BIASPAD: A BIAS-PROGRESSIVE AUTO-DEBIASING FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

While large pre-trained language models have made great strides in natural language understanding benchmarks, recent studies have found that models rely more on the superficial or short-cut features to make predictions. In this paper, we study how to progressively and automatically detect and filter the biased data to train a robust debiased model for NLU tasks. Rather than focusing on the human-predefined biases or biases captured by a bias-only model of limited-capacity, we introduce a new debiasing framework, called **Bias**-**P**rogressive **A**uto-**D**ebiasing (BIASPAD), based on two observations: i) the higher the proportion of bias in the training data, the more biased the model will be, and ii) a more biased model has higher confidence in predicting the bias. The framework progressively trains a bias-only model by using the most biased samples detected in the previous epoch, which ensures a more biased model and leads to a robust debiased model. The extensive experiments demonstrate the effectiveness of the proposed framework on several challenging NLU datasets, where on HANS, we achieve 5% accuracy improvement.

1 INTRODUCTION

In the last decade, deep representation learning has shown its general capability on a broad spectrum of tasks and made significant progress on natural language understanding datasets, e.g., GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019). However, recent studies (Poliak et al., 2018; Gururangan et al., 2018; McCoy et al., 2019) reveal that the models tend to capture *dataset biases* (i.e., the superficial clues such as word overlaps and negative words) to make predictions, rather than learning from the underlying features. Such an issue becomes the main barrier to the models' reliability in deployment, especially on out-of-distribution generalization. Moreover, the issue still remains for the recent large-scale pre-trained models with generic representations.

As such, reducing the impact of *dataset biases* becomes the key challenge to learn robust natural language understanding (NLU) models. Early works of debiasing methods rely heavily on human experts (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020) to manually analyze the potential biases in a specific artificial dataset and then define the most likely bias types in that dataset. Since these experience-dependent methods are usually time-consuming, recent studies focus on automatic and dataset-agnostic debiasing methods for NLU to cover more extensive bias types, including those are hardly induced by the experts. The fundamental idea behind those automatic debiasing methods is to firstly train a *bias-only model* and use it to implicitly or explicitly detect biased samples. Then these samples are downweighed in the training of a *debiased model*.

Therefore, the key problem reduces to train a *bias-only model*. In previous works, two heuristic assumptions are usually applied to train the bias-only model. The first is the '*weak-model*' assumption: models of lower capacity (e.g., Bag-of-words model or TinyBERT) are easier to learn from shallow heuristics of the datasets, which results in a bias-only model (Sanh et al., 2020). The second is the '*small-data*' assumption: a model is prone to fitting shortcut or biased features in the dataset in its early training stages (Utama et al., 2020a).

However, both assumptions cannot always hold and are full of uncertainty with many uncontrollable factors. Intuitively, it is difficult to define how weak the model is and how small the dataset is, leaving redundant hyperparameters. Moreover, the model for the bias-only purpose is inevitably fed with normal or robust samples due to both i) the unknown dataset-specific biasing sample proportion and

ii) the randomness of model-selecting or data-sampling. All these uncontrollable factors possibly lead to a less-biased bias-only model and cause negative effects on the debiased model learning.

Thereby, we aim at a stable learning method for a better biased model in an automatic manner, and the method should be agnostic to the datasets, bias types, model sizes, and data scales. First of all, we conduct a pilot empirical study (see §3) that tries to figure out what's more essential to a better biased model. We straightforwardly observed i) a higher the proportion of bias in the training data results in the more biased model, and ii) a more biased model has higher confidence in predicting the bias. This motivates us to propose a brand-new debiasing framework, dubbed bias-progressive auto-debiasing (BiasPAD), to obtain a better bias-only model by taking the inspiration of boosting learning. Specifically, we propose to alternate between the biased data selection and bias-only model training, where we use the most biased samples from the previous step to train the bias-only model. Given our progressively-improved bias-boosted model that accurately identifying the biased samples, we can simplely obtain a robust debiased model by a products-of-experts (PoE) loss (He et al., 2019).

We evaluate our approach in various settings and receive a huge improvement. To the best of our knowledge, our model delivers state-of-the-art performance on HANS (Zhang et al., 2019), NLI Hard (Gururangan et al., 2018) and FEVER-SYMMETRIC (Schuster et al., 2019) without leveraging extra data. We will open-source the code after publication.

