CAST: Sparse Fine-Tuning with Counterfactual Data Augmentation

Anonymous ACL submission

Abstract

In the domain of transfer learning for pre-trained models, fine-tuning specific parameters rather than the entire model has become a prevalent trend. Sparse fine-tuning has proven effective. Counterfactual Data Augmentation have been shown to enhance the generalization ability of models. This study proposes a fine-tuning method that combines the advantages of both approaches, which is called "Counterfactual Augmented Sparse Tuning" (CAST). Inspired by the Lottery Ticket Hypothesis, this method identifies significant parameter changes by comparing models trained on counterfactual data with those trained on original data, thereby constructing a mask table for model parameters. To further enhance model sparsity, we introduce a counterfactual data impact factor, which adjusts the specific influence of counterfactual data on the model training outcomes. The CAST method achieved the best accuracy rates of 90.2% and 76% in counterfactual data augmentation tasks for sentiment analysis and natural language inference tasks. It was observed that CAST successfully resisted catastrophic shifts in dataset distribution. The CAST model not only improves performance in specific NLP tasks but also reduces the risk of data distribution shift and enhances the model's ability to capture key features.

1 Introduction

Introducing transfer learning methods into the field of deep neural network research represents a landmark advancement(Han et al., 2021). Transfer learning has gradually evolved into a two-stage learning framework: the first stage is pre-training, where knowledge is acquired from large datasets; the second stage is fine-tuning, where the pre-trained network architecture is aligned with downstream tasks using a small amount of data. The models derived from the pre-training phase are commonly known as pre-trained models (PTMs). PTMs, having been trained on a vast amount of data, can quickly adapt to new tasks, reducing the time and resources required to train models from scratch. Since pre-trained models have learned general features on a wide range of datasets, they typically offer better performance on specific tasks. For tasks with limited data, pre-trained models can significantly enhance model performance, as they have already learned rich features from a large amount of data.

The NLP community has recognized the potential of PTMs and has begun developing PTMs suitable for NLP tasks(Qiu et al., 2020). In the field of NLP, PTMs are generally referred to as Pre-trained Language Models(PLMs). These models are usually pre-trained on large-scale text data through unsupervised learning. The fine-tuning stage is generally divided into two types: full-parameter fine-tuning, where the entire model's parameters are adjusted during training; parameter-efficient fine-tuning (PEFT), where only a subset of the model's parameters are adjusted during training. Sparse fine-tuning (SFT) is a type of PEFT, which is inspired by the Lottery Ticket Hypothesis (LTH, Frankle and Carbin 2019), which suggests that there are redundant parameters in neural networks, allowing for the pruning of some parameters during training while maintaining the model's performance.

Data Augmentation is a technique in machine learning and deep learning used to increase the quantity and diversity of available data by generating new variants from existing datasets. This method is particularly suitable for tasks with limited data and can help improve the model's generalization ability and reduce the risk of overfitting. For text data, diversity can be increased through methods such as synonym replacement, random insertion, deletion, or swapping of words. Introducing Counterfactual Data Augmentation (CDA) into the fine-tuning stage is a promising research direction(Kaushik et al., 2020; Zmigrod et al., 2019).

