Curriculum Debiasing: Toward Robust Parameter-Efficient Fine-Tuning Against Dataset Biases

Mingyu Lee[1] **Yeachan Kim**[1] **Wing-Lam Mok**[1] **SangKeun Lee**[1,2]
[1] Department of Artificial Intelligence, Korea University, Seoul, Republic of Korea
[2] Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea
{decon9201, yeachan, wlmokac, yalphy}@korea.ac.kr

Abstract

Parameter-efficient fine-tuning (PEFT) addresses the memory footprint issue of full fine-tuning by modifying only a subset of model parameters. However, on datasets exhibiting spurious correlations, we observed that PEFT slows down the model's convergence on unbiased examples, while the convergence on biased examples remains fast. This leads to the model's overfitting on biased examples, causing significant performance degradation in out-of-distribution (OOD) scenarios. Traditional debiasing methods mitigate this issue by emphasizing unbiased examples during training but often come at the cost of in-distribution (ID) performance drops. To address this trade-off issue, we propose a Curriculum Debiasing framework that presents examples in a *biased-to-unbiased* order. Our framework initially limits the model's exposure to unbiased examples, which are more difficult to learn, allowing it to first establish a foundation on easy-to-converge biased examples. As training progresses, we gradually increase the proportion of unbiased examples in the training set, guiding the model away from reliance on spurious correlations. Compared to the original PEFT methods, our method accelerates convergence on unbiased examples by approximately twofold and improves ID and OOD performance by 1.2% and 8.0%, respectively.[1]

1 Introduction

Natural language processing (NLP) has achieved remarkable success across a wide range of downstream applications, largely due to the advent of large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Brown et al., 2020; He et al., 2023). They typically pre-train transformer architecture (Vaswani et al., 2017) on large corpora, followed by fine-tuning the entire pre-trained parameters to adapt the models to specific downstream tasks. Despite its advantages, fine-tuning comes with a significant downside; each application requires storing parameters equivalent to the original model. This issue has become increasingly challenging for deploying models in resource-constrained real-world scenarios, especially as larger models are released at an ever-increasing pace (Treviso et al., 2023).

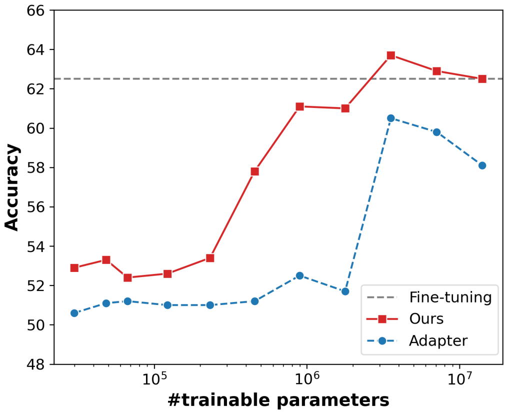


Figure 1: In most settings, the Adapter performs similarly to random guessing. We train models on the MNLI training set (Williams et al., 2018) and measure their OOD performance on HANS (McCoy et al., 2019). As a backbone, we leverage $\text{BERT}_{\text{Base}}$ (Devlin et al., 2019).

To address this challenge, parameter-efficient fine-tuning (PEFT) has been proposed, which adjusts only a part of the model's parameters (Zaken et al., 2022) or introduces external modules (Houlsby et al., 2019; Lester et al., 2021; Mahabadi et al., 2021) for downstream tasks. With these approaches, we only need to store and load a small number of parameters during deployment, significantly reducing memory footprint.

However, we find that PEFT methods significantly impair model generalization in training environments dominated by spurious correlations. As shown in Figure 1, an Adapter trained on a biased

[1] Our code is available at https://github.com/KoreaMGLEE/curriculum_sampling/

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9524–9540
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

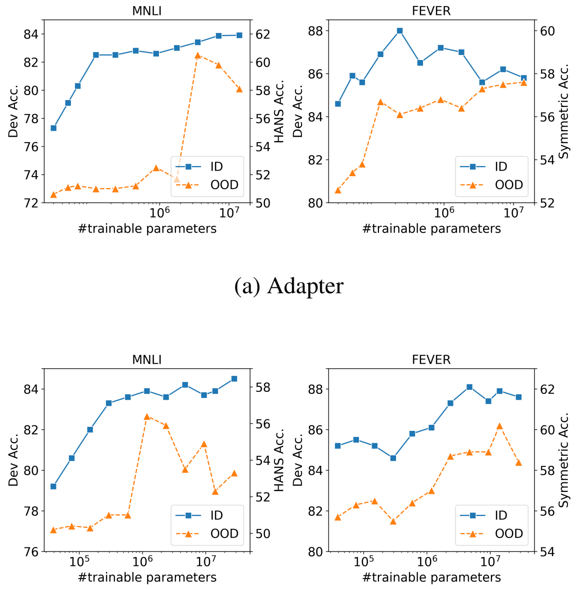
dataset fails to generalize, exhibiting performance close to random predictions across most parameter settings in the out-of-distribution (OOD) evaluation. This suggests that applying PEFT to adjust models on biased datasets may hinder their learning of the intended task features[2], which are critical for generalization. This issue is particularly concerning in real-world scenarios, where biased training datasets are more prevalent than unbiased ones. Despite its significance, the behavior of PEFT in biased training environments and its impact on model generalization remains underexplored in previous studies (Hu et al., 2022; Dettmers et al., 2023; Fu et al., 2023; Xie and Lukasiewicz, 2023; Zhang et al., 2024).

Our pilot study in Section 2 reveals that the deterioration in generalization capability stems from the strong regularization imposed by PEFT. While PEFT does not hinder convergence on biased data, it significantly slows down learning on unbiased data. Consequently, the model overfits to biased data during training and relies on spurious correlations when performing the task.

To address the issue, we introduce a new training framework, dubbed CURRICULUM DEBIASING, to improve the PEFT's generalization capability in biased training environments. Inspired by curriculum learning (Bengio et al., 2009), our framework improves the generalization of PEFT by presenting examples in a *biased-to-unbiased* order. It delays exposure to unbiased examples, which are known to be difficult to learn from (Utama et al., 2020; Sanh et al., 2021), allowing the model to first build a foundation more easily before gradually learning more challenging ones. Subsequently, our framework reduces training on biased examples in later stages, encouraging the model to rely more on intended task features. Experimental results show that our method improves the ID and OOD performance of the Adapter by 1.2% and 8.0%, respectively.

Our contributions are summarized as follows:

- We demonstrate that PEFT hinders the model's learning of unbiased examples and significantly worsens OOD performance.
- We propose a CURRICULUM DEBIASING framework that presents training examples in a *biased-to-unbiased* order to enhance the model's generalization.



(a) Adapter

(b) LoRA

Figure 2: The impact of the number of parameters of PEFT methods on ID and OOD performance. The solid line represents ID performance and the dashed line represents OOD performance.

- We validate the general applicability of our proposed framework across different architectures, model scales, and diverse tasks, demonstrating its practical utility in a wide range of applications.

