.

Finding #1: As we add more tasks or domains, the predictive performance of every single task decreases, reaching its lowest point when we incorporate the maximum of 13 adapters. The same behaviors were also observed in (Jin et al., 2023b) where they merge the weight of pre-trained models. Notably, task accuracy remains stable for QNLI and SST2 when mixing adapters, indicating that incorporating knowledge from other domains does not consistently impair model performance. In contrast, a substantial decrease in accuracy is observed for the remaining tasks during the merging of domain adapters, especially between RCT, IMDB, Ag News, and Authorship. This suggests that *the nature of mixture domains or tasks might have a crucial effect on the mixture of adapters*, causing it to forget the task-specific knowledge.

Finding #2: On average, adversarial setting dropped 12.1% in black-box compared to 11.3% in clean. The overall predictive performance was significantly lower under the white-box compared to the black-box attack, highlighting the efficacy

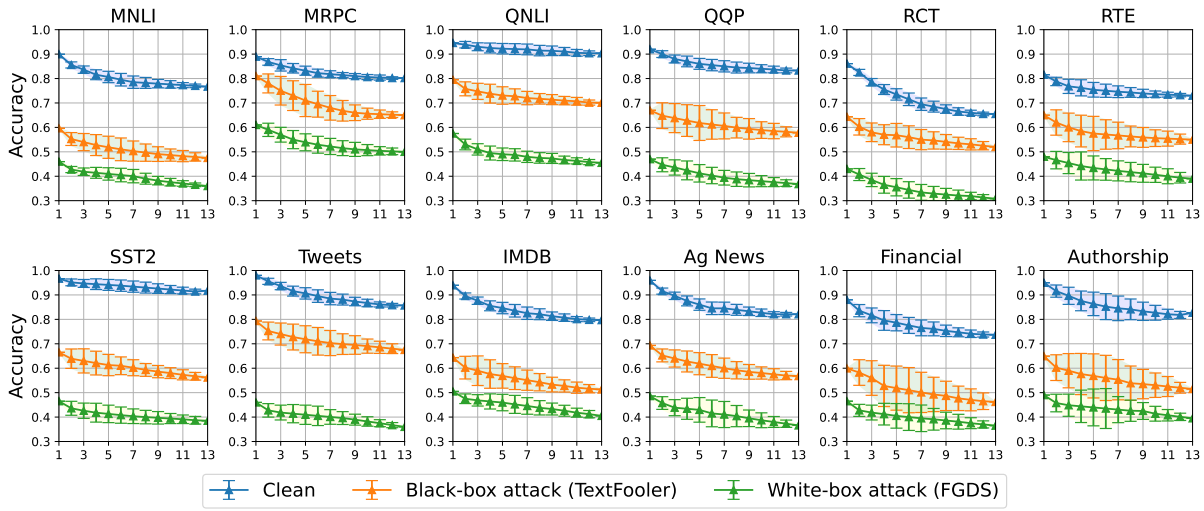

Figure 4: Accuracy of RoBERTa with Houlsby Adapter across various distribution datasets. *The x-axis denotes the number of domain adapters to be mixed, ranging from 1 to 13.*

Dataset	mnli	mrpc	sst2	rte	qnli	qqp	rct	ag	authorship	financial	imdb	tweets	wiki	Average
∇_{clean}	**13.0**	8.3	4.9	8.5	5.2	8.6	**20.1**	**13.8**	11.4	**14.3**	**14.2**	**12.3**	**12.1**	11.3
$\nabla_{blackbox}$	12.1	**16.0**	**10.4**	**10.2**	10.1	9.8	12.3	12.6	**13.4**	13.8	13.0	11.8	10.5	**12.1**
$\nabla_{whitebox}$	10.0	11.0	8.3	9.4	**12.0**	**10.2**	12.5	12.1	10.5	9.5	10.0	10.2	7.5	10.2

Table 1: Average *absolute* performance drop (*in percentage %*) when mixed from all domain adapters on clean, black-box and white-box attack.

of white-box attack methods. Specifically, on the RCT dataset, the accuracy drop on clean is 20% which is much bigger on black-box and white-box attacks with 12.3% and 12.5%.

Finding #3: Variance in task accuracy on adversarial attacks, when combining different domain adapters, is observed to be larger than the variance in clean accuracy (Figure 4). The difference in performance variance with and without adversarial attacks suggests that the incorporation of adapters from different domains enhances the model's robustness, contributing to a more resilient performance in the face of adversarial attacks.

Mixing adapters from different domains makes a model that can provide prediction out-of-distribution/ unseen tasks without retraining. However, mixing the adapter decreases performance in the current task. Therefore, it is important to understand which task we should use to mix to have a final adapter that can have good enough performance in the out-of-distribution while maintaining its performance in the original task.

5 Effects of Sign Differences of Adapter Weights during Mixing: A Hypothesis

A Hypothesis. The ideal scenario when incorporating an adapter with new tasks is to make minimal adjustments to its weights, both in terms of values–i.e., magnitudes, and directions, to sustain its original performance. However, quantifying such disparity in adapter weights during fusion, considering both values and directions, proves to be overly intricate. In our analysis, we simplify this assessment by focusing on the *sign directions* of the adapter weights. Intuitively, following the mixing process of k individual adapter weight in Equation 1 (Sec. 4), *we hypothesize that the sign differences–i.e., positive v.s. negatives, in adapter weights during mixing correlate with the mixture of k number of domain-specific adapters' generalizability*. As illustrated in Figure 1, the averaging of adapter weights across various tasks may lead to the nullification of importance weights for individual tasks. Consequently, the merged adapter might lose influential features during the fusion process, emphasizing the importance of carefully managing the fusion procedure.

We first test and analyze such correlation with individual adapters (mixture with k=1), dual adapters (k= 2), and then generalize to multiple adapters ($k\geq$2). To demonstrate the utility of our analysis, we provide an application of our hypothesis on effective model pruning in Sec. 6.