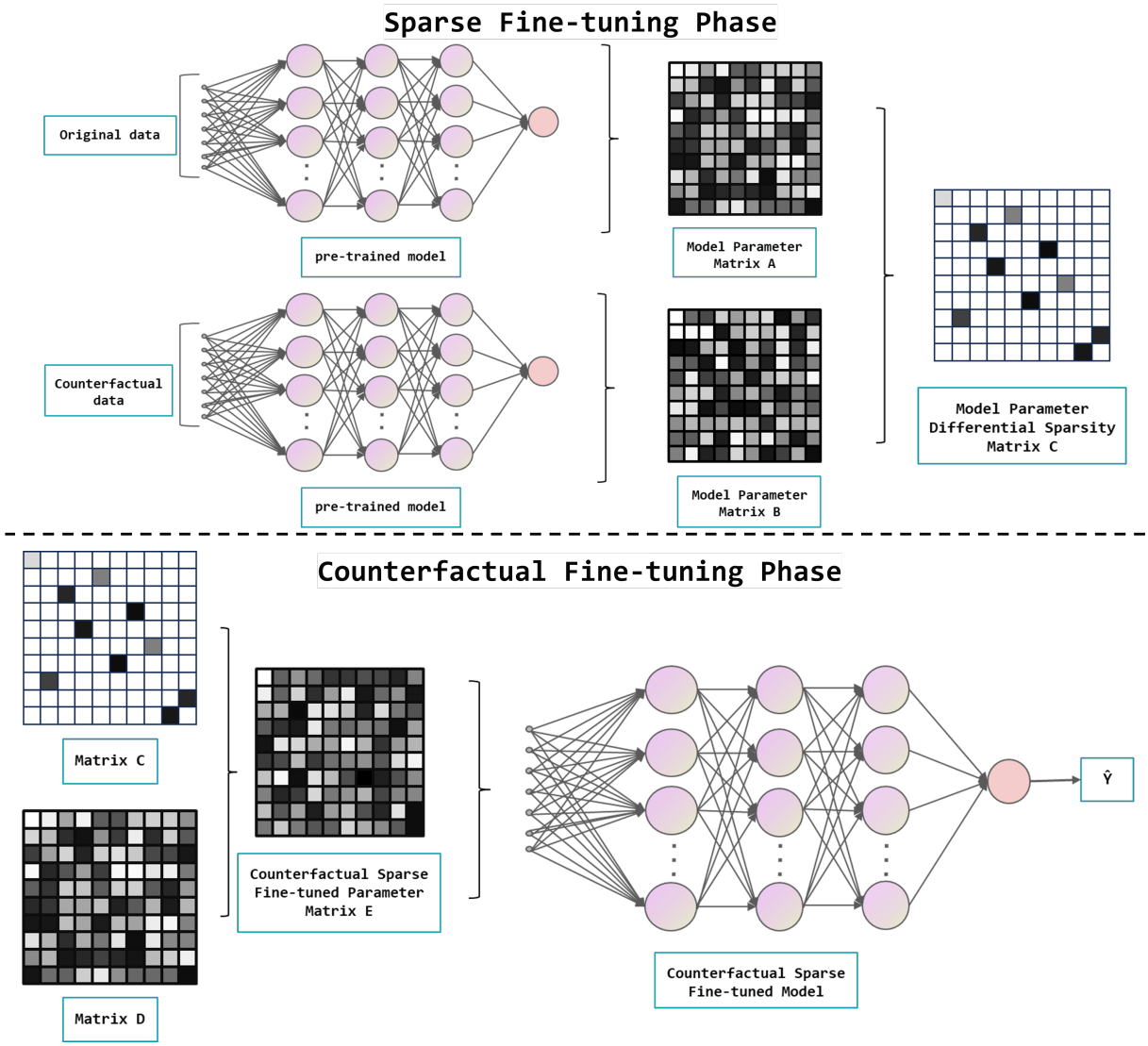

Figure 1: Framework of CAST Algorithm. Sparse fine-tuning: CAST trains the PTM with original data and counterfactual data to obtain the parameter matrix A and matrix B. Based on matrices A and B, CAST calculates matrix C with a specific method. Counterfactual fine-tuning: CAST integrates the sparse parameter matrix C with the matrix D (which is the matrix A after rollback) to calculate the counterfactually fine-tuned sparse parameter matrix E for downstream task computations. For detailed description, refer to the "Method" section.

The fundamental idea of CDA is to enhance the diversity of training data by constructing hypothetical counterfactual scenarios, thereby improving the generalization capability and robustness of the model.

In our study, we propose a method called "Counterfactual Augmented Sparse Tuning" (CAST). We provide a graphical representation of our method in Figure 1, illustrating the key components and an overview of the workflow in our research approach. For more detailed information, please refer to the 'Method' section.

CAST combines counterfactual data augmentation with sparse fine-tuning techniques, leveraging the structural symmetry of counterfactually augmented data to enhance model performance, while using the selective parameter resistance of the sparse fine-tuned model to counteract the data distribution skew caused by counterfactual data augmentation. The addition of counterfactual data structurally enhances the dataset. By designing a duality in the fine-tuning strategy, it improves the model's recognition of key features. Our research indicates that simply augmenting the dataset with counterfactual data can lead to severe data distribution shifts, especially in complex scenarios. However, the sparse processing method we adopt effectively mitigates this risk. Experiments

demonstrate that our method achieves significant performance improvements across various natural language processing tasks, further proving its effectiveness in practical applications.