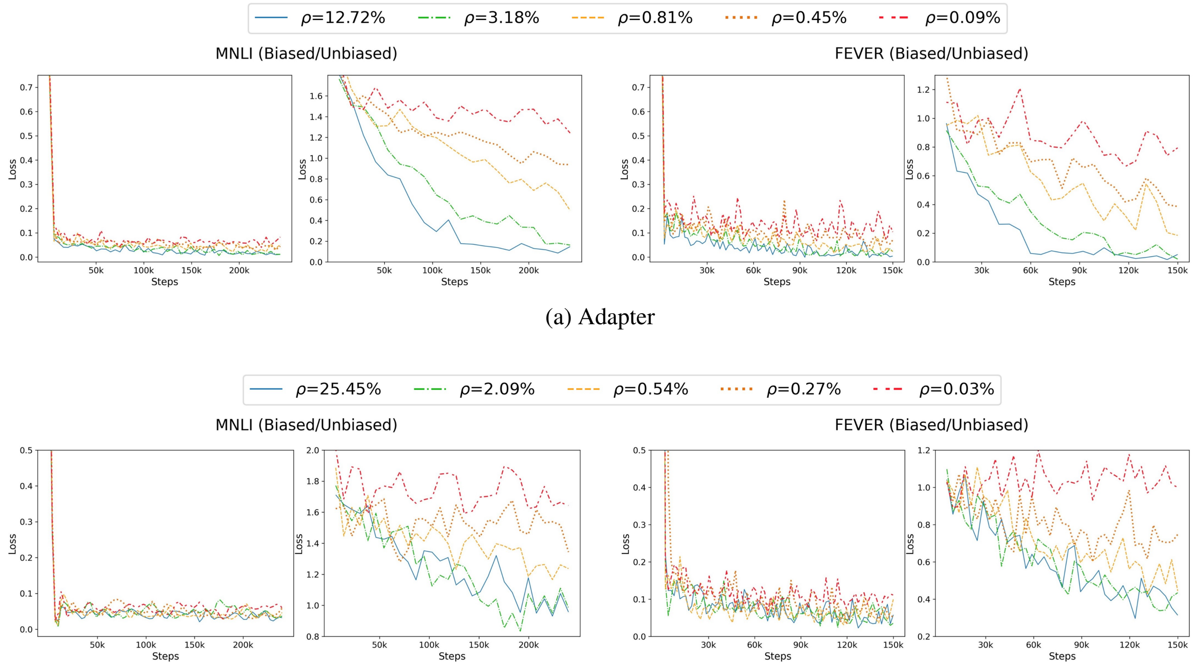
2 PEFT in Biased Scenarios

In this section, we study two research questions regarding the application of PEFT in biased training environments: (1) Does PEFT degrade the model's generalization performance? and (2) Why does this degradation occur? This pilot study provides insights for designing and understanding our framework.

2.1 Overall Setup

We investigate the behavior of Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2022), which are representative PEFT methods with sequential and parallel insertion forms, respectively. We leverage $\text{BERT}_{\text{Base}}$ (Devlin et al., 2019) as the base PLM. Following previous works (Utama et al., 2020; Sanh et al., 2021; Jeon et al., 2023), we consider MNLI and FEVER as biased training datasets.

[2] The intended task features refer to the essential attributes or patterns a model should learn to perform a given task.



(a) Adapter

(b) LoRA

Figure 3: Training curves for Adapter and LoRA. Here, ρ indicates trainable parameters per task. Adapter and LoRA easily converge on biased data regardless of ρ, whereas they show slower convergence on unbiased data as ρ decreases. In other words, the stronger the regularization effect, the more difficult it becomes for the PEFT methods to converge on unbiased data.

2.2 Does PEFT Degrade the Model's Generalization Performance?

Recent studies (Ding et al., 2022; Fu et al., 2023) have shown that PEFT enhances the generalization of models by regulating the number of updated parameters. Therefore, we observe ID and OOD performance by varying the number of parameters in PEFT methods. The OOD performance for MNLI and FEVER is evaluated on HANS (McCoy et al., 2019), and FEVER-Symmetric (Schuster et al., 2019), respectively. We use accuracy as the performance metric.

First, as shown in Figure 2, we observe that the ID performance of the model is maintained to some extent even as the number of trainable parameters decreases. On the other hand, the OOD performance drops to nearly random prediction levels as the number of parameters decreases. These experimental results suggest that, in biased training environments, PEFT may severely impairs the generalization of the model.

2.3 Why Does This Degradation Occur?

According to previous works (Utama et al., 2020; Sanh et al., 2021; Jeon et al., 2023), in the biased training environment, the generalization ability of the model depends on how well they learn unbiased data. Therefore, we analyze PEFT's training curve on unbiased data. To do this, we use datasets whose bias features are already known. For MNLI, we identified bias and unbiased examples using *lexical-overlap*, a well-known bias feature of the entailment class (McCoy et al., 2019). For FEVER, we leverage *LMI-ranked bigrams*, the bias feature of the refuge class, to identify biased and unbiased examples (Schuster et al., 2019). More details are provided in Appendix B.

Based on previous study (Du et al., 2023) and our observation (See Appendix D), fine-tuning converges first to the biased examples and then quickly converges to the unbiased examples. However, in the case of PEFT methods, we find that the model initially converges on biased examples similar to fine-tuning, but struggles to converge on unbiased examples (See Figure 3). As a result, full fine-tuning shows improvement in generalization performance after converging on biased examples (See Figure 7), whereas, in PEFT, generalization performance exhibits negligible improvement even after converging on biased data (See Figure 8). These findings suggest that the nature of low-rank adaptation in PEFT constrains the effective learning of

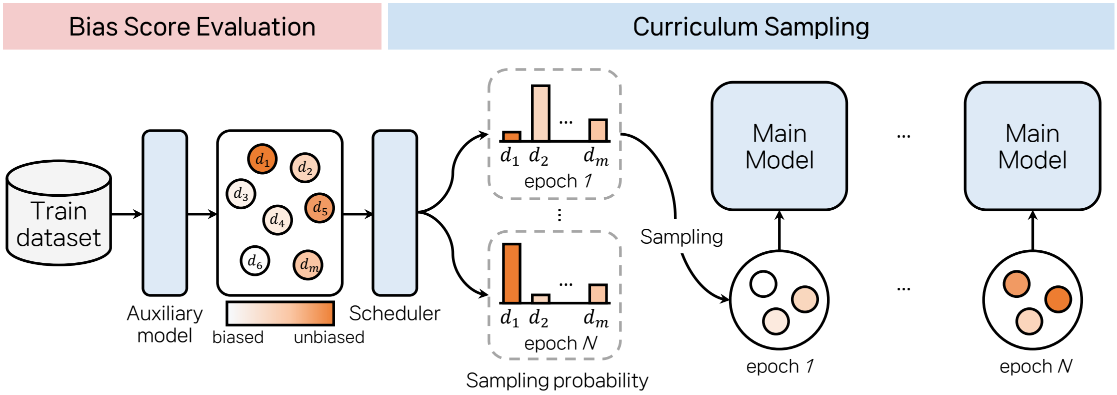


Figure 4: An overview of our proposed framework is provided. The framework comprises two main components: *bias score evaluation* and *curriculum sampling*. First, an auxiliary model assesses the bias scores of training examples, which are then utilized to compute sampling probabilities. Based on these probabilities, the training set for the main model is dynamically restructured by sampling examples at each stage.

unbiased examples, thereby slowing their convergence and ultimately limiting generalization. Accordingly, in this work, we aim to facilitate PEFT's learning on unbiased examples to accelerate convergence and improve generalization.