2 Related Work

Bias in Datasets. *Dataset bias* is inevitable in most human-crafted datasets (Wang et al., 2018; 2019), such bias could be simple word co-occurrence (Gururangan et al., 2018), negation words (Utama et al., 2020a), or overlap relation between premise and hypothesis in natural language inference tasks (McCoy et al., 2019). Recent studies reveal that models can outperform random guesses by utilizing such bias as shortcuts (Tsuchiya, 2018; Poliak et al., 2018), whereas the performance of the fine-tuned models significantly drops when testing on a filtered bias-free dataset or on new complex samples. Debiasing methods are therefore very much needed to obtain robust models with reasoning skills to capture the underlying semantics.

Debiasing methods. Existing debiasing methods can be roughly classified as *data-centric* and *model-centric* methods. *Data-centric methods* focus on improving the quality of the training data by either i) removing the biased samples (Le Bras et al., 2020) or ii) generating new unbiased samples (Zhang et al., 2019; Wu et al., 2022). For example, Le Bras et al. (2020) adversarially filter the dataset biases and train the model on the filtered datasets. Zhang et al. (2019) generates additional training samples through controlled word exchange and back-translation, supplemented by human checks for fluency and paraphrase judgment. While promising, the researchers also showed that the newly constructed datasets are hardly to be entirely bias-free and may even introduce significant overhead, therefore it is crucial to build robust debiasing algorithms.

Model-centric methods share a common idea of building robust debiased models by reducing the importance of the biased instances during training. They first build a bias-only model to identify the biased instances, then reduce their importance in training by methods such as i) example reweighting (Schuster et al., 2019), i.e., downweighting the biased samples, ii) confidence regularization (Utama et al., 2020b), i.e., force the model to be less confident on the biased samples, and iii) product-of-experts (He et al., 2019; Mahabadi et al., 2020), which leads the model learning from less bias information by introcuding the output of the bias-only model to the training objective function. Current ways to build bias-only models are basically based on the observations by (Sanh et al., 2020) and (Utama et al., 2020a). Sanh et al. (2020) find a model with limited capacity (e.g., TinyBERT) can be more biased than the models with larger capacity. They train a TinyBERT on the whole training data and regard it as a bias-only model; then, they freeze the parameters of the TinyBERT and co-train it with the larger model, where the trained larger model will be regarded as the debiased model. Utama et al. (2020a) do an empirical study on a synthetic dataset and find a model will be more biased if trained on a smaller dataset at an early training stage. They then obtain their bias-only model by training a BERT-base model on a small fraction of the training dataset. However, both works cannot assure a strong biased model since the bias-only models are not trained on the bias-only datasets – the former utilize the entire dataset and the latter randomly select the subset

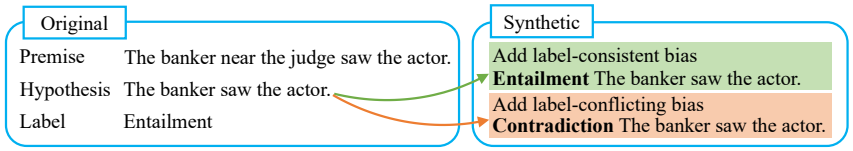

Figure 1: Example of the synthetic dataset, which is construct by inserting artificial shortcut in front of the hypothesis of original samples. Two types of synthetic bias, i.e., *label-consistent bias* and *label-conflicting bias*, are injected into the raw dataset.

of the training dataset to train the bias-only model, which can still bring general knowledge to the bias-only model. In this work, we introduce a new bias-progressive auto-debiasing framework based on two observations in Section 3, which ensures a stronger bias-only model and a robust debiased model.

3 Empirical Study

Task Definition. We target natural language understanding tasks and regard the tasks as a general multi-class classification problem. Given an input sentence pair $x \in X$, the goal is to predict the semantic relationship label $y \in \{1, 2, ..., K\}$, where K is the number of classes. Specially, we aim to obtain a robust debiased model F_d to make predictions free from reliance on the biased features $x_b \in x$ and focus on the unbiased features $x_u \in x$ only.