2 Background

PLMs The earliest PLMs in the field of natural language processing can be traced back to shallow networks pre-trained to capture the semantic meanings of words, such as the Word2Vec(Mikolov et al., 2013). The Transformer architecture(Vaswani et al., 2017) made it possible to train deep network models for NLP tasks. Representative deep neural network models based on the Transformer architecture, such as BERT(Devlin et al., 2019), created a significant impact in the NLP community in 2018. With the advancement of distributed computing capabilities, the GPT-3 (Brown et al., 2020), which has hundreds of billions of parameters, emerged. This model is considered to have the potential for few-shot learning. The approach of fine-tuning PLMs for downstream tasks, rather than training language models from scratch, has gained increasing recognition(Dai and Le, 2015; Howard and Ruder, 2018).

CDA Existing research has demonstrated that training models using counterfactually augmented data (CAD) can significantly enhance model robustness(Sen et al., 2022). Previous findings reveal that training on CAD not only improves the model's generalization ability across different domains but also that the performance gains primarily stem from the model's learning of core features rather than merely relying on specific patterns within the dataset(Kaushik et al., 2020; Samory et al., 2021). However, it remains unclear how learning these core features affects the model's misclassification issues when the role of these features changes in specific contexts.

PEFT As models become increasingly large, full-parameter fine-tuning on consumer-grade hardware has become prohibitively expensive, drawing researchers' attention to PEFT(Treviso et al., 2022). Peters et al. (2018) proposed a technique for adapting PTMs to downstream tasks by introducing a new classification layer while keeping the other model parameters fixed. Although this approach is effective in certain cases, its generalization performance across multiple tasks is suboptimal. Another approach, Adapters(Houlsby et al., 2019), enhances model adaptability by injecting trainable new layers into the PTM without altering the original model parameters. While this method reduces the number of parameters needed for training, it also increases model complexity by introducing new structures.To avoid changing the model structure, some studies have proposed methods that directly adjust the learned vectors, which are conceptually similar to our research.

If ϕ is a sparse vector, then we define $F' = F(\cdot; \theta + \phi)$ as the SFT of the pre-trained neural network model $F(\cdot; \theta)$(Ansell et al., 2021). Current SFT methods include DiffPruning(Guo et al., 2021), BitFit(Ben Zaken et al., 2022), and LT-SFT(Ansell et al., 2021). DiffPruning simulates the sparsity of the difference vector during training by continuously relaxing a binary mask. BitFit only allows non-zero differences in the bias parameters. LT-SFT combines the core ideas of adapters and sparse fine-tuning. We believe that simply concatenating(Liu et al., 2023; Li and Liang, 2021; Lester et al., 2021), multiplying(Liu et al., 2022), or adding(Ben Zaken et al., 2022) the learning vectors is insufficient to fully exploit the inherent potential and complexity of these vectors.

LTH LTH(Frankle and Carbin, 2019; Malach et al., 2020) provides a possible explanation for SFT: within a randomly initialized network, there exists a subnetwork (the "winning ticket") that can achieve performance comparable to the original network after proper training. This theory has inspired extensive research(Ansell et al., 2021) into how sparse techniques can be used to discover and train these efficient subnetworks.

3 Method

The CAST method achieves refined adjustments of model parameters by controlling the changes in model parameters obtained from training on the original dataset and the counterfactual dataset. Referencing the algorithm proposed by LTH, The CAST method is divided into two stages: the SFT phase and the counterfactual fine-tuning phase.

3.1 SFT

Firstly, we use the same PLM and perform full parameter training on both the factual dataset and the counterfactual dataset to complete the sentiment analysis tasks. At the beginning of each training epoch, the current model parameters are denoted as θ^{ori}. This process aims to obtain two sets of

model parameters, which we denote as θ^{fac} and $\theta^{counter}$. Next, we compare these two sets of parameters by calculating their absolute differences $\Delta\theta$. The specific calculation formula is shown below as Equation 1:

$$\Delta\theta = |\theta^{counter} - \theta^{fac}| \quad (1)$$

We have sorted the model parameters according to the magnitude of their differential values$\Delta\theta$ and selected the top Kparameters, referred to as θ^K. These parameters are pivotal in discerning the nuances within counterfactual datasets. Additionally, we have formulated a parameter mask matrix, represented as M^K, where the elements corresponding to θ^K are set to 1, indicating their significance, while the remaining parameters are assigned a value of 0.