3 Methodology

To enhance the generalization capability of PEFT, we propose a novel curriculum learning framework, termed CURRICULUM DEBIASING (see Figure 4). Our proposed framework organizes training examples in a *biased-to-unbiased* order. Specifically, it estimates the bias score of each example using an auxiliary model's predictions (Section 3.1). Subsequently, based on these estimated scores, the training set is reconstructed at each training stage (Section 3.2).

3.1 Bias Score Evaluation

Curriculum learning improves generalization by presenting training examples in an easy-to-difficult order (Bengio et al., 2009). Inspired by this, we propose a novel curriculum that shifts training from biased to unbiased examples. Since unbiased examples are typically more difficult to learn (Utama et al., 2020; Sanh et al., 2021), we first train the model on biased data to facilitate early learning. Prior work suggests that when learning from biased examples, models do not solely capture spurious correlations but also learn intended task features (Kirichenko et al., 2023). Therefore, by later transitioning to unbiased ones, the model can reduce its reliance on spurious correlations while leveraging intended task features from biased ones, ultimately leading to better generalization.

To implement this, we first need to distinguish between biased and unbiased examples. However, due to the vast scope of NLP tasks, manually identifying bias in each example is impractical. Instead, we train an auxiliary model f_a to exploit dataset biases, following previous works (Sanh et al., 2021; Kim et al., 2022; Jeon et al., 2023), with details provided in the Appendix J. Then, we utilize f_a's predictions to estimate the bias score for each example. Specifically, for a given training example $(x^i, y^i) \in \mathcal{D}$, f_a produces a probability distribution p_a^i. We use $p_a^{i,c}$, which is the probability assigned to the correct label y^i, as a proxy for the bias score. Since f_a primarily exploits dataset biases, a high confidence score p_a^i indicates a biased example that is likely predicted based on spurious correlations.

3.2 Curriculum Sampling

To present examples in meaningful order, most curriculum learning approaches adopt the so-called *baby step* strategy (Spitkovsky et al., 2010). It splits the entire training set $\mathcal{D}$ into multiple buckets and starts training from the easiest bucket, gradually adding more challenging buckets (Cirik et al., 2016; Zhou et al., 2020; Lee et al., 2022). While this helps organize examples by difficulty, it can also cause overfitting or decrease training efficiency by repeatedly exposing the model to overly easy examples (Xu et al., 2020).

As an alternative, we propose a new training strategy called *curriculum sampling*. Instead of incrementally merging buckets, *curriculum sampling* reconstructs the training set at each stage by sampling examples based on their bias scores $p_a^{i,c}$. Formally, the sampling probability $\mathcal{P}_i$ of the exam-

Model	% FT Params	MNLI		FEVER		QQP	
		ID	OOD	ID	OOD	ID	OOD
BERT$_{\text{Base}}$ (Devlin et al., 2019)	100.00%	84.6	62.5	85.6	63.2	91.0	33.4
Adapter (Houlsby et al., 2019)	0.81%	82.6	52.5	86.5	56.8	89.4	34.5
Prompt-tuning (Lester et al., 2021)	0.09%	78.6	51.4	79.0	50.7	86.6	26.7
Prefix-tuning (Li and Liang, 2021)	1.06%	80.6	50.6	85.2	55.0	87.7	25.7
BitFit (Zaken et al., 2022)	0.09%	79.4	50.2	81.9	51.1	86.4	30.6
LoRA (Hu et al., 2022)	0.53%	83.6	51.0	85.8	56.4	90.0	34.1
AdaLoRA (Zhang et al., 2023)	0.40%	83.6	53.2	84.3	55.3	88.0	34.3
CURRICULUM DEBIASING$_{\text{Adapter}}$	0.81%	84.1	**64.1**	87.2	**63.8**	90.5	39.7
CURRICULUM DEBIASING$_{\text{LoRA}}$	0.53%	**84.3**	58.2	**88.6**	63.7	**91.1**	**41.5**

Table 1: Performance results of the BERT$_{\text{Base}}$ model on MNLI, FEVER, and QQP, along with their corresponding challenge test sets for out-of-distribution (OOD) evaluation. The best results are highlighted in **bold**, with the second-best results underlined.

Model	% FT Params	MNLI		FEVER		QQP	
		ID	OOD	ID	OOD	ID	OOD
Llama-3.2-1B (MetaAI, 2024)	100.00%	87.8	67.8	88.1	64.5	89.3	37.0
Adapter (Houlsby et al., 2019)	0.09%	86.2	60.1	88.4	62.7	89.0	43.8
LoRA (Hu et al., 2022)	0.03%	86.0	61.3	**89.9**	65.2	89.1	44.6
AdaLoRA (Zhang et al., 2023)	0.10%	86.1	60.7	89.8	63.2	**89.5**	44.8
CURRICULUM DEBIASING$_{\text{Adapter}}$	0.09%	86.1	64.3	89.2	66.1	89.2	49.0
CURRICULUM DEBIASING$_{\text{LoRA}}$	0.03%	**86.3**	**65.1**	**89.9**	**67.3**	**89.5**	**49.9**

Table 2: Performance results of the Llama-3.2-1B model. The best results are highlighted in **bold**.

ple (x^i, y^i) is calculated as:

$$\mathcal{P}_i = \alpha \times (p_a^{i,c})^{\mathcal{S}} \tag{1}$$

where α is a hyperparameter that determines the granularity of the curriculum and $\mathcal{S}$ is a scheduler that changes over training stages to gradually increase the proportion of less biased examples in the training set. To this end, we let $\mathcal{S}$ decrease linearly over the course of training. At training stage (or epoch) t, $\mathcal{S}$ is calculated as:

$$\mathcal{S} = \beta - \frac{2t}{T} \tag{2}$$

where T is the total number of training stages, and β is a hyperparameter specifying the initial sampling distribution. When $\mathcal{S}$ is large, biased examples (large $p_a^{i,c}$) have a higher probability of being sampled. As $\mathcal{S}$ decreases, less biased examples (small $p_a^{i,c}$) are sampled more frequently, thereby shifting the training focus to unbiased data.

By probabilistically reconstructing the training set at each epoch based on $p_a^{i,c}$ and dynamically adjusting $\mathcal{S}$ over time, *curriculum sampling* mitigates overfitting to easy examples while still leveraging the benefits of curriculum learning. In summary, *curriculum sampling* enables a smooth transition from "biased" to "unbiased" examples, helping the model converge faster and generalize better.