Insights about Debiasing Architectures. Generally, debiasing architectures have two stages: first, a bias-only model F_b is built to directly compute $P(y|x_b)$, which can be regarded as the confidence of a sample being biased; then, a debiased model F_d is learned to behave differently from the bias-only model by reducing the importance for the samples with high probability being biased. Existing methods to build bias-only models are basically based on the following findings: i) smaller models are easier to learn the bias information than larger models since the biased features are easier to access than unbiased features (Sanh et al., 2020), and ii) a model will be biased if training on a small fraction of the training dataset (Utama et al., 2020a). However, both findings cannot ensure a strong biased model, since they do not have constraints on the dataset used to train the bias-only model, where the model can still easily learn some general knowledge, especially on less-biased datasets. In this work, we propose a bias-progressive training strategy to obtain a more biased bias-only model without any additional prior knowledge. The strategy is mainly based on the following assumptions: **1)** the more biased samples in the training data, the more biased a model will be; **2)** the samples predicted by a bias-only model in high confidence are more likely to be biased.

Exploring with Synthetic Bias. To better verify these assumptions, we construct a controllable synthetic dataset by inserting artificial bias into the MNLI dataset (Williams et al., 2018) (dataset details can be found in Section 5.1). Figure 1 shows an example of the synthetic dataset. We simulate two types of bias by appending a specific string in front of the original hypothesis as a shortcut feature: One is *label-consistent bias*, which is constructed by inserting the golden label; Another is *label-conflicting bias*, where a random label other than the golden label will be appended to the raw hypothesis sentence. Specifically, we add the synthesized bias to $\eta \in [0, 1]$ percentage of the training dataset. For each instance, the injected bias could either be a *label-consistent bias* or a *label-conflicting bias* (anti-bias), with a ratio of 8:2 to simulate the real-world distribution. Besides, we construct two synthetic evaluation sets as a *label-consistent bias*-only set (*bias set*) and a *label-conflicting bias*-only set (*anti-bias set*). Ideally, a strong bias-only model should have learned the shortcuts, i.e., utilizing the inserted words as the predictions. Therefore it will have a large performance gap on the *bias set* and the *anti-bias set*.

We verify our first assumption by fine-tuning a BERT-base model on several synthesized training datasets with different $\eta \in \{0.1, 0.3, 0.5, 0.7\}$ and evaluate them on three evaluation sets, i.e., the original MNLI evaluation set, the bias set, and the anti-bias set. We can observe from Figure 2 that, at the early stages of the training process, the accuracy tends to increase to 100% on the *bias set* and drop to 0% on the *anti-bias set*, indicating the language models leaning to overfit superficial features in the first few training epochs, which is also proved in Utama's work (Utama et al., 2020a).

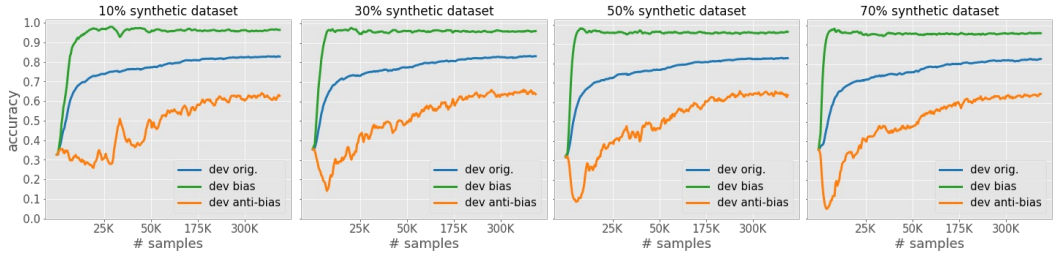

Figure 2: Learning dynamics of BERT-base models fine-tuned on four synthetic MNLI training datasets with different $\eta \in \{0.1, 0.3, 0.5, 0.7\}$. All models are evaluated on three evaluation sets, the original MNLI dev set, the bias set, and the anti-bias set.