We posit that θ^K is indispensable for capturing the intrinsic features of counterfactual data and is exceptionally responsive to counterfactual signals. Drawing from the LTH, these parameters are potential candidates for the "winning ticket," suggesting a greater probability of forming a robust and efficient subnetwork within the model architecture. The parameters identified in θ^K are slated for subsequent fine-tuning stages to enhance their efficacy further.

3.2 Rollback

To enhance the sparsity of the model, we have devised a parameter rollback strategy. Our goal is to retain the influence of parameter θ^K on the model while eliminating updates to other parameter values. Based on M^K, we calculate a sparsely updated parameter matrix θ^{rol} according to Equation 2.

$$\theta^{rol} = \theta^{ori} + (\theta^{fac} - \theta^{ori}) \cdot M^K \quad (2)$$

The significance of this formula is that the parameter corresponding to θ^K has undergone training and updating with factual data, while the other parameters have been rolled back to their state before the training began. The core objective of this strategy is to mitigate any adverse effects that the counterfactual dataset may have on the model, ensuring the stability of the model's performance when dealing with the original tasks.

3.3 Counterfactual Fine-tuning

Based on a hyperparameter α, which represents the model's learning degree from the counterfactual data, the changes in the model parameters after CAST are calculated according to the following formula, where θ represents the parameters of the PLM after fine-tuning :

$$\theta = \theta^{rol} + \alpha \cdot (\theta^{counter} - \theta^{fac}) \cdot M^K \quad (3)$$

After the computation using Equation 3, we obtain a set of parameters, which are combined with those parameters rolled back during the training process to constitute the current state of the model. Subsequently, through meticulous selection of learning rates and precise adjustments using the backpropagation algorithm, the model's performance is further enhanced. The completion of this series of steps signifies the successful conclusion of one full iteration of the CAST algorithm.

4 Experiment

We designed two experiments for this study:

1) Sentiment Analysis Experiment: This experiment aims to fine-tune the two hyperparameters of the model to determine the optimal values for the counterfactual data learning rate α and the model sparsity parameter K. We will apply the CAST algorithm on a sentiment analysis dataset to evaluate its performance and identify the hyperparameter combination that yields the best sparsity effect.

2) Transfer Learning Test: After determining the optimal hyperparameter settings for sentiment analysis tasks, we plan to apply these parameters to natural language inference (NLI) tasks to examine the generalization ability of the CAST algorithm. Additionally, this test will explore the potential issue of significant data distribution shift caused by counterfactual data augmentation in different NLP scenarios and seek effective strategies to maintain the stability and performance of the model during the transfer process.

4.1 Data Construction

The datasets used for the sentiment analysis experiment and the transfer learning test in the CAST model are the sentiment analysis dataset and the NLI dataset proposed by Kaushik et al. (2020).

Each data record of the sentiment analysis dataset consists of a piece of movie review text and its corresponding sentiment label, categorized as either negative or positive. The number of records in the training set, validation set, and test set of this dataset are 1707, 245, and 488. To better adapt to the CAST model in this study, we integrated

Version	Premise	Hypothesis	Label
RP_1	A man in a blue jacket riding a purple bike.	A man is on his yellow bike.	contradiction
RP_2	A man in a blue jacket riding a yellow bike.	A man is on his yellow bike.	entailment
RAW	A man in a blue jacket riding a bike.	A man is on his yellow bike.	neutral
RH_1	A man in a blue jacket riding a bike.	A man wearing blue is on a bike.	entailment
RH_2	A man in a blue jacket riding a bike.	A man wearing a red jacket is on his bike.	contradiction

Table 1: Example Table of NLI Dataset Division. The NLI dataset is divided into four sub-datasets based on their origin and augmentation method: 'Raw' for the original data, and 'RP' and 'RH' for counterfactual data augmentation applied to the premise and hypothesis, respectively. Given that NLI is a three-class task, both 'RP' and 'RH' are differentiated by two subscripts. Combining "RAW" with the remaining four versions can yield four sub-datasets.

the original movie review dataset with its corresponding counterfactual version dataset. During the merging process, we expanded each data record to include not only the original movie review text but also a counterfactual movie review text field. Through this intrecordegration, we obtained a more comprehensive dataset.