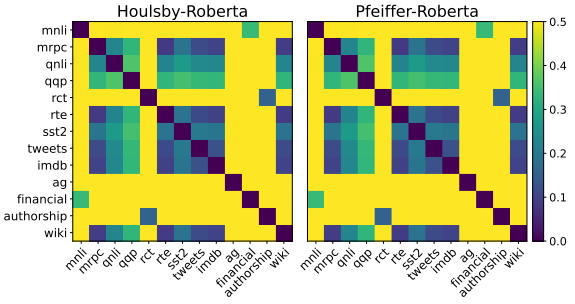

Figure 5: Fraction on differences of adapter weight direction.

5.1 Individual Adapters (k=1)

We calculate the difference in the direction of adapter weights on RoBERTa and normalize it by the total number of adapter weights, as depicted in Figure 5. We refer the readers to Sec. A.5 in the Appendix for results on BERT. Figure 3 indicates that datasets with distinct topic distributions and cosine similarities (Sec. 3.1) exhibit varying weight directions. This suggests that adapters follow different optimization trajectories, leading to diverse optimal weight values. Interestingly, a consistent trend in the difference in weight direction is observed across various model architectures (BERT, RoBERTa) and adapter methods (e.g., Figure 5 and Figure 9 of Appendix A.5). The reason is that adapters act as small MLP layers that integrate task-specific knowledge into pre-trained models (Meng et al., 2022) in different adapter methods. This shared functionality contributes to a similar trend in weight direction differences, highlighting the robustness and generalizability of the observed behavior across different architectural and methodological variations.

5.2 Dual Adapters (k=2)

Weight Direction. Figure 6 illustrates the proportion of changes in weight direction for each adapter during the integration of two domain-specific adapters. This investigation is conducted within the context of the Pfeiffer Adapter using a pre-trained RoBERTa model. **Task Performance.** Figure 6 illustrates the proportion of weight changes in direction and task accuracy resulting from the combination of two adapter weights. Overall, task performance experiences a decline when merging its adapter with the adapter weight from another task. Notably, tasks with substantial differences in the fraction of weight direction witness a pronounced performance decrease. Specifically, MNLI, RCT, Ag News, Financial, and Authorship exhibit significant performance drops due to substantial differences in adapter weight direction. This results in a substantial change in the current adapter weight upon mixing. Conversely, tasks such as MRPC, QNLI, and RTE demonstrate either marginal improvement or no change in performance when mixed with other adapters. This is attributed to the minimal difference in adapter weight direction after mixing, ranging from 5% to 10% compared to the original weight.

5.3 Multiple Adapters ($k\geq$2)

Weight Direction. Figure 7 depicts the proportion of weight direction conflicts relative to the number of adapter weights being merged. Similar to the dual-adapter setting, we observe that increasing the number of mixed adapters amplifies the disparity in the target adapter weight direction. This effect is significant in tasks such as MNLI, QQP, RCT, Ag News, Authorship, and Financial, where the target adapter weight exhibits substantial dissimilarity compared to other adapters. **Task Performance.** In the set of tasks exhibiting substantial disparities when mixing adapter weights, the model experiences a large performance decrease post-mixing. Conversely, tasks within the MRPC, QQP, RTE, and SST2 groups demonstrate a more modest performance drop when their adapters are mixed. This can be attributed to the relatively minor differences in adapter weights compared to other tasks, as depicted in Figure 5. The mixing of these adapters does not result in a significant alteration of model parameters, preserving crucial feature weights for the target task.

6 Towards Effective Model Pruning

To demonstrate the utility of our analysis in Sec. 5, we will apply it to tackle the task of model pruning. As shown in Figure 6, we reveal that task performance experiences a significant drop when integrating adapters with pronounced disparities in weight signs–i.e., positive with negative signs. Moreover, our observation indicates that only a limited number of weights significantly contribute to task performance, suggesting redundancy within the weights that can be pruned without compromising the original task performance (Frankle et al., 2021). To mitigate the impact of weight sign differences, we propose mixing only a sparse version of the adapter's weights. This strategy indirectly reduces the fraction of weight sign conflicts. Consequently, by minimizing the fraction of weight

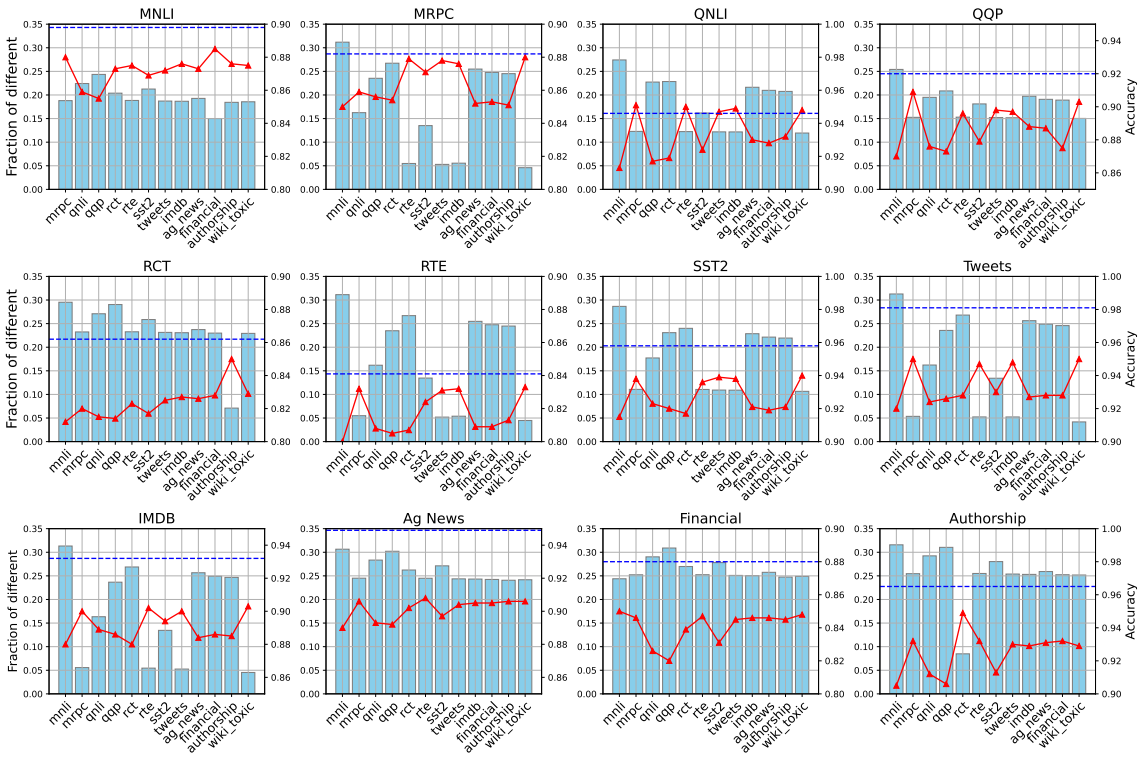

Figure 6: Fraction of weights altering direction during the consolidation of two adapters. The *dashed blue line* denotes the accuracy achieved by a standalone adapter trained on a specific task. While the *solid red line* illustrates the variations in accuracy when merging the adapter with another task's adapter.