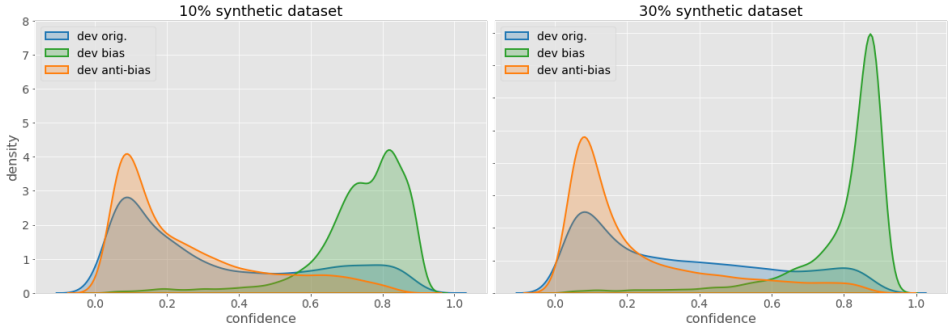

Figure 3: The confidence distribution of samples on three evaluation sets. Models are trained with 2000 random samples in synthetic MNLI datasets with different $\eta \in \{0.1, 0.3\}$ for three epochs.

Furthermore, as the proportion of the biased data η grows in the raw training data, i.e., more biased samples exist in the training data, the performance gap becomes more pronounced and stable between the *bias set* and the *anti-bias set*, and we will obtain a more biased model – which justify the first assumption that, the more biased samples in the training data, the more biased a model will be.

To verify the second assumption, we explore the distribution of the bias-only model's confidence on the three three evaluation sets. Figure 3 shows that bias-only model will make predictions with high confidence on the *label-consistent bias* samples, whereas the model has low confidence in predicting the label for the *label-conflicting bias* sample. We can observe apparent confidence deviation among the three evaluation datasets on a training dataset with a small fraction (i.e., 10%) of biased samples, such confidence deviation becomes more significant as the proportion of the biased samples in the training dataset increases. This observation suggests i) the bias-only model will have high confidence in predicting the biased samples, and ii) such confidence increases as the model becomes more biased.

In a summary, our observations are:

- The more biased samples in the training data, the more biased a model will be.
- The samples predicted by a bias-only model in high confidence are more likely to be biased, and such confidence increases as the bias-only model becomes more biased.

4 METHODOLOGY

4.1 OVERVIEW

We propose a **Bias**-**P**rogressive **A**uto-**D**ebiasing (BIASPAD) framework to automatically and sufficiently training debiased model without any requirements for the prior knowledge about the biases. Figure 4 shows the outline of the proposed framework, which includes i) a bias-boosted model

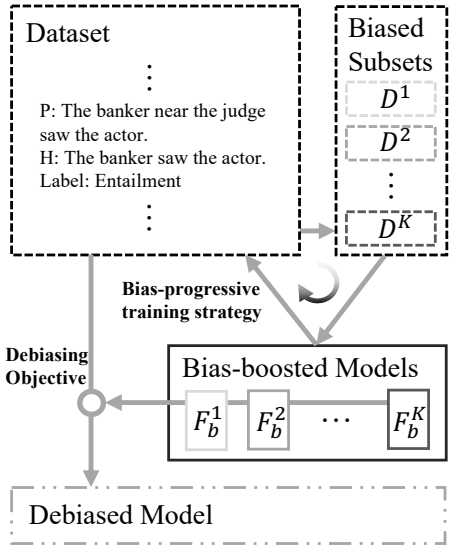

Figure 4: An overview of the bias-progressive auto-debiasing framework.

Algorithm 1 Bias-progressive Training

Input: dataset $\mathbb{D}$ with N samples; boost step K; subset size n; average coefficient λ
Output: bias-boosted model F_b^K
$s_i^1 \leftarrow 0 \; \forall i \in 1...N$
for $k \in 1...K$ **do**
  for $i \in 1...N$ **do**
    $w_i^k = \exp(s_i^k) / \sum_{j=1}^{N} \exp(s_j^k)$
  end for
  Sample $D^k \subset \mathbb{D}$ in size n based on w^k
  Re-initialize pre-trained F_b^k
  Finetune F_b^k on D^k with cross-entropy
  for $i \in 1...N$ **do**
    $\Delta s_i = P(\hat{y}_i^t | x_i, \theta_b^k)$
    $s_i^{k+1} \leftarrow \lambda * s_i^k + (1 - \lambda) * \Delta s_i$
  end for
end for

learned by a bias-progressive training strategy, and ii) a robust debiased model co-trained with the fixed bias-boosted model on the debiasing training objectives.