The NLI task involves classifying two sentences—a premise and a hypothesis—to determine the logical relationship between them, which may include entailment, contradiction, and neutral. The number of records in the training set, validation set, and test set of this dataset are 1750, 250, and 500. The version obtained by modifying the premise is denoted as "Revised Premise" (RP), and the version obtained by modifying the hypothesis is denoted as "Revised Hypothesis" (RH). The improvements made by us to the NLI dataset were similar to those made to the sentiment analysis dataset.

It is worth noting that as the NLI dataset entails annotating three relationships for two sentences, merging the counterfactual data resulted in four new datasets, denoted as RH_1, RH_2, RP_1, and RP_2. Refer to Table 1 for the specific rules on dataset division.

4.2 Model

In the sentiment analysis experiment, we employed the BERT-base model as the pretraining foundation for the CAST model. To better adapt the BERT model to the sentiment analysis task, we introduced a Dropout activation layer at the end of the BERT model, followed by a linear classification layer, to enhance the model's capability for sentiment classification. In the data preparation phase before model training, we set the maximum input length to 300 tokens and performed padding on texts with insufficient length to ensure consistent input data. For training settings, we utilized the Adam(Kingma and Ba, 2014) optimizer and trained all models for a maximum of 100 epochs. We adopt the early stopping strategy, with the patience parameter set to 10, monitoring after each epoch, and a minimum improvement threshold of 0.001. During training, we set the learning rate to 1e-5 and used a batch size of 128. To utilize the training process more efficiently, we employed a linear learning rate scheduler.

In the next experiment, we employed the same model settings as the first. We changed the linear layer at the end of the BERT model , with its dimension modified to (768, 3), to adapt to the NLI three-category classification task.

4.3 Baselines

For our study on sparse fine-tuning techniques, we have chosen BitFit, Adapters, LoRA, and LT-SFT as our comparative baselines, alongside the performance of a BERT model in its unaltered state. To tailor the BERT model for both binary and multi-class classification tasks, we have introduced an additional dropout layer followed by a linear layer at the model's output. Within the BitFit methodology, we have opted to freeze all parameters except for the bias terms. For the Adapters and LoRA approaches, we have strictly limited the fine-tuning to around 1%(Since CAST performs best at a sparsity level of 1%) of the model's parameters. Despite LT-SFT's original intent to tackle Cross-Lingual Transfer challenges, its structural similarity to our model framework positions it as our primary point of reference among the baselines.

4.4 Experimental Setup

In the sentiment analysis experiment, we set the K value range between 10^4 and 10^7 and selected the α value range between 0.01 and 0.15. To establish a performance baseline, we employed two methods: one using the BERT directly on the original dataset for sentiment analysis without any counterfactual data augmentation (under the index "No CAD" in the tables), and the other performing sentiment analysis on the counterfactually augmented

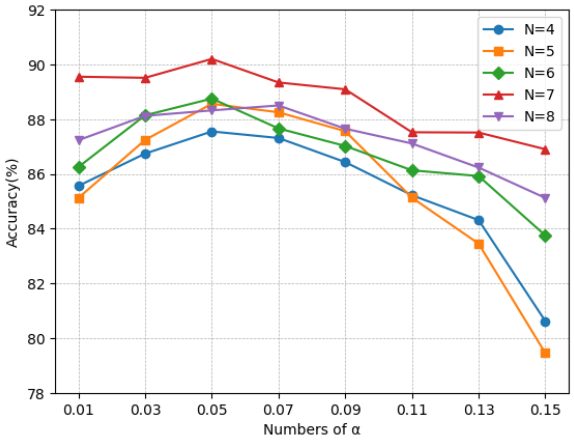

Figure 2: CAST Algorithm Fine-tuning Results Line Chart of Sentiment Analysis Experiment: In the legend section, the index of the sparsity parameter K for N, where $K = 10^N$.