sign conflicts, the mixing process becomes more resilient to the inadvertent elimination of important weights by less significant or redundant weights. This phenomenon is visually depicted in Step 2 of Figure 1, where significant weights in the two adapters are preserved, and small or unimportant weights of opposing signs are minimized.

Pruning Adapters at Initialization. Sparse Adapter (He et al., 2022), employs pruning at initialization across every layer of adapters, being able to achieve comparable or even superior predictive performance than standard adapters, even when the sparse ratio reaches up to 80%. By adopting a similar process of pruning adapters at initialization *together with our analysis in Sec. 5*, we can eliminate redundant parameters at an early stage, circumventing the need for a time-consuming iterative pruning process, as discussed in prior work by Frankle et al. (Frankle and Carbin, 2019).

Specifically, considering an adapter with weights w^l inserted in the layer $l \in \{1, \cdots, L\}$, parameters can be pruned by a binary mask m^l as $\tilde{w}_i^l = w_i^l \odot m_i^l$, where $\tilde{w}_i^l$ denotes the pruned parameters, w_i^l and m_i^l denote the i-th element of w^l and m^l, respectively. Given target sparsity s, scores z are assigned to all parameters w, and redundant parameters with scores below the threshold z_s (the s-th lowest percentile of z) are removed. In this study, we adopt magnitude pruning (Han et al., 2015), where each parameter is assigned a score $z = |w|$ as its score and removes parameters with the lowest scores. Following the methodology outlined in (Frankle et al., 2021), we employ magnitude pruning at the initialization stage as shown in Algorithm 1.

Algorithm 1: Magnitude pruning on adapters weight.

Input: adapter paramters w, sparse ratio s.
Output: pruned adapter $\tilde{w}$.

$w \leftarrow \text{Initialization}(w)$
Compute important score $z = |w|$
Compute the s-th percentile of z as z_s
$m \leftarrow \mathbb{1}\left[z - z_s \geq 0\right]$
$\tilde{w} \leftarrow m \odot w$

Task Performance with Pruned Adapters. We present RoBERTa's performance in Figure 8a, where we systematically prune the weight of the Pfeiffer adapter from 0% (without using the adapter) to 100%. For a single task at the density level k, we retain only the largest, i.e., the top-k% influential parameters of the corresponding adapter, and evaluate task performance with

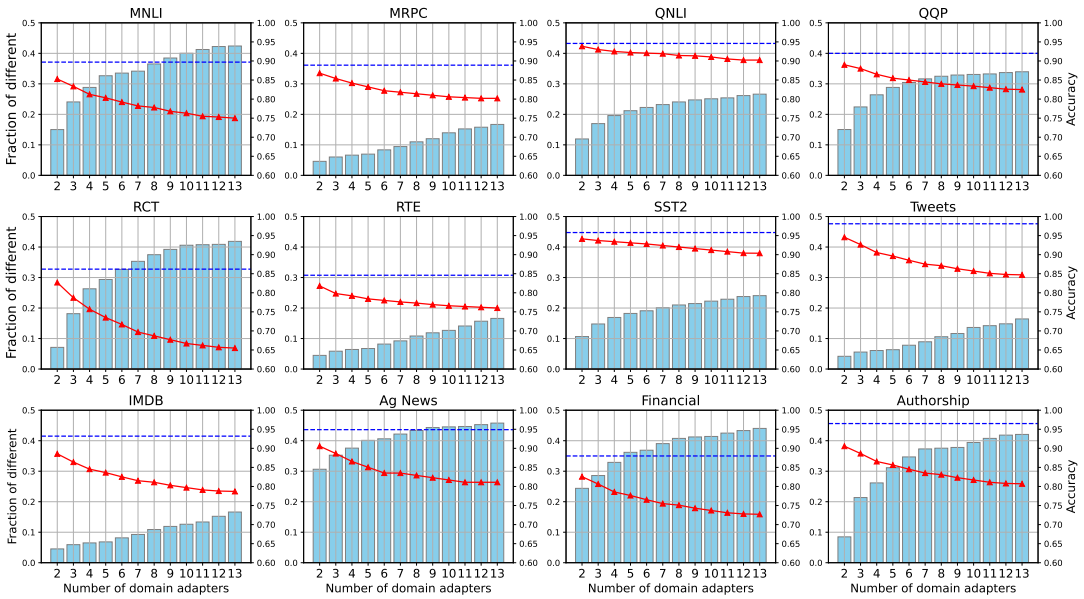

Figure 7: Fraction of weights changing direction during the mixing of multiple adapters, ranging from 2 to 13. The *dashed blue* line corresponds to the accuracy of a single adapter trained on a specific task, while the *solid red line* depicts the fluctuation in task accuracy resulting from merging the adapter with another task's adapter.

the pruned adapter. This evaluation is conducted across all 13 tasks, and the average performance is computed. Remarkably, retaining only the top 30% influential parameters does not lead to performance degradation. This observation suggests redundancy in adapter parameters, contributing to the increase in the fraction of weight direction conflicts when merging these adapters (Figure 7).

Task Performance when Mixing Pruned Adapters. Weight ensemble is practical for out-of-distribution domains or resource-restricted settings, preserving PLM performance on new domains while delivering robust in-domain results (Jin et al., 2023a). However, element-wise average parameter weights can lead to a reduction in the importance of influential parameters due to the combination of weights in opposite directions. To evaluate our hypothesis, we first pruned all the adapter weights to retain only the top 30% weight parameters. We then mixed these adapters to investigate whether removing redundancy parameters would help improve performance on mixed adapters. Figure 8b shows the performance on MNLI when different sparse adapters are mixed. Overall, when pruning important weights, the fraction of weight difference in direction decreases. Interestingly, mixing these sparse adapters can help remove parameter redundancy on tasks, contributing to improved model generalization.