4.2 Bias-boosted Bias-only Model

Previous empirical studies reveal that a more biased bias-only model can be obtained by increasing the proportion of the biased samples in the training dataset. Since it is difficult to identify the exact biased samples without any prior knowledge, we use a bias-progressive training process to greedily learning from the most biased samples at each training step. Algorithm 1 shows the steps to obtain a bias-boosted model. First, given a dataset $\mathbb{D}$ with N samples, we initialize the bias scores $\{s_i | i \in (1, \ldots, N)\}$ for all samples $\{x_i | i \in (1, \ldots, N)\}$ as zero. Then, at each step k, the weight w_i^k for x_i to be sampled is calculated by $w_i^k = \exp(s_i^k) / \sum_{j=1}^{N} \exp(s_j^k)$, where $w_i^1 = 1/N$ for all samples at the first step. We then sample n instances $D^k \subset \mathbb{D}$ from the dataset with the weights $\{w_i^k | i \in (1, \ldots, N)\}$ to train the bias-only model with the loss $L_{\text{CE}} = CrossEntropy(y, F_b^k(x, \theta_b^k))$, where θ_b^k stands for the parameters of the bias-only model. At the end of each step, we update the bias score for all samples with $s^{k+1} = \lambda s^k + (1 - \lambda) \cdot P(\hat{y}^t | x_i, \theta_b^k)$, where $P(\hat{y}_i^t | x_i, \theta_b^k)$ is the confidence for model to predict true label of x_i, and λ is the moving average coefficient.

We repeat the above steps for K times to obtain the final bias-only model. At each step, **1)** we update the weights for the samples according to the confidence of the bias-only model – according to our *second observation*, the biased samples will have higher confidence score and thus will have higher weights to be sampled to form the subset D^k, and **2)** the sampled subset D^k will contain more biased samples after each step, meaning a more biased model F_b will be learned based on D^k according to our *first observation*, which in turn, a more biased model F_b will have more confidence in identifying the biased samples (according to our *second observation*) therefore increases the weights for the more biased samples for the next step. By learning with the above bias-progressive training strategy, we obtain a strong bias-boosted model.

4.3 Debiased Model Learning

After obtaining a bias-boosted bias-only model with the above steps, we now freeze the parameters of the bias-boosted model and train the debiased model through one of the two debiasing training objectives, i.e., example reweighting (Schuster et al., 2019) and product-of-experts (Sanh et al., 2020).

Example reweighting (ER) directly adjusts the weights of each training instance in the loss function based on the likelihood a training instance is biased, where the likelihood is obtained from the trained bias-boosted model F_b. The training objective for the debiased model F_d is:

$$L_{\text{ER}} = -\sum_{(x_i,y_i)\in\mathbb{D}} (1 - P(\hat{y}_i^t|x_i,\theta_b)) \log P(\hat{y}_i^t|x_i,\theta_d). \tag{1}$$

where $P(\hat{y}_i^t|x_i,\theta_b)$ is the confidence by the bias-only model for x_i to be its golden label, θ_d and θ_b are the parameters for the debiased model F_d and the bias-boosted model F_b, respectively.

Product-of-experts (PoE) encourages the debiased model to conpensate for the errors of the bias-boosted model, instead of sampling with frequently on the difficult samples. It learns the debiased model F_d via the following ensemble loss:

$$L_{\text{PoE}} = -\sum_{(x_i,y_i)\in\mathbb{D}} \log P(\hat{y}_i^t|x_i,\theta_d,\theta_b), \text{ where } P(\hat{y}_i|x_i,\theta_d,\theta_b) = \text{softmax}(\boldsymbol{l}_i^d + \boldsymbol{l}_i^b), \tag{2}$$

where $\boldsymbol{l}^d$ and $\boldsymbol{l}^b$ indicate the logits obtained from the debiased model and the bias-boosted model, respectively.

We mainly use product-of-expert to train the debiased model in the following experiments if without explicity.

5 Experiments

5.1 Evaluation Datasets

We evaluate our proposed **Bias**-**P**rogressive **A**uto-**D**ebiasing framework (BiasPAD) on two real-world natural language understanding tasks, i.e., natural language inference and fact verification.