4 Experiment

4.1 Evaluation Datasets

Following prior works (Utama et al., 2020; Sanh et al., 2021; Jeon et al., 2023), we evaluate our models on three tasks: natural language inference (MNLI), fact verification (FEVER), and paraphrase identification (QQP). Further details are provided in the Appendix M.

4.2 Baselines

We consider 6 PEFT methods that are widely adopted in NLP applications as baselines. *Adapter* (Houlsby et al., 2019) represents a method that adds small bottleneck layers to the network. *Prompt-tuning* (Lester et al., 2021) optimizes task-specific prompts that are inserted into the input embeddings to guide the model. *Prefix-tuning* (Li and Liang, 2021) is a method that prepends learnable vectors

Method	MNLI		FEVER		QQP	
	ID	OOD	ID	OOD	ID	OOD
CURRICULUM DEBIASING$_{\text{Adapter}}$	84.1	64.1	87.2	63.8	90.5	39.7
w/o Curriculum Scheduler	83.4	58.0	86.7	62.4	87.9	35.6
w/o Difficulty-based Sampling	82.7	50.7	84.2	59.1	89.2	30.3
w/o Curriculum Scheduler & Difficulty-based Sampling	79.8	50.1	83.2	59.8	88.9	30.1

Table 3: Ablation study of CURRICULUM DEBIASING. w/o Difficulty-based Sampling and w/o Scheduler indicate the model without the corresponding component.

to the input to guide the model's attention. *BitFit* (Zaken et al., 2022) adjusts only the bias terms of the model parameters. *LoRA* (Hu et al., 2022) is a technique that decomposes weight updates into low-rank matrices. *AdaLoRA* (Zhang et al., 2023) dynamically adjusts the rank of the low-rank matrices during training.

To demonstrate the effectiveness of CURRICULUM DEBIASING, we applied it to both Adapter and LoRA, which are the most representative modular insertion approaches in PEFT. Additional experimental results with another representative method, prompt-tuning, are provided in Appendix G.

4.3 Setup

We use BERT$_{\text{Base}}$ (Devlin et al., 2019) with 110M parameters and Llama-3.2 with 1B parameters for the base PLM. We use BERT$_{\text{Tiny}}$ (Turc et al., 2019) with 4M parameters for the auxiliary model in the experiments. Additional implementation and hyperparameter details are provided in Appendix K.

4.4 Results

In Table 1, PEFT models, regardless of their type, show significant performance degradation compared to full fine-tuning on OOD evaluation sets, with most performing similarly to random guessing. On the other hand, our models show significant performance improvements compared to these baselines across all evaluation sets. Specifically, the Adapter tuned with CURRICULUM DEBIASING achieves a better average OOD performance of 55.9, compared to 47.9 achieved by the vanilla model. Additionally, regarding average ID performance, using CURRICULUM DEBIASING results in a 1.2-point improvement. These results demonstrate the effectiveness of the proposed method. These results illustrate that our method can boost OOD performance in PEFT without sacrificing ID performance.

Furthermore, as shown in Table 2, with Llama-3.2 models, CURRICULUM DEBIASING also shows significant OOD performance improvements. Specifically, it achieves 3.7 points higher OOD performance and 0.2 points higher ID performance than baseline models. These results indicate that the proposed method is also effective for decoder-based models.

5 Analysis

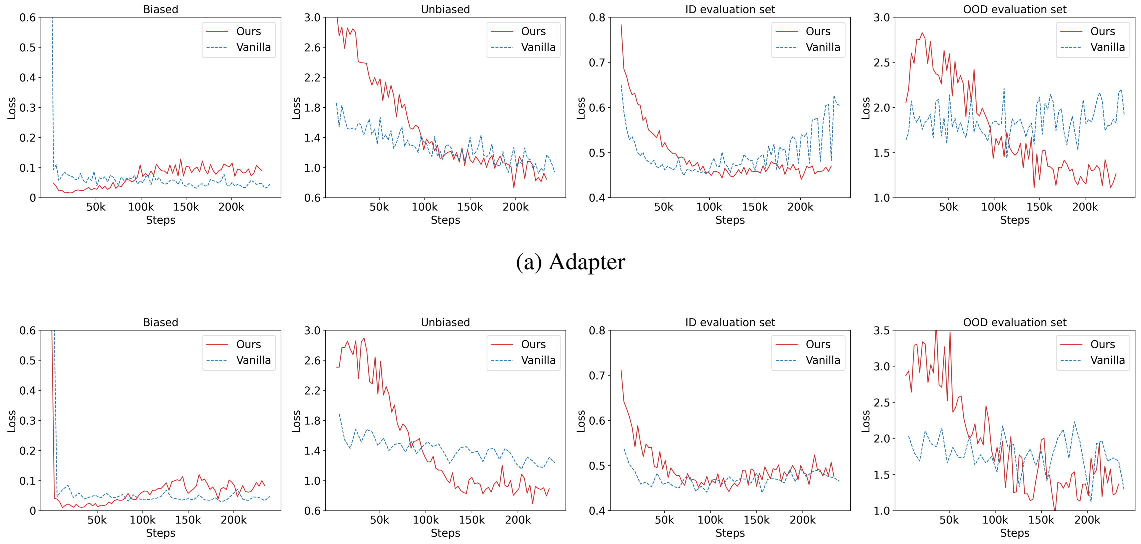
5.1 Ablation Study

We experimented with various ablation settings in CURRICULUM DEBIASING to investigate the effect of each component. (1) w/o Curriculum Scheduler adopts a curriculum that divides the training dataset into five buckets according to difficulty levels, following (Zhang et al., 2018), and (2) w/o Difficulty-based Sampling accumulates difficult examples to the initial training set as the curriculum progresses rather than re-construct training set by sampling.

Table 3 indicates that all the components are important in the model's generalization improvement. With the coarse-grained curriculum (w/o Curriculum Scheduler), the model shows a significant performance drop in OOD evaluation sets. It indicates that frequently changing the curriculum stage helps to improve the model's generalization. In addition, re-constructing the training set at each stage (w/o Difficulty-based Sampling) significantly influences both ID and OOD performances. These results indicate that example sampling is the key to our method and over-iterating easy examples substantially harms the model's generalization ability. As a supplementary analysis, the results of the reverse curriculum setting are provided in the Appendix H.

5.2 Convergence Speed

We introduce CURRICULUM DEBIASING to accelerate the convergence speed of the model on



(a) Adapter

(b) LoRA

Figure 5: Training curves for the CURRICULUM DEBIASING and vanilla PEFT methods on MNLI. Here, the biased and unbiased examples were the examples used in the pilot study in Section 2.

unbiased data. To verify the effectiveness of our framework in terms of convergence, we monitor the model's convergence during training.

In Figure 5, we observe that CURRICULUM DEBIASING consistently accelerates the convergence speed of PEFT. In particular, it significantly improves the convergence speed of LoRA on unbiased data, which has the slowest convergence speed. These results show the effectiveness of our framework in improving the convergence speed of PEFT on unbiased data. Additionally, unlike the vanilla Adapter, which shows increased losses on development data after 100k steps, the losses with CURRICULUM DEBIASING consistently decrease. This suggests that the vanilla Adapter overfits the training data, while CURRICULUM DEBIASING mitigates overfitting by restructuring the training dataset.