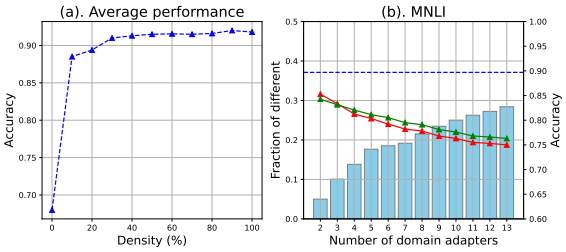

Figure 8: (a). Average RoBERTa performance with a single sparse adapter on 13 tasks with increasing density of adapter weight. 0%: pruning all adapter weights, keeps only the weight of the pre-trained model, while 100% denotes keeping all the weights of the adapters. (b). Model performance when increasing the # of sparse adapters being mixed. The *red* line represents the MNLI accuracy when mixing out-of-domain adapters, while the *dashed green* line depicts the MNLI performance when mixing sparse out-of-domain adapters. The *dashed blue* line represents the MNLI performance of a single adapter.

7 Conclusion

In conclusion, our study provides a comprehensive investigation into the inner workings of the mixture of out-of-distribution adapters, yielding insights across diverse aspects from the characteristics of training data to the sign of adapter weights. By examining the signed directions of adapter weight, we offer valuable insights and analyses to guide the optimal selection of adapters for achieving peak performance and establish a correlation between model pruning and the mixture of adapters, enhancing our understanding of their interconnected roles in the context of sparse neural networks.

Limitation

Despite the progress we made, there still exist limitations in our work. Primarily, our exploration focused solely on one classic pruning method, namely Magnitude Pruning. It is essential to acknowledge the possibility of other advanced pruning techniques that can take advance of neural network architecture. Consequently, future endeavors should delve into investigating these alternatives. Furthermore, our examination was confined to natural language understanding tasks. A valuable avenue for future research would involve extending our analysis to encompass text generation tasks, particularly within the context of the current transformer-based language model, such as machine translation utilizing GPT.

References

Malik Altakrori, Jackie Chi Kit Cheung, and Benjamin CM Fung. 2021. The topic confusion task: A novel evaluation scenario for authorship attribution. In *EMNLP*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. In *Journal of machine Learning research*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *EMNLP*.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. 2023. Adaptersoup: Weight averaging to improve generalization of pre-trained language models. In *EACL*.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang. 2023. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pre-trained language models memories. In *ACL*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing*.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. 2021. Pruning neural networks at initialization: Why are we missing the mark? In *ICLR*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *NeurIPS*.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. 2022. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In *EMNLP*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICLR*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First quora dataset release: Question pairs. In *data. quora. com.*

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *AAAI*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023a. Dataless knowledge fusion by merging weights of language models. In *ICLR*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023b. Dataless knowledge fusion by merging weights of language models. In *ICLR*.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. In *arXiv*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. In *Journal of the Association for Information Science and Technology*.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *NeurIPS*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *EMNLP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *ACL*.

Xinyi Wang, Yulia Tsvetkov, Sebastian Ruder, and Graham Neubig. 2021b. Efficient test time adapter ensembling for low-resource language varieties. In *EMNLP*.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *EMNLP*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Dataset	**mnli**	**mrpc**	**sst2**	**rte**
Train	392,702	3,668	67,349	2,490
Test	9,815	408	872	277
Dataset	**qnli**	**qqp**	**rct**	**ag**
Train	104,743	363,846	178,882	120,000
Test	5,463	40,430	30,135	7,600
authorship	**financial**	**imdb**	**tweets**	**wiki**
2,743	4,846	22,500	31,962	127,656
686	484	2,500	3,196	63,978

Table 2: Number of instances for each dataset divided by training and test set.

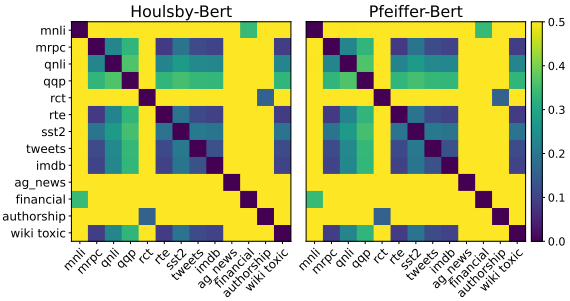

Figure 9: Fraction on differences of adapter weight direction.

A Appendix

A.1 Linguistic statistic

Table 3 shows linguistic statistics in terms of training text.

A.2 Topic distribution of training datasets

Tables from 4 to 16 show 10 topics and corresponding important words which are exacted from LDA for each training dataset.

A.3 Hyper-parameter

Training and evaluation datasets. To assess performance in out-of-distribution scenarios, we conduct evaluations on a diverse set of 13 datasets covering various topics, ranging from movie reviews, news, authorship, and healthcare, to non-formal language text such as Wiki Toxic and Tweets. For datasets within the GLUE corpus, we employ training and evaluation datasets to gauge accuracy across different settings. In the case of Ag News, Authorship, Financial, IMDB, Tweets, and Wiki-Toxic, we partition the training set into three segments with an 8:1:1 ratio, utilizing them for training, evaluation, and test datasets, respectively. This approach ensures a comprehensive evaluation of model performance across a wide spectrum of domains and linguistic styles. Table 2 shows data statistics on train/test datasets.

Data Source	Average Document Length	Average Sentence Length	Average # Sentences per Document
MNLI	15.1	14.7	1.0
MRPC	21.9	21.1	1.0
QNLI	18.2	18.0	1.0
QQP	11.1	9.9	1.2
RTE	26.2	18.1	1.4
SST	10.4	10.4	1.0
RCT	26.5	26.3	1.0
Ag-news	38.4	29.1	1.3
Authorship	1038.6	20.2	51.3
Financial	23.1	22.8	1.0
IMDB	233.8	21.6	10.8
Tweets	15.9	9.6	1.6
Wiki-toxic	67.8	15.4	4.4

Table 3: Length statistics.