Method	No CAD(%)	CAD (%)
BERT	87.2	88.4
CAST(0.05, 10^7)	-	**90.2**
BitFit	57.4	44.5
Adapters	68.5	60.8
LoRA	63.7	65.2
LT-SFT	80.6	77.6

Table 2: Sentiment Analysis Experiment Prediction Accuracy Table. For each fine-tuning algorithm, predictions were trained on both the original data and the counterfactual data augmentation. CAST inherently includes counterfactual data, so there are no results for training on the original data alone.

dataset(under the index "CAD" in the tables). We selected accuracy as the main metrics to evaluate model performance.

In the transfer learning test, to ensure the rationality of the transfer learning test and verify the generalizability of the CAST model, we decided to use the optimal parameter values determined in the sentiment analysis experiment. Considering that after merging the counterfactual data, we would obtain four new counterfactual datasets.

All experiments were conducted with at least five repetitions, and the average values were taken.

5 Results and Discussion

5.1 Sentiment Analysis Experiment

Based on the experimental results presented in Figure 2, we can conclude that the optimal choice for K is 10^7, while the best choice for α is 0.05. From the data presented in the graph, it becomes evident that the parameter K, dictating the extent of parameter involvement in numerical updates, exerts a more pronounced influence on the model's accuracy. Notably, the peak performance across the entire spectrum of α values was consistently attained at $K = 10^7$. When N takes the value of 8, the optimal point of model performance is associated with an alpha value of 0.07. In contrast, for all other values assigned to N, the peak model performance aligns with an alpha value of 0.05. Independent of N's influence, the model performance exhibits a consistent pattern of ascent to its zenith with the increment of alpha, beyond which it descends, indicating that an increase in alpha value leads to a degradation in performance post the point of optimality.

5.1.1 The Coordinate System of CAST

The variable N signifies the degree of sparsity control within the CAST method, where a smaller N value corresponds to a sparser model. In CAST, model sparsity is intricately balanced by the dual regulation of parameters α and K. Observations from the provided graph illustrate that model performance does not linearly correlate with changes in sparsity, irrespective of whether N is held constant or α is fixed. The role of K in model training is more nuanced; it operates on a horizontal axis, determining the quantity of parameters that participate in training during an epoch. In contrast, α exerts a vertical influence, precisely modulating the extent to which the top K parameters are affected by counterfactual data augmentation. CAST has thus established a sparse model parameter coordinate system, with coordinates being directed and finely tuned by α and K. However, the generalizability of this conclusion is limited because the values of α tested in the experiment are discrete, whereas the theoretically optimal value might lie within the continuous range between these discrete values.

5.1.2 CAD-induced Data Distribution Shift

The research conducted by Sen et al. 2022 has demonstrated that CAD can improve the generalization capabilities of models. This conclusion is corroborated by the experimental outcomes associated with the BERT and LoRA methodologies. Yet, the experimental outcomes for the BitFit, Adapters, and LT-SFT methods presented in Table 2 indicate lower predictive accuracies with CAD than in the absence of CAD ("No CAD" condition). This

discrepancy can be attributed to the fact that the incorporation of counterfactual data leads to a distributional shift within the dataset. Data augmentation carries the risk of skewing the model's focus excessively towards specific classes or characteristics, which can lead to overfitting and a subsequent decline in model performance. Counterfactual data augmentation fundamentally involves the modification of a subset of critical terms within the original data, prompting a reversal of the associated labels. Consequently, the augmented dataset experiences a diminished occurrence of these pivotal terms, thereby causing a distributional shift that can impact the model's ability to generalize effectively.

The resistance to data distribution shifts exhibited by LoRA is particularly striking and merits attention. This could be due to the symmetrical matrix compression characteristic of the LoRA algorithm, a feature that aligns with the CAST method's own countermeasures against the adverse effects of data skew. The inherent symmetry in LoRA's approach may provide a structural basis for its observed resilience, mirroring the protective mechanisms employed by CAST to mitigate the impact of data distribution shifts.

5.1.3 CAST and Baselines

The CAST method demonstrated a remarkable prediction accuracy of 90.2% at the optimal parameter values, outperforming all other approaches and thereby highlighting its inherent advantages. Additionally, the BERT model consistently secured the top position in baseline comparisons, irrespective of the presence or absence of counterfactual data augmentation. This observation implies that existing fine-tuning methods may not be optimally adapted to the nuances of counterfactually augmented fine-tuning. It suggests a shortfall in exploiting the structural symmetry between counterfactual and factual data, indicating a potential area for further refinement in the fine-tuning strategies.