5.3 Comparison with Debiasing Methods

In Table 4, existing debiasing methods show a slight performance improvement on the OOD evaluation set with the expense of a significant performance drop on the ID evaluation set. These results suggest that the existing methods tend to overfit the model to unbiased examples rather than improving the model's generalization capability. On the other hand, our model shows significant performance improvement on both ID and OOD evaluation sets. This suggests that gradually introducing unbiased examples after initially learning the task's representative features through biased examples, rather than emphasizing unbiased ones from the start, helps improve the model's generalization. Settings for this experiment are described in Appendix N.

6 Related Work

6.1 Parameter-efficient Fine-tuning

Parameter-efficient fine-tuning (PEFT) is a technique to adapt pre-trained neural models to new tasks with minimal trainable parameters. For instance, the Adapter (Houlsby et al., 2019) and its variants (Pfeiffer et al., 2021; Zhang et al., 2024) introduce small, task-specific modules into the model. Prompt-tuning (Lester et al., 2021) and Prefix-tuning (Li and Liang, 2021) adjust the input prompts or prefixes fed into the model, enabling task-specific adjustments without altering the main model parameters. BitFit (Zaken et al., 2022) updates only the bias terms of a pre-trained model, and LoRA (Hu et al., 2022) and its variants (Zhang et al., 2023; Dettmers et al., 2023) decompose weight into low-rank matrices and fine-tune only these low-rank components. Recent studies (Hu et al., 2022; Fu et al., 2023; Xie and Lukasiewicz, 2023) have shown that these methods not only improve computational efficiency but also enhance models' generalization capability by providing a regularization effect to training. However, in biased datasets, we have found that the regularization effect impedes the model's convergence on unbiased data, thus significantly impairing its generalization

Method	% FT Params	MNLI		FEVER		QQP	
		ID	OOD	ID	OOD	ID	OOD
Full Fine-tuning	100.00%	84.6	62.5	85.6	63.2	91.0	33.4
Adapter (Houlsby et al., 2019)	0.81%	82.6	52.5	86.5	56.8	89.4	34.5
Focal Loss (Lin et al., 2017)	0.81%	80.3	55.0	82.1	55.0	81.4	36.2
Reweighting (Schuster et al., 2019)	0.81%	77.5	60.0	78.7	61.0	81.8	**50.9**
PoE (Clark et al., 2019)	0.81%	79.8	62.7	73.6	56.2	82.6	44.1
Reweighting+Anneal. (Utama et al., 2020)	0.81%	78.3	56.5	80.2	58.1	79.5	49.8
PoE+CE (Sanh et al., 2021)	0.81%	80.0	54.8	78.3	56.5	86.3	39.8
PoE+CE+Bias Experts (Jeon et al., 2023)	0.81%	80.7	58.5	81.4	59.8	85.1	45.4
CURRICULUM DEBIASING$_{\text{Adapter}}$	0.81%	**84.1**	**64.1**	**87.2**	**63.8**	**90.5**	39.7

Table 4: Comparison of debiasing methods. Sanh et al. (2021) utilizes a multi-loss objective to mitigate the decline in ID performance, while Utama et al. (2020) employs an annealing mechanism. For a fair comparison, we used our auxiliary model for all debiasing baselines. The best results are highlighted in **bold**.

ability. This is a critical issue to real-world applicability, considering that NLP training data are often inherently biased (Sun et al., 2019; Patel et al., 2021; Branco et al., 2021).

6.2 Debiasing NLU Models

Several studies have shown that NLU models often exploit biases in datasets for inference. For instance, in the natural language inference (NLI) task, models can predict correct answers with only partial inputs (Gururangan et al., 2018; Poliak et al., 2018) or by exploiting lexical overlap biases (McCoy et al., 2019; Dasgupta et al., 2018). Similar phenomena are also observed in other NLP tasks (Schuster et al., 2019; Zhang et al., 2019; Yang et al., 2018; Welbl et al., 2018). Such bias exploitation hinders the model from learning underlying tasks, leading to incorrect predictions on out-of-distribution (OOD) or adversarial data.

To address the issue, several debiasing methods have been proposed, which can be grouped into two categories: unbiased dataset construction and adversarial training. Methods for constructing unbiased datasets involve designing elaborate protocols to avoid acquiring biased data (Reddy et al., 2019; Choi et al., 2018), eliminating biased data using adversarial filtering (Zellers et al., 2018; Sakaguchi et al., 2021; Bras et al., 2020), or augmenting datasets with adversarial data (Jia and Liang, 2017; Zmigrod et al., 2019). On the other hand, adversarial training algorithms aim to make models more robust to dataset biases. They typically emphasize losses of unbiased examples in the main model's training objective. To identify unbiased examples from datasets, initial attempts utilize a biased auxiliary model heuristically designed to exploit biases in the datasets (Schuster et al., 2019; Clark et al., 2019; Mahabadi et al., 2020). However, acquiring human prior knowledge about the biases for numerous datasets requires huge costs. Thus, recent studies have attempted to train the biased auxiliary model without human supervision (Bras et al., 2020; Ghaddar et al., 2021; Liu et al., 2021; Kim et al., 2022; Jeon et al., 2023). Despite their promising results, we observe that existing debiasing methods hinder the convergence of PEFT by emphasizing unbiased examples from the beginning, ultimately harming the ID performance.

7 Conclusion

We have demonstrated that, in biased training environments, PEFT slows down convergence on unbiased examples, which in turn increases the model's reliance on spurious correlations in biased datasets. To mitigate this issue, we introduced CURRICULUM DEBIASING, a simple yet effective strategy that accelerates PEFT convergence on unbiased data, leading to better generalization. Our experiments, conducted across multiple NLP benchmarks, confirm the effectiveness of our approach, showing that the proposed framework significantly improves OOD performance without sacrificing ID accuracy—a limitation of many existing debiasing methods. These findings underscore the potential of CURRICULUM DEBIASING as a promising solution for enhancing the robustness and generalization of PEFT methods in real-world applications.

Limitations

Although CURRICULUM DEBIASING has significantly improved the generalizability of PEFT in biased training scenarios, several limitations remain, presenting valuable opportunities for future research.

First, our work has focused mainly on addressing biases in NLU tasks, aligning with previous studies (Sanh et al., 2021; Jeon et al., 2023). However, the investigation of bias in natural language generation (NLG) remains an open challenge. Given the potentially far-reaching impact of biased text generation, exploring bias mitigation in NLG tasks is a promising avenue for future research.