Table 4: Topic distribution on MNLI dataset

#Topic	MNLI
1	well, time, got, take, one, much, day, something, ive, even, way, long, little, make, back
2	kind, system, though, come, went, well, today, view, church, including, president, seems, across, run, policy
3	say, get, cost, guess, were, business, car, local, whole, north, rather, getting, question, technology, capital
4	service, state, world, get, big, pretty, give, war, yes, standard, real, here, came, call
5	probably, high, thought, however, set, hand, enough, said, since, type, jon, yet, and, service
6	could, mean, around, part, another, change, percent, made, course, life, book, fact, name, room
7	government, program, federal, information, country, problem, le, new, national, may, number, agency, report, organization
8	year, two, house, case, old, three, town, street, century, one, city, study, man, four, different
9	know, like, think, thats, right, really, people, thing, good, go, one, lot, going
10	yeah, work, legal, rule, last, year, he, american, small, home, company, act, group, analysis, public

Setting on text adversarial attack. In this study, we employ two types of attacker methods: *TextFooler* (Jin et al., 2020) and FGDS (Goodfellow et al., 2015).

TextFooler word-level attacks focus on replacing words within the text with synonyms or contextually similar words. By making ostensibly minor alterations to the input text, these attacks can deceive LLMs into producing incorrect outputs or substantially modifying their predictions. We meticulously fine-tune the hyperparameters of TextFooler to obtain more appropriate synonyms. We set the minimum embedding cosine similarity between a word and its synonyms as 0.8, and the minimum Universal Sentence Encoder similarity is 0.84.

FGDS (Goodfellow et al., 2015) is a white-box embedding-level attack. FGDS uses the Fast Gradient Sign Method (FGSM) to calculate gradients of the model's loss to the input text and generates an adversarial example by perturbing the embedding of input text in the direction that maximizes the loss. We choose the magnitude of the perturbation in embedding space as 0.01 on BERT and RoBERTa models.

Adapter Configuration. We use adapters with a dimension of 64 and 256 using RoBERTa-large and BERT-base encoders following the setup of (Houlsby et al., 2019), (Pfeiffer et al., 2021). With LoRA, we use rank $r = 4$ following the setup of (Hu et al., 2022).

Hardware Information. We evaluate model performance on AMD Ubuntu 22.04.2 with Ryzen Threadripper PRO 5975WX, 1800MHz, Cached 512 KB and 4 $\times$ GPU Nvidia A6000. **Hyper-Parameters.** Detailed hyper-parameter configuration for different tasks is presented in Table 17.

A.4 Model performance when mixing adapters across tasks

Tables 10, 11, 12 and 13 show task accuracy when mixing multiple adapters.

A.5 Additional result on weight different

Figure 9 shows the difference in the direction of adapter weights and normalizes it by the total number of adapter weights in BERT.

Table 5: Topic distribution on MRPC dataset

#Topic	MRPC
1	said, court, company, would, official, statement, decision, made, state, appeal, two, board
2	said, year, people, president, program, time, million, two, last, house, official, weapon
3	said, million, would, state, period, compared, men, get, democratic, plan, company, united, also, could
4	percent, share, cent, million, stock, point, nasdaq, billion, new, index, trading, rose, per, year
5	said, also, state, iraq, center, united, attack, hospital, killed, war, three, american, people
6	said, two, home, police, told, state, friday, last, year, federal, company, yesterday, national
7	standard, poor, index, chief, point, said, percent, justice, one, spx, broader, executive, three
8	said, analyst, expected, street, many, suit, call, yesterday, angeles, wall, los, research, one, change, according
9	case, said, court, filed, death, also, charged, lawsuit, charge, state, found, reported, office, cancer
10	said, would, server, window, network, one, new, microsoft, also, taken, people, company

Table 6: Topic distribution on QNLI dataset

#Topic	QNLI
1	city, american, south, large, west, season, de, roman, service, art, london, first, located, street, new
2	state, united, new, including, people, city, national, million, school, north, government, army, many, within, building
3	also, system, later, early, used, based, part, control, four, use, death, official, known, act, called
4	group, language, east, among, found, common, company, india, federal, movement, population, early, included, production, range
5	the, church, term, example, university, greek, german, like, english, specie, god, word, per, old, one
6	form, although, following, law, central, rule, culture, without, often, modern, territory, society, treaty, considered, christian
7	war, world, british, life, development, empire, first, region, community, year, france, though, time, set, began
8	well, three, include, place, power, party, league, may, needed, right, one, political, club, a, event
9	became, first, time, john, film, president, number, year, french, one, day, land, america, process, le
10	century, music, around, house, home, period, age, record, late, established, several, standard, time, world, river

Table 7: Topic distribution on QQP dataset

#Topic	QQP
1	like, become, feel, get, job, movie, good, student, want, engineering, girl, website, sex, study, go
2	best, way, difference, learn, whats, money, make, online, book, india, buy, start, good, language, programming
3	much, best, time, weight, year, lose, old, place, month, day, iphone, read, possible, class
4	thing, day, business, get, first, going, example, one, prepare, video, woman, word, men
5	work, note, india, indian, ever, computer, black, r, science, you, help, rupee, different
6	would, life, trump, world, country, new, donald, war, india, win, happen, president, clinton, hillary
7	get, friend, used, long, why, bad, back, see, take, cant, good, facebook, system, relationship, person
8	someone, love, english, one, know, improve, account, people, get, instagram, tell, average, hair, password
9	mean, app, song, name, android, give, bank, right, what, company, india, working, get, now, create
10	people, quora, question, think, do, me, answer, google, stop, use, state, get, many, live