It is important to note that results may vary with different datasets or experimental conditions, and in some cases, they may even be completely opposite. Indeed, an experiment might show trends entirely different from the current one, with significant declines in various metrics. This variability emphasizes the importance of considering the effectiveness and applicability of data augmentation methods, particularly counterfactual data augmentation, in different contexts.

In summary, while the specific values of K and α showed good results in this experiment, further validation on a broader range of experimental settings and different datasets is necessary. Additionally, the variability in experimental results reminds us to carefully consider the specific impact of data augmentation techniques on model performance and to flexibly adjust strategies according to the actual situation.

Method	**N.C(%)**	**CAD(%)**			
		RP_1	RP_2	RH_1	RH_2
BERT	72.2	70.3	**72.1**	74.2	70.8
CAST	-	**72.0**	71.8	**76.0**	**73.5**
BitFit	43.5	34.2	35.3	42.6	37.8
Adapters	68.2	59.2	60.1	64.8	60.7
LoRA	58.1	54.8	55.3	60.7	55.9
LT-SFT	70.3	63.4	62.6	66.1	63.5

Table 3: Results of the Transfer Learning Test. The two parameters α and K for CAST are 0.05 and 10^7. 'N.C' means 'No CAD'. 'RP' stands for 'revised premise', and 'RH' stands for 'revised hypothesis'. Both 'RH' and 'RP' are branches of counterfactual data augmentation in the NLI task.

5.2 Transfer Learning Test

The results of transfer learning test are shown in Table 3. The CAST method once again demonstrated its outstanding performance. While the inference accuracy of the CAST method in RP_2 is lower than that of the BERT model, ranking it second among all methods, the CAST method achieves the best inference performance on the other three sub-datasets. Notably, the RH_1 model achieved an accuracy of approximately 76%, the highest among all models. These results indicate that the CAST model not only possesses strong generalization capabilities but also that the optimal hyperparameter choices derived from training on the sentiment analysis dataset effectively enhance performance in the NLI task. Furthermore, this confirms the significant advantage of the sparse processing method employed by the CAST model in the field of CAD.

5.2.1 Data Distribution Shift in NLI

In the first experiment, we discovered that training directly with a combination of raw and counterfactually augmented data, without any sparse processing, could lead to a significant drop in model performance. This discovery gained further validation in the follow-up experiment, which entailed an assessment of various baseline models. Notably, a

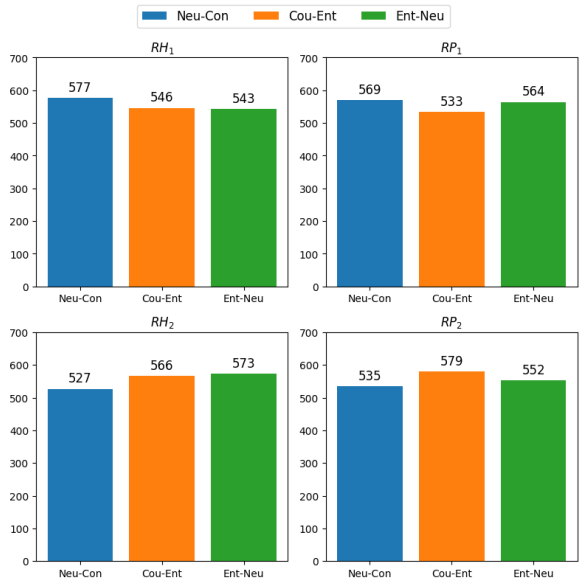

Figure 3: The Distribution of Label Pairs. In the NLI task, the distribution of label pairs for CAD across the four sub-datasets shows in this figure. "Neu-Con" represents the pair neutral and contradiction. "Cou-Ent" represents the pair contradiction and entailment. "Ent-Neu" represents the pair entailment and neutral.

decline in performance was also discernible within the BERT model, underscoring the broad implications of the observed phenomenon. We believe this decline in performance may be attributed to the nature of the NLI task, which involves predicting the relationship between a premise and a hypothesis, typically classified into three categories. When counterfactually augmenting the data, simply training with one set of counterfactual data can result in a shift in data distribution.