Second, we introduce hyperparameters in our approach, which can be a potential limitation in debiasing scenarios due to the lack of validation sets in most out-of-distribution situations (Utama et al., 2020). However, as shown in Tables 9 and 10, our method demonstrates greater robustness to hyperparameter variations compared to previous studies (Utama et al., 2020; Sanh et al., 2021; Kim et al., 2022; Jeon et al., 2023), mitigating many of the sensitivity issues commonly observed in debiasing methods. Nevertheless, exploring strategies for ensuring continued robustness to hyperparameter choices remains a key goal for future work.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2025-00517221 and No.RS-2024-00415812) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2024-00439328, Karma: Towards Knowledge Augmentation for Complex Reasoning (SW Starlab), No.RS-2024-00457882, AI Research Hub Project, and No.RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University)).

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 7319–7328. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1078–1088. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184. Association for Computational Linguistics.

Volkan Cirik, Eduard H. Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. In *CoRR, abs/1611.06204*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4069–4082. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *CoRR, abs/1802.04302*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 1233–1249. Curran Associates, Inc.

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

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Yanrui Du, Jing Yan, Yan Chen, Jing Liu, Sendong Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Bing Qin. 2023. Less learn shortcut: Analyzing and mitigating learning of spurious feature-label correlation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5039–5048. ijcai.org.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 12799–12807. AAAI Press.

Abbas Ghaddar, Philippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 1923–1929. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 107–112. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *Proceedings of the 11th International Conference on Learning Representations*. OpenReview.net.

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, pages 2790–2799. PMLR.

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 *Proceedings of the 10th International Conference on Learning Representations*. OpenReview.net.

Eojin Jeon, Mingyu Lee, Juhyeong Park, Yeachan Kim, Wing-Lam Mok, and SangKeun Lee. 2023. Improving bias mitigation through bias experts in natural language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11053–11066. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031. Association for Computational Linguistics.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages = 9012–9020, publisher = IEEE Computer Society.*

Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. 2022. Learning debiased classifier with biased committee. In *Advances in Neural Information Processing Systems*, volume 35, pages 18403–18415. Curran Associates, Inc.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2023. Last layer re-training is sufficient for robustness to spurious correlations. In *Proceedings of the 11th International Conference on Learning Representations*. OpenReview.net.

Mingyu Lee, Jun-Hyung Park, Junho Kim, Kang-Min Kim, and SangKeun Lee. 2022. Efficient pre-training of masked language model via concept-based curriculum masking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7417–7427. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4582–4597. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pages 2999–3007. IEEE Computer Society.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 6781–6792. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*. OpenReview.net.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1022–1035. Curran Associates, Inc.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448. Association for Computational Linguistics.

MetaAI. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035. Curran Associates, Inc.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Communications of the ACM*, 64.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *Proceedings of the 9th International Conference on Learning Representations*. OpenReview.net.

Tal Schuster, Darsh J. Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3419–3425. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 751–759. Association for Computational Linguistics.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth M. Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1630–1640. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819. Association for Computational Linguistics.

Marcos V. 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 A. Milder, Colin Raffel, Edwin Simpson, Noam Slonim, Niranjan Balasubramanian, Leon Derczynski, and Roy Schwartz. 2023. Efficient methods for natural language processing: A survey. *Transactions of the*

Association for Computational Linguistics, 1:826––860.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Can Udomcharoenchaikit, Wuttikorn Ponwitayarat, Patomporn Payoungkhamdee, Kanruethai Masuk, Weerayut Buaphet, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. Mitigating spurious correlation in natural language understanding with counterfactual inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11308–11321. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7597–7610. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

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

Zhongbin Xie and Thomas Lukasiewicz. 2023. An empirical analysis of parameter-efficient methods for debiasing pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 15730–15745. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. 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*, pages 1–9. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *Proceedings of the 11th International Conference on Learning Representations*. OpenReview.net.

Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. 2024. Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *Proceedings of the 12th International Conference on Learning Representations*. OpenReview.net.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *CoRR*, abs/1811.00739.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1298–1308. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944. Association for Computational Linguistics.

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

Appendix

A Details Setups for Figure 1

The models in the experiment in Figure 1 were trained for 250k steps. In this experiment, we save a checkpoint every 5k steps and report results based on the model checkpoint corresponding to the highest ID performance. Other settings are the same as in the main experiment.

B Details of Pilot Study

Here, we present additional details of the pilot study discussed in Section 2.

MNLI. We first select examples that all words in the hypothesis occur in the premise. Then, among the selected cases, we classify those where the correct answer is entailment as biased, and the others as unbiased. Consequently, the number of biased and unbiased examples is 1,807 and 295, respectively.

FEVER. We select examples where the claim includes more than one of the top 10 LMI-ranked bigrams for *REFUTES* class listed by (Schuster et al., 2019). Then, we classify those where the correct answer is *REFUTES* as biased, and the others as unbiased. In this process, we exclude cases from the unbiased examples where any of the top 10 LMI-ranked bigrams from the *SUPPORT* or *NOT ENOUGH INFO* class appear in the claim. Consequently, the number of biased and unbiased examples is 3,959 and 1,476, respectively.

C Trade-off

In this work, we save a checkpoint at every epoch and report the OOD performance for the model checkpoint corresponding to the highest ID performance. However, according to previous studies (Utama et al., 2020; Sanh et al., 2021; Jeon et al., 2023), there is a trade-off between ID and OOD performance in biased training environments. This indicates that the OOD performance of our model and the Vanilla PEFT methods may have been underestimated. Therefore, we measured and reported the ID and OOD performance every 3k steps.

As shown in Figure 6, the OOD performance of the Vanilla Adapter gradually improves after 100k steps, where it achieves the highest performance on the ID evaluation set. In other words, there is a trade-off between ID and OOD performance, and achieving better OOD performance requires sacrificing ID performance. Additionally, the Vanilla Adapter exhibits a high variance in OOD performance across steps, making model selection difficult. In contrast, with CURRICULUM DEBIASING, both ID and OOD performance improve simultaneously as training progresses. This indicates that the model effectively learns the underlying task before overfitting to the training data. Furthermore, our model exhibits less variance in OOD performance, making it practical for deployment in various applications.

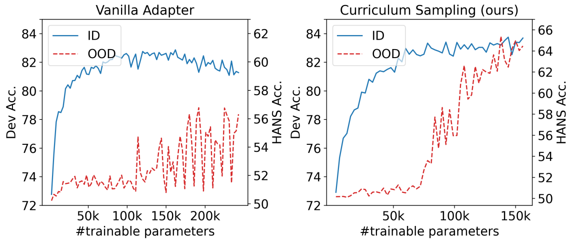
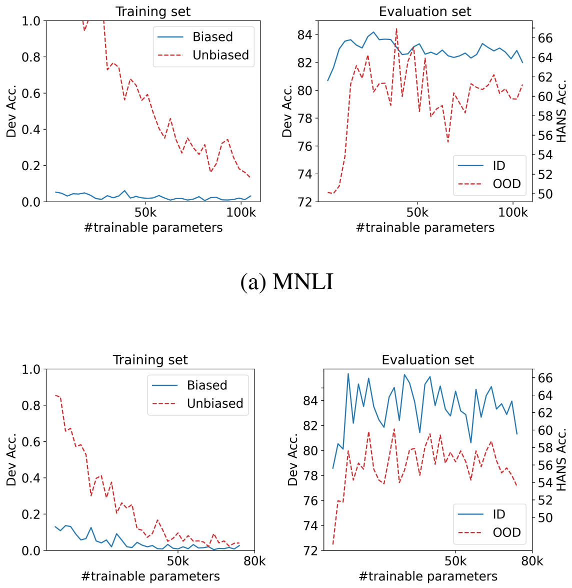


Figure 6: Trade-off curve between ID and OOD performance on MNLI dataset.