Table 8: Topic distribution on RTE dataset

#Topic	RTE
1	year, bank, world, ago, police, place, human, people, said, man, problem, game, many, took, explosion
2	people, attack, california, killed, life, united, day, lost, air, one, space, injured, national, capital, said
3	oil, said, nuclear, company, new, president, iran, million, military, john, un, country, bush, price
4	said, world, state, united, minister, country, million, people, nobel, south, peace, war, trade, prize, mexico
5	woman, corp, parliament, case, confirmed, said, rabies, represented, cause, poorly, fire, president, police, loss
6	year, new, said, one, would, died, university, show, company, family, first, service, since, country, home
7	state, iraq, said, bush, bomb, found, used, water, home, killed, caused, damage, one, police
8	party, police, president, new, two, officer, name, drug, state, prime, people, minister, last, year, democratic
9	new, said, government, year, iraq, would, york, official, today, baghdad, also, euro, announced, percent, minister
10	said, year, leader, new, sanfrancisco, work, justice, two, president, government, end, free, guerrilla

Table 9: Topic distribution on SST2 dataset

#Topic	SST2
1	film, really, enough, movie, something, make, interesting, many, like, subject, intelligent, laugh, short
2	movie, bad, film, better, great, fun, one, look, director, story, ultimately, smart, cinema, put
3	performance, funny, way, moment, film, cast, another, screen, yet, big, work, perfect, made
4	new, material, ve, movie, rather, film, special, seen, minute, enjoyable, might, offer, story, effect
5	comedy, drama, thriller, romantic, documentary, actor, moving, clever, funny, sometimes, pleasure, often, movie, film
6	work, film, movie, hard, well, keep, filmmaker, ever, life, original, sense, dull, quite, could
7	like, feel, movie, much, people, film, make, see, get, character, one, thing
8	good, real, film, worth, fascinating, make, time, lack, bit, amusing, humor, tale, pretty, run
9	character, one, best, film, movie, story, far, compelling, two, every, year, picture, little
10	love, audience, film, story, character, seems, entertainment, way, powerful, care, take, one, movie, spirit

Table 10: Topic distribution on RCT dataset

#Topic	RCT
1	group, patient, week, randomized, study, received, control, year, mg, randomly, placebo, day
2	patient, session, visit, cohort, failure, lesion, myocardial, hospital, twice,death, heart, infarction
3	analysis, using, data, model, used, test, sample, analyzed, regression, characteristic, time, collected, cell, method, performed
4	outcome, primary, month, patient, baseline, measure, score, secondary, treatment, scale, assessed, symptom, week, followup
5	risk, associated, level, weight, factor, effect, disease, body, increased, diabetes, insulin, high, glucose, change, activity
6	study, patient, treatment, effect, therapy, efficacy, may, effective, result, evaluate, safety, weather, clinical, outcome, intervention
7	group, difference, significant, significantly, compared, control, treatment, score, higher, lower, time, observed, rate
8	trial, study, randomized, intervention, care, health, controlled, clinical, quality, life, conducted, prospective, effectiveness, child, number
9	patient, event, surgery, adverse, postoperative, complication, procedure, pain, undergoing, surgical, rate, incidence, common, infection, injection
10	mean, respectively, ratio, patient, group, median, versus, interval, year, day, month

Table 11: Topic distribution on Tweets dataset

#Topic	Tweets
1	new, get, here, music, home, cool, playing, free, want, fun, season, shop, update, reason
2	day, one, night, time, good, week, last, never, first, get, year, got, lot, today
3	day, father, love, happy, time, weekend, take, friday, dad, fathersday, model
4	want, bull, up, do, help, trump, whatever, direct, dominate, waiting, libtard, yet, sleep, post
5	thankful, need, good, positive, orlando, morning, city, tear, news, blessed, friend, dream, bing, yeah, bong
6	user, amp, day, see, cant, go, like, new, today, one, people, get, wait, make
7	birthday, like, positive, affirmation, happy, baby, amp, god, girl, woman, feel, hate, hot, you
8	love, work, life, happy, happiness, make, always, food, quote, smile, wedding, moment, right, feeling, music
9	healthy, blog, gold, silver, altwaystoheal, forex, healing, grateful, dog, buffalo, peace, really, story
10	love, me, smile, summer, beautiful, fun, cute, girl, selfie, friend, sun, instagood, beach, photo

Table 12: Topic distribution on IMDB dataset

#Topic	IMDB
1	story, film, life, movie, character, one, love, time, people, see, way, family, would, well
2	movie, like, one, good, really, it, film, bad, see, even, time, would, make, get
3	get, one, man, the, go, woman, take, back, he, find, there, scene, two, girl
4	hamilton, gadget, arkin, scooby, talespin, stallion, smoothly, tenderness, shaggy, gil, inspector, keller, nevada, hopelessness
5	war, american, documentary, soldier, political, world, german, country, history, america, military, army, hitler
6	bollywood, indian, kapoor, khan, akshay, fi, amitabh, ramones, verhoeven, christina, sci, braveheart, kumar, chiller
7	film, one, the, scene, character, story, director, much, plot, well, even, work, time
8	film, role, performance, great, play, best, good, cast, one, actor, comedy, john
9	show, series, episode, year, tv, time, great, first, kid, dvd, one, funny, still, watch
10	match, matthau, luke, shakespeare, neil, bruce, scarface, boxing, hamlet, elvis, branagh, lucas, polanski

Table 13: Topic distribution on Ag News dataset

#Topic	Ag News
1	palestinian, said, iraqi, killed, iraq, reuters, attack, baghdad, arafat, israeli, bomb, scored, force, city
2	win, world, first, point, coach, cup, lead, victory, team, second, no, champion, night, final
3	president, afp, said, minister, election, bush, leader, india, state, reuters, prime, united
4	reuters, oil, price, stock, new, search, dollar, google, market, york, rate, apple, share, record
5	court, drug, say, ap, could, may, new, year, eu, case, said, state, scientist, trial
6	space, nasa, canadian, dec, press, former, nba, williams, winter, houston, monday, arsenal, sunday
7	said, company, inc, million, deal, corp, billion, sale, year, percent, reuters, buy, business
8	microsoft, new, software, internet, service, system, computer, technology, phone, ibm, music, online, web, company
9	china, police, said, reuters, people, worker, british, government, official, party, japan, group, chinese
10	game, new, year, red, one, time, season, first, team, series, last, york