Natural language inference (NLI) tasks predict for the relationship between two sentences such as *entailment* and *contradiction*. We select the widely used Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) to train the bias-boosted model and the debiased model, then evaluate the performance of the debiased model by fine-tuning it on three evaluation datasets: MNLI-dev, HANS (Zhang et al., 2019), and MNLI-Hard (Gururangan et al., 2018). MNLI dataset contains ∼392K pairs of premise and hypothesis labeled in three categories, i.e., *entailment*, *neural* and *contradiction*. MNLI-dev is the original evaluation set for the MNLI dataset. HANS is a challenging test set for NLI tasks, which includes ∼30K high word-overlapping sentence pairs generated by various templates with each sample labeled as *entailment* or *non-entailment*, where the two types of labels are equally distributed. One example for *non-entailment* is given a premise "the doctor was paid by the actor", the hypothesis is "the doctor paid the actor". In MNLI, the high word-overlapping pairs are highly correlated with the label *entailment*, where a model will typically perform random on HANS without the debiasing strategies. MNLI-Hard (Gururangan et al., 2018) is a subset of MNLI-dev which consists of only challenging samples.

Fact verification tasks predict for whether an evidence can support the given claim. Fact Extraction and Verification (FEVER) dataset (Thorne et al., 2018) is a commonly used dataset for this task, it consists of ∼145K pairs of claims and evidence with each pair marked as *supporting*, *refuting*, and *insufficiently informative*. We use FEVER to train the bias-boosted model and the debiased model and evaluate the performance of the debiased model on two evaluation sets: one FEVER-dev, which is the original evalution set for FEVER; another is SYMMETRIC (Schuster et al., 2019), which is a challenging test set synthesized based on the original sentence pairs in FEVER by inserting conflict facts. Models rely heavily on negation words such as "not" or "reject" will face a huge performance drop on this evaluation set.

5.2 Implementation Details

According to the observations from our empirical study and the work by (Utama et al., 2020a), we set the number of samples in the subset to train the bias-only model n as 2000, and train the bias-only model for 3 epochs in each iteration. To obtain a bias-boosted model, the number of iterations

Table 1: Comparison results on the evaluation datasets in accuracy, where HANS-Ent, HANS-Non-Ent, and HANS-Total are the results for the *entailment* labeled samples, *non-entailment* labeled samples, and all samples, respectively.

	Objective	MNLI dev	HANS			MNLI Hard	FEVER	
			Total	Ent	Non-Ent		Dev	Symm.
BERT-base	CE	**84.52**	62.43	**98.12**	26.74	76.96	85.60	63.10
Mahabadi et al. (2020)	PoE	84.19	64.65	95.99	33.30	76.81	86.46	66.25
Utama et al. (2020a)	PoE	80.70	68.50	86.24	50.76	-	85.40	65.30
Utama et al. (2020a)	ER	81.40	68.60	87.06	50.14	-	87.20	65.60
Sanh et al. (2020)	PoE	81.35	68.77	81.13	56.41	76.54	-	-
BiasPAD	ER	83.35	71.23	86.54	55.92	77.25	87.60	65.31
BiasPAD	PoE	82.24	**73.82**	87.64	**60.20**	**77.48**	**87.80**	**66.62**

for the bias-progressive training process K is set to 3, which is verified to be enough to convergence; the moving average coefficient λ in Section 4.2 is set to 0.5.

Both the bias-only model and the debiased model are fine-tuned on a BERT-base model (Devlin et al., 2019) with $\sim$110M parameters. The embedding size is set to 32 and the learning rate is set to 2e-5. The learning rate for the debiased model is linearly increased for 2000 warming steps and linearly decreased to 0 afterward, while keeping 2e-5 for the bias-boosted model. We use an Adam optimizer with default hyperparameters.

5.3 Main Results

We compare our proposed bias-progressive auto-debiasing framework with a BERT-base model trained with cross-entropy loss as the baseline and four existing state-of-the-art model-centric debiasing frameworks. Mahabadi et al. (2020) utilize prior knowledge of the bias types to identify the biased samples to train a bias-only model then train the debiased model via product-of-experts (PoE). Utama et al. (2020a) obtain the bias-only model by training it on a small fraction of the training dataset, and train their debiased model by either PoE or example reweighting (ER). Sanh et al. (2020) get their bias-only model by training BERT-tiny on the whole training dataset and obtain the debiased model via PoE. Our proposed BiasPAD framework and the two latter works do not require the prior knowledge about the dataset bias. Table 1 shows the comparison results on the evaluation datasets, where the results for the comparison methods are collected from the original papers, and our results are the results averaged from five trials.