To mitigate the risk of uneven distribution of counterfactual data label pairs during the manual subdivision of sub-datasets, we conducted a tally of label pair counts across the four NLI sub-datasets, as depicted in Figure 3. From the graph, it is evident that the largest disparity exists in task RH_2, with a difference of 46 pairs between Ent-Neu and Neu-Con, comprising less than 3% of the total data volume. Consequently, any potential data offset stemming from uneven label pair distribution can be disregarded.

5.2.2 Sparsity and CAD

Through sparse processing, we can leverage the enhanced knowledge provided by the counterfactual data while minimizing the catastrophic shift in data distribution. In experiments combining multiple sparse fine-tuning methods with CAD, where a performance decline was observed, CAST surpasses the BERT model across multiple sub-datasets. This superiority stems from its consideration of the structural symmetry inherent in the fine-tuning process, specifically tailored to counterfactual enhancement data. By combining and fine-tuning a set of mutually counterfactual data, CAST effectively screens out winning ticket subnetworks, resulting in parameter updates solely based on these subnetworks. This approach mitigates the data distribution shift induced by counterfactual data enhancement. Further maintenance of the original data distribution is achieved through the introduction of an influential factor, the alpha coefficient.

6 Conclusion

We propose a novel sparse fine-tuning method that reinforces the duality structure of counterfactually augmented data, aiming to gain a deeper understanding and utilization of the characteristics of learning vectors. Our proposed CAST model offers several advantages:

Enhanced Key Feature Identification: During training, we use a strategy that strengthens key feature recognition by blending the absolute parameter changes from counterfactual training with those from the original data in each epoch. This approach guides the model to focus more on crucial phrases and features. This helps the model to more acutely detect elements essential for understanding the full text. Additionally, this method, which combines counterfactual and original data, also improves the model's sensitivity to subtle linguistic details.

Reduced Risk of Data Shift: The CAST model tackles the issue of data distribution shifts caused by mixing counterfactual data with the original dataset by introducing a control coefficient. We also apply a sparse processing technique, keeping only the top K most changed parameters and resetting the rest to their original state. This minimizes the risk of data shift and maximizes the benefits of counterfactual data. As a result, the CAST model can integrate new insights without losing the original dataset's characteristics, improving its comprehension and generalization of linguistic phenomena. This approach boosts the model's precision in key feature detection and its sensitivity to language nuances, leading to more stable and reliable performance in complex NLP tasks.

7 Limitations

The CAST method has a few drawbacks: it adds algorithmic complexity due to parameter sparsification, requiring sorting and computation of changes, which can be computationally intensive. However, once the parameter mask is created, the fine-tuning process becomes more efficient, somewhat mitigating the complexity. Additionally, handling large PTMs demands substantial GPU memory, which might necessitate model compression or enhanced computing resources.

Despite these, CAST's compatibility with existing algorithms is a significant advantage, allowing for the integration of various techniques to improve training efficiency and model optimization.

References

Alan Ansell, E. Ponti, Anna Korhonen, and Ivan Vulic. 2021. Composable sparse fine-tuning for cross-lingual transfer. In *Annual Meeting of the Association for Computational Linguistics*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3079–3087, Cambridge, MA, USA. MIT Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Jonathan Frankle and Michael Carbin. 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*. OpenReview.net.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *ArXiv*, abs/2106.07139.

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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, abs/2101.00190.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6682–6691. PMLR.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872 – 1897.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021. "call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. In *Proceedings of the international AAAI conference on web and social media*, volume 15, pages 573–584.

Indira Sen, Mattia Samory, Claudia Wagner, and Isabelle Augenstein. 2022. Counterfactually augmented data and unintended bias: The case of sexism and hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4716–4726, Seattle, United States. Association for Computational Linguistics.

Marcos Vinícius Treviso, Tianchu Ji, Ji-Ung Lee, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Pedro Henrique Martins, André F. T. Martins, Peter Milder, Colin Raffel, Edwin Simpson, Noam Slonim, Niranjan Balasubramanian, Leon Derczynski, and Roy Schwartz. 2022. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.