Table 14: Topic distribution on Financial dataset

#Topic	Financial
1	company, finnish, new, plant, finland, construction, order, line, contract, service, unit, production, investment
2	company, share, bank, said, also, capital, start, issue, term, financial, price, business, executive, dividend
3	eur, profit, sale, net, operating, million, period, quarter, compared, loss, year
4	finnish, said, today, million, company, first, helsinki, year
5	company, mobile, said, phone, nokia, solution, business, pretax, finland, network, product, group, store, customer
6	market, board, option, company, share, stock, director, member, concerning, meeting, general, bank, flow, chairman
7	share, company, group, lower, helsinki, stock, president, capital, holding, new, right
8	service, finland, customer, corporation, company, electronics, solution, industry, business, helsinki, ltd, group
9	company, expected, sale, said, people, production, paper, year, finland, plant, cut, staff, expects
10	euro, service, company, item, nokia, excluding, technology, business, mobile, device, market, product

Table 15: Topic distribution on Authorship dataset

#Topic	Authorship
1	one, would, may, people, year, even, could, time, last, minister, public, police, many, blair, say
2	one, would, war, farmer, even, new, blair, bush, could, need, time, iraq, much, week
3	labour, new, people, government, tax, year, time, even, public, brown, blair, party, money
4	would, one, government, new, world, year, labour, much, state, blair, last, british
5	new, public, government, labour, people, year, one, would, may, way, time, make, right, life, need
6	people, time, public, said, even, government, lord, like, party, make, day
7	one, bush, american, world, year, right, war, child, people, british, state, new
8	people, one, child, like, time, family, get, year, burrell, may, still, even, much
9	would, one, blair, bush, war, nuclear, even, it, new, make, could, weapon, people, party
10	would, one, year, people, could, even, royal, like, woman, time, war, right, iraq

Table 16: Topic distribution on Wiki Toxic dataset

#Topic	Wiki Toxic
1	page, talk, edit, please, user, edits, wikipedia, editor, comment, block, blocked, editing, discussion, thanks, stop
2	image, use, you, copyright, page, fair, picture, please, medium, wikipedia, see, template, deleted, file, photo
3	article, deletion, deleted, page, please, tag, may, speedy, notable, talk, guideline, subject, wikipedia, criterion, add
4	nigger, hate, bitchfuck, faggot, lol, class, rape, fat, asshole, mama, fucker, hairy, ha, boymamas
5	like, know, get, people, it, think, you, want, one, time, go, thing, me, really
6	state, english, country, american, language, people, name, war, city, world, government, history, british, jew, group
7	fuck, ass, suck, fucking, shit, u, hi, cunt, school, moron, go, bitch, shut, cock, dick
8	utc, year, new, game, redirect, song, old
9	page, wikipedia, talk, help, please, link, welcome, question, article, thank, thanks, like, name, best
10	article, one, would, source, also, think, section, fact, see, it, like, point, say, time, reference

Task	Learning rate	epoch	batch size	warmup	weight decay	adapter size
BERT$_{\text{BASE}}$						
MNLI	4e-4	20	32	0.06	0.1	256
MRPC	4e-4	5	32	0.06	0.1	256
QNLI	4e-4	20	32	0.06	0.1	256
QQP	4e-4	20	32	0.06	0.1	256
RCT	4e-4	20	32	0.06	0.1	256
RTE	4e-4	5	32	0.06	0.1	256
SST2	4e-4	10	32	0.06	0.1	256
Tweets	4e-4	5	32	0.06	0.1	256
IMDB	4e-4	5	32	0.06	0.1	256
Ag News	4e-4	20	32	0.06	0.1	256
Financial	4e-4	5	32	0.06	0.1	256
Authorship	4e-4	5	32	0.06	0.1	256
RoBERTa$_{\text{LARGE}}$						
MNLI	3e-4	20	64	0.6	0.1	64
MRPC	3e-4	5	64	0.6	0.1	64
QNLI	3e-4	20	64	0.6	0.1	64
QQP	3e-4	20	64	0.6	0.1	64
RCT	3e-4	20	64	0.6	0.1	64
RTE	3e-4	5	64	0.6	0.1	64
SST2	3e-4	10	64	0.6	0.1	64
Tweets	3e-4	5	64	0.6	0.1	64
IMDB	3e-4	5	64	0.6	0.1	64
Ag News	3e-4	20	64	0.6	0.1	64
Financial	3e-4	5	64	0.6	0.1	64
Authorship	3e-4	5	64	0.6	0.1	64

Table 17: Hyperparameter configurations for various tasks.

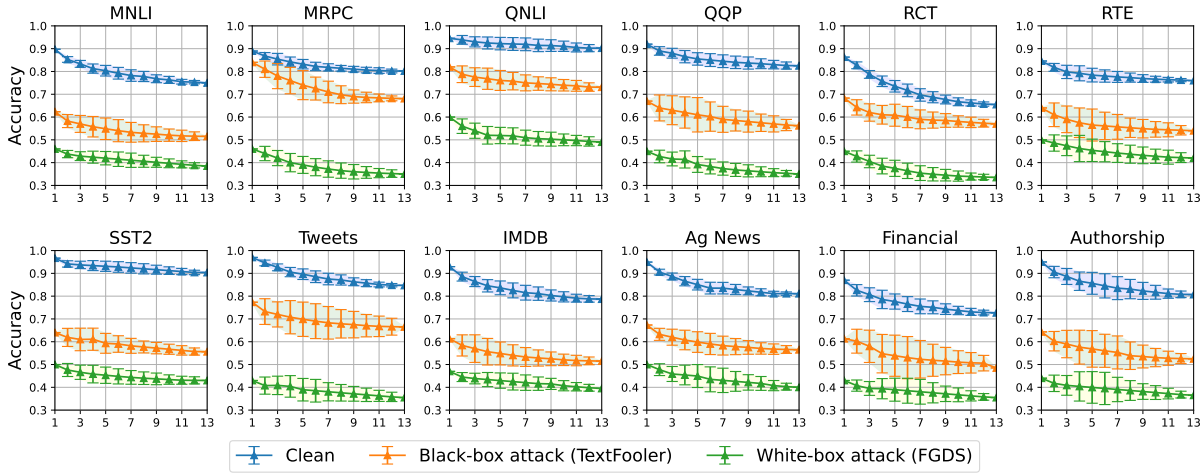

Figure 10: Performance Evaluation of RoBERTa Using the Pfeiffer Adapter across Varied Domain Datasets.