D Training Curves for Full Fine-tuning



(a) MNLI

(b) FEVER

Figure 7: The training curves for full fine-tuning. We trained BERT$_{\text{Base}}$ for 10 epochs on both datasets.

As shown in Figure 7, the model first converges on the biased data and then on the unbiased data. However, after a certain number of steps, the accuracy on the OOD evaluation sets does not increase, unlike the training curve for PEFT in Figure 6. In some cases, it even decreases. These results indicate that overfitting occurs as the model memorizes the training data during over-iterations. Therefore, we need to prevent the model from memorizing the training data while learning the underlying task. CURRICULUM DEBIASING, which reconstructs the training dataset at each epoch, aligns well with this objective.

E Additional Training Curves

As shown in Figure 8, our model shows better convergence on training data than the Vanilla Adapter with only half of the training steps. Consequently, our model performs better than the baseline on both ID and OOD evaluation sets, indicating that our method not only improves the generalization of the model but also improves the training efficiency of PEFT.

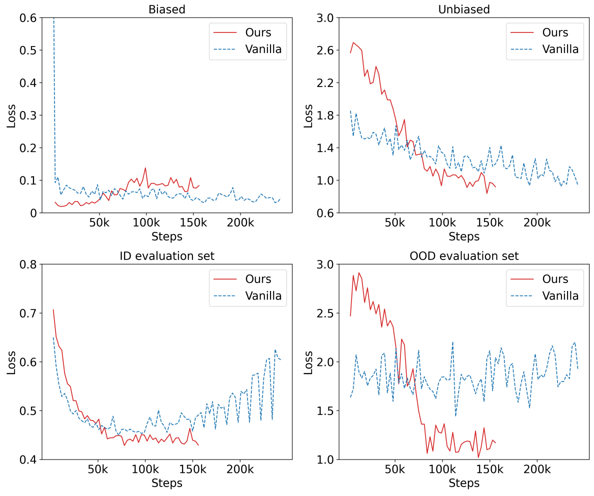


Figure 8: The training curves for our Adapter, trained for 30 epochs on MNLI.

F GLUE Results

We evaluated our model on the GLUE benchmark, excluding the regression task STS-B due to its distinct evaluation metric. The results, summarized in Table 5, show that our proposed method consistently outperforms the Adapter baseline across all tasks in the GLUE benchmark. These results highlight the effectiveness of our method in improving generalization across diverse NLP tasks.

Task	Metric	Adapter	Ours	Δ
MNLI	Accuracy	82.6	84.2	+1.6
RTE	Accuracy	70.7	72.2	+1.5
QQP	Accuracy	89.4	90.8	+1.4
CoLA	MCC	59.0	59.7	+0.7
SST	Accuracy	92.5	93.0	+0.5
MRPC	Accuracy	84.8	85.2	+0.4
QNLI	Accuracy	90.9	91.0	+0.1

Table 5: Evaluation results on the development sets of GLUE.

G Results on Prompt-tuning

We apply CURRICULUM DEBIASING to prompt-tuning and evaluate its impact on performance. As shown in Table 6, incorporating CURRICULUM DEBIASING improves both ID and OOD performance compared to standard prompt-tuning. These results demonstrate that our proposed method generalizes well to prompt-based tuning approaches, further validating its effectiveness.

Nonetheless, we observe that in BERT, applying curriculum debiasing to prompt-tuning does not lead to significant performance improvements. Since smaller models are known to require a higher intrinsic dimension (Aghajanyan et al., 2021), we suspect that models like BERT may need more expressive PEFT methods such as Adapter and LoRA to effectively capture the complexity of unbiased examples.

Method	MNLI		FEVER	
	ID	OOD	ID	OOD
Prompt-tuning	84.9	52.3	87.1	53.9
CURRICULUM DEBIASING	**85.2**	**58.61**	**89.9**	**57.0**

Table 6: Comparison with original prompt-tuning. This experiment was conducted on LLaMA-3.2-1B.

H Effect of Curriculum Design

To examine the impact of the order of presenting examples in curriculum design, we compare two training strategies: Biased-to-Unbiased (ours) and Unbiased-to-Biased (reverse curriculum). As shown in Table 7, the model trained with Biased-to-Unbiased achieves an average OOD performance of 66.2, compared to 59.6 achieved by the Unbiased-to-Biased strategy. Similarly, in terms of ID performance, the Biased-to-Unbiased model outperforms the Unbiased-to-Biased model by 0.6 points on average. These results highlight that starting with biased data and gradually introducing unbiased examples is essential for effective generalization.

Method	MNLI		FEVER	
	ID	OOD	ID	OOD
Unbiased-to-Biased	85.7	59.8	83.1	59.4
No Curriculum	86.0	61.3	**89.9**	65.2
Biased-to-Unbiased	**86.3**	**65.1**	**89.9**	**67.3**

Table 7: Comparison of different curriculum designs. The experiment was conducted using LoRA on LLaMA-3.2-1B.

I Effectiveness on Larger Models

Method	MNLI		FEVER	
	ID	OOD	ID	OOD
Random Sampling	86.2	63.5	87.5	63.8
CURRICULUM DEBIASING	**87.1**	**69.2**	**89.2**	**66.5**

Table 8: Performance comparison of CURRICULUM DEBIASING and random sampling using $\text{BERT}_{\text{Large}}$.

To validate the scalability of our proposed method, we conducted additional experiments using $\text{BERT}_{\text{Large}}$. As shown in Table 8, CURRICULUM DEBIASING consistently outperforms random sampling on both ID and OOD evaluation sets for MNLI and FEVER. Specifically, on MNLI, CURRICULUM DEBIASING achieved a 5.7 points improvement in OOD performance while also improving ID performance by 0.9 points. A similar trend is observed on FEVER, where CURRICULUM DEBIASING improves OOD performance by 2.7 points and ID performance by 1.7 points. These results demonstrate both the scalability and effectiveness of CURRICULUM DEBIASING when applied to larger models.

J Details of the Auxiliary Model

Deep neural networks often exploit spurious correlations in datasets, leading to high in-distribution (ID) performance but poor generalization to out-of-distribution (OOD) data. To mitigate this issue, previous debiasing methods employ an *auxiliary model* that identifies biased examples, allowing the main model to down-weight them during training (Kim et al., 2019; Schuster et al., 2019; Clark et al., 2019; Mahabadi et al., 2020). Traditional approaches train the auxiliary model using explicit bias labels, but acquiring such annotations is costly and impractical for large-scale datasets.