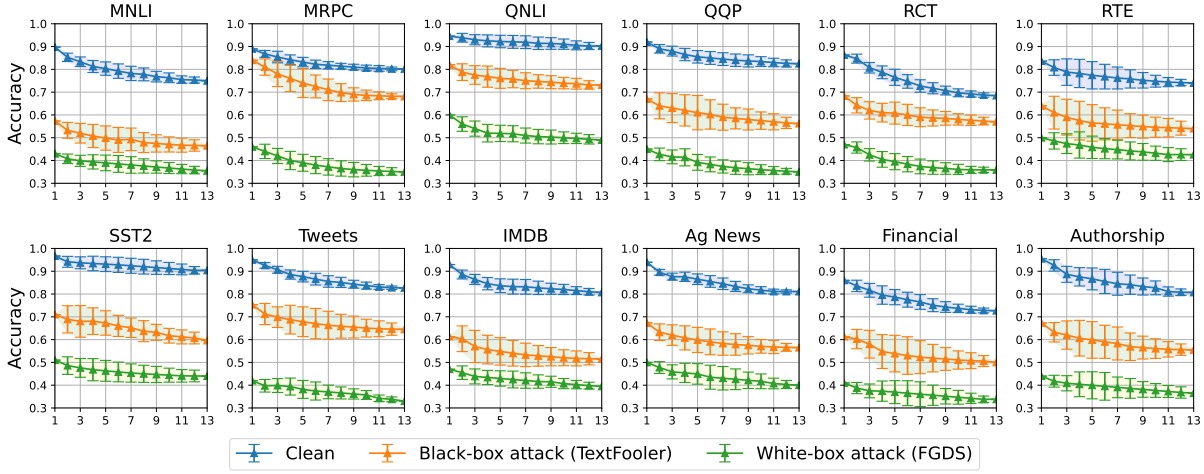

Figure 11: Performance Evaluation of RoBERTa Using the LoRA Adapter across Varied Domain Datasets.

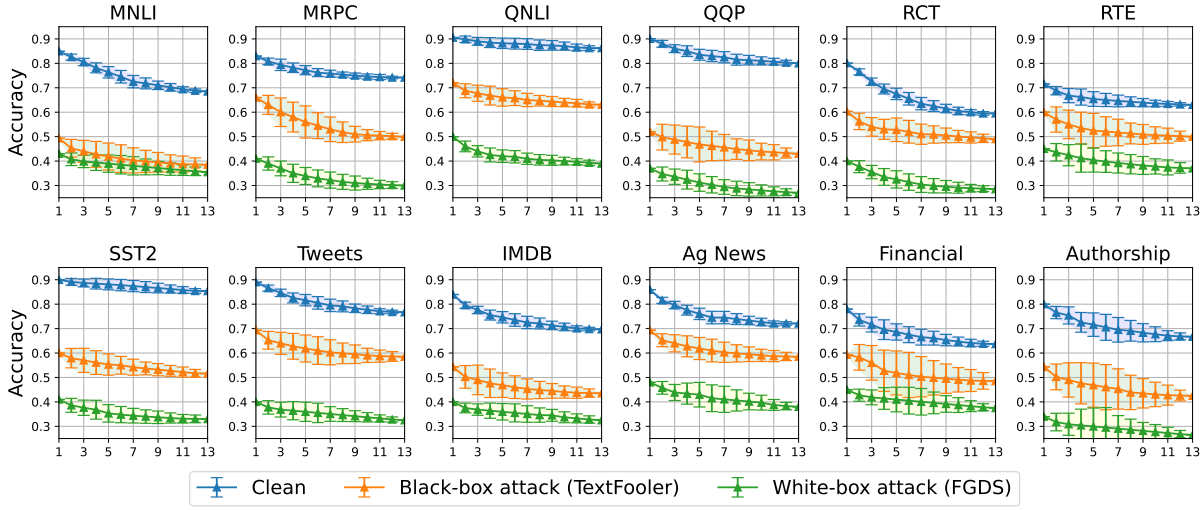

Figure 12: Performance Evaluation of BERT Using the Houlsby Adapter across Varied Domain Datasets.

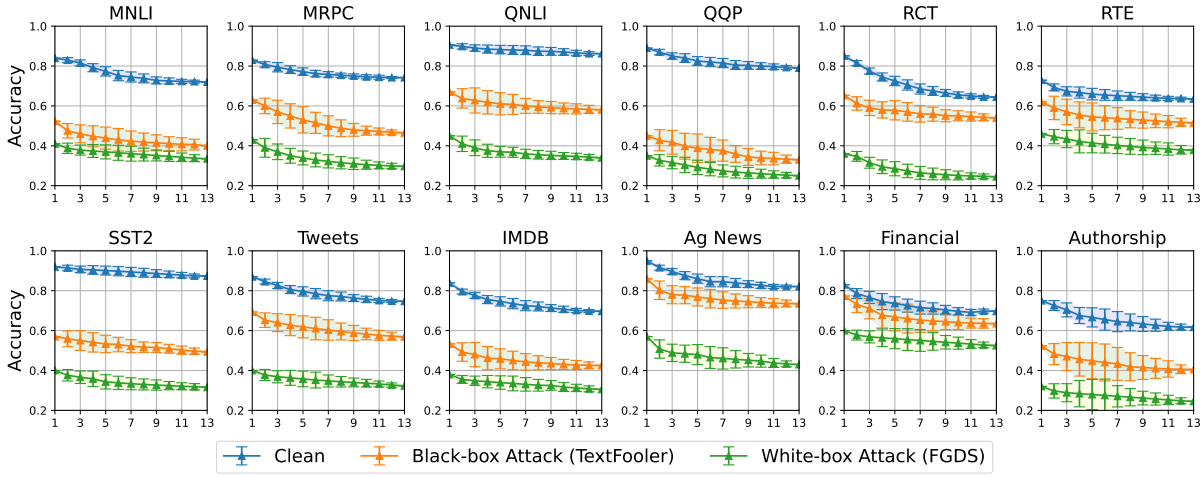

Figure 13: Performance Evaluation of BERT Using the Pfeiffer Adapter across Varied Domain Datasets.