Recent studies have introduced bias-inducing strategies to train the auxiliary model without explicit bias labels. These methods constrain the training environment, such as limiting model capacity (Sanh et al., 2021; Jeon et al., 2023), training with fewer epochs (Liu et al., 2021), or restricting accessible data (Utama et al., 2020; Kim et al., 2022). Such constraints encourage the model to rely on superficial correlations when making predictions. Consequently, we classify examples where the auxiliary model confidently predicts the correct answer as biased examples. In this work, we follow the strategy of reducing model capacity, adopting $\text{BERT}_{\text{Tiny}}$ as the backbone for the auxiliary model.

K Additional Experimental Setups

For $\text{BERT}_{\text{Base}}$, we set the bottleneck dimension of the Adapter to 48, the prompt length for prompt-tuning to 64, the prefix length for prefix-tuning to 64, and the rank of LoRA to 16. For Llama-3.2, we set the bottleneck dimension of the Adapter to 8 and the rank of LoRA to 4. We use AdamW (Loshchilov and Hutter, 2019) as the optimizer, with the learning rate searched within the range of 2×10^{-5} to 5×10^{-4}. The batch size is set to 32 for BERT and 8 for Llama-3.2. All baselines using BERT are trained for 20 epochs, while those using Llama-3.2 are trained for 5 epochs.

Notably, CURRICULUM DEBIASING utilizes only a subset of the training set in each epoch, resulting in fewer total training steps than the baselines. To ensure a fair comparison by maintaining a similar number of training steps, we train our Adapter for 30 epochs on MNLI and 45 epochs on FEVER and QQP; LoRA is trained for 45 epochs across all three datasets. For Llama-3.2, we train our Adapter for 5 epochs on all three tasks.

We save a checkpoint at every epoch and report results based on the model checkpoint that achieves the highest in-distribution (ID) performance. We report the median scores from three independent random runs. All experiments are conducted on RTX 3090 and RTX 2080 GPUs. We implement our models and baselines using PyTorch (Paszke et al., 2019) and the Hugging Face libraries[3].

[3] https://github.com/huggingface/

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
ID	81.2	83.4	83.8	83.7	84.1	84.0	84.3	83.8	83.4	83.4
OOD	69.4	66.0	67.6	65.8	64.1	63.2	62.3	56.2	51.6	51.2

Table 9: Analysis results of the hyperparameter α. Here, we set β to 1.0.

β	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
ID	83.8	84.1	84.1	84.1	83.7	83.7	83.4	83.9	83.7	83.9	83.7
OOD	63.4	64.3	64.3	64.8	64.2	65.8	65.2	63.1	57.2	50.1	52.9

Table 10: Analysis results of the hyperparameter β. Here, we set α to 0.4.

L Hyperparameter Analysis

We empirically chose our hyperparameters for CURRICULUM DEBIASING. We analyzed different values while keeping them constant, and vice versa. The results are summarized in Tables 9 and 10. Firstly, we observed that smaller α values yield better OOD performance. However, when α becomes extremely small, ID performance drops significantly. This is likely because overly fine-grained curriculum steps result in insufficient learning of easy examples during the final training stage. Meanwhile, β shows the highest ID and OOD performance around 1, suggesting that learning with a subset composed mainly of easy examples for about half of the total training period is effective in improving generalization performance.

M Evaluation Datasets

We use accuracy as the performance metric for each task.

Natural Language Inference. MNLI (or MultiNLI) (Williams et al., 2018) is a task that determines the relationship between a pair of sentences (premise and hypothesis) as either contradiction, entailment, or neutral. We train models using the MNLI training set. Subsequently, we evaluate the ID performance on the MNLI development set, and the OOD performance on HANS (McCoy et al., 2019), a challenging test dataset specifically designed to evaluate whether models exploit bias features, such as lexical overlap, for inference.

Fact Verification. The goal of FEVER (Thorne et al., 2018) is to determine if the evidence supports, refutes, or lacks sufficient information to evaluate a claim. We train the models using the FEVER training set. Then, we evaluate the models' ID performance on the FEVER development set and their OOD performance on FEVER Symmetric (Schuster et al., 2019), a challenging test dataset designed to check whether models depend on bias features in claims.

Paraphrase Identification. The objective of QQP[4] is to determine whether a pair of questions has the same meaning. Following previous works (Udomcharoenchaikit et al., 2022; Jeon et al., 2023), we divide this dataset into train and development sets so that the development set contains 5k examples. We then train the models on the training set and evaluate their ID performance on the QQP development set and OOD performance on PAWS (Zhang et al., 2019) to test whether the models exploit bias features, such as lexical overlap bias. Statistics of datasets are provided in Table 11.

Task	#train data	#evaluation data		#classes
		ID	OOD	
MNLI	392,702	9,815	30,000	3
FEVER	242,911	16,664	717	3
QQP	394,287	5,000	677	2

Table 11: Detailed statistics of datasets.

N Debiasing Baselines

We compare CURRICULUM DEBIASING with existing debiasing methods. Specifically, we compare our proposed method to the following debiasing methods: Focal Loss (Lin et al., 2017), example reweighting (Reweighting) (Schuster et al., 2019), product-of-experts (PoE) (Clark et al., 2019; Mahabadi et al., 2020), and bias experts (Jeon et al., 2023). They identify biased examples in the dataset and place greater emphasis on unbiased ones, in

[4] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

other words, the model is guided to focus on difficult examples from the start. Thus, a substantial decline in ID performance often occurs. To alleviate performance degradation, recent works have further adopted a multi-loss objective (Sanh et al., 2021; Jeon et al., 2023) or annealing mechanism (Utama et al., 2020). Thus, we also compare our model with these variants for a faithful comparison.

O Comparison of Scheduling Strategies

In addition to the linear schedule used in our main experiments, we evaluate two alternative scheduling strategies—exponential and cosine. As shown in Table 12, linear scheduling achieves the best overall performance, while cosine scheduling slightly outperforms exponential scheduling across most metrics. However, both alternatives are implemented as simple baselines, and given the vast space of possible variants, a thorough investigation of scheduling strategies remains an important direction for future work.

Method	MNLI		QQP	
	ID	OOD	ID	OOD
Exponential	83.4	55.2	90.2	38.7
Cosine	83.3	57.2	90.7	40.8
Linear (ours)	**84.3**	**58.2**	**91.1**	**41.5**

Table 12: Performance comparison across different scheduling strategies.

P Robustness to Auxiliary Model Variants

We conducted additional experiments using the auxiliary model with different capacity and architecture — specifically, BERT$_{\text{small}}$—to assess the robustness of our method to the choice of auxiliary model. The results are included in the table below and show that our method performs consistently across different auxiliary models, indicating that it does not overly rely on any specific architecture.

Method	MNLI		QQP	
	ID	OOD	ID	OOD
None	83.6	53.2	88.0	34.3
BERT$_{\text{small}}$	83.6	**60.6**	88.2	39.1
BERT$_{\text{tiny}}$ (ours)	**84.3**	58.2	**91.1**	**41.5**

Table 13: Performance comparison with different auxiliary model capacities.