The proposed BiasPAD framework achieves SOTA results on the three challenging test sets, i.e., HANS, MNLI-Hard, and FEVER-Symm, by 5.1%, 0.7% and 0.4%, compared with all the previous SOTA results on each dataset by either prior knowledge-available or prior knowledge-free frameworks. Specifically, we significantly outperforms two other prior knowledge-free frameworks on all three challenging test sets by 5.1%, 1.0%, and 1.0%, respectively, which indicates the proposed BiasPAD framework has better performance in automatic bias capturing and debiasing. Comparing our framework with the framework utilizing the manual prior knowledge, the proposed BiasPAD still steadily provides better performance, which suggests the proposed framework may also capture unknown bias that may be hardly identified by a human, exhibiting a strong generalization capability. Under different training objectives, the proposed BiasPAD consistently outperms the other works with the same training objective, indicating the effectiveness of the bias-progressive training strategy to obtain a strong bias-only model. Comparing the BERT-base model, with the other debiasing frameworks, we can see that all debiased models show degradation on the in-distribution datasets, i.e., MNLI-dev and HANS Ent, where BiasPAD shows the minimal reduction among all the knowledge-free debiasing methods, showing the advantage of the proposed framework. Overall, we obtain a stronger bias-boosted model and a robust debiased model through the proposed bias-progressive auto-debiasing (BiasPAD) framework.

6 ANALYSIS

6.1 THE BIAS-BOOST MODEL IS A MORE BIASED MODEL

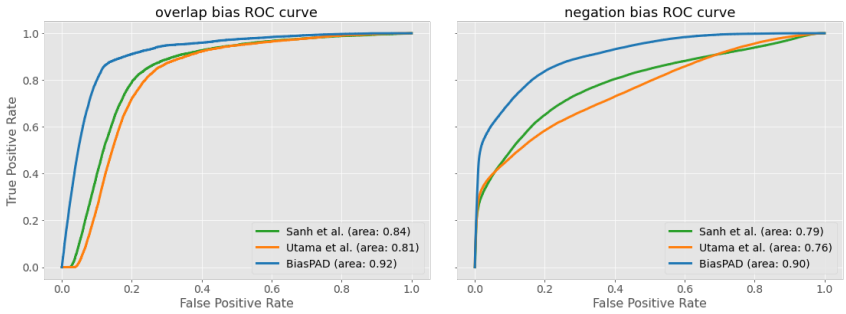

Figure 5: AUC-ROC curve on datasets synthesized for two known biases on MNLI-dev dataset by three bias-only models trained with different strategies.

In Section 5.3 we show that the proposed bias-progressive training strategy ensures a strong bias-only model compared with the methods by Utama et al. (2020a) and Sanh et al. (2020). In this part, we visualize such bias difference in the bias-only model by the following experiments. We first obtained three bias-only models by either the strategy used in Utama et al. (2020a), Sanh et al. (2020), or our bias-progressive training strategy. We reproduce the two other methods with the suggested hyper-parameters claimed in the original papers, and the details can be found in Appendix. We further synthesize two evaluation sets based on MNLI-dev set: one labels i) the samples with high word overlap rate and with *entailment* as the true label to be 1 and ii) the other samples to be 0; another labels i) the samples containing negation words in hypothesis and with *contradiction* as the true label to be 1 and ii) the other samples to be 0. The details for the calculation of the word overlap rate and the list for the negation words can be found in the Appendix. We then verify the performance of the above three bias-only models on the two synthesized datasets, where Figure 5 shows the AUC-ROC curve based on the confidence of the bias-only model. From the result, we discover that our bias-progressive training strategy outperforms the other two on both types of biases, with higher AUC scores and dominant ROC curves. This experiment shows, our bias-boosted model has stronger ability in discriminating the two well-known biases compared with the others. That is, our bias-boost model is a more biased model.

6.2 NUMBER OF ITERATIONS TO OBTAIN THE BEST BIAS-BOOSTED MODEL

We obtain our bias-boosted model through a bias-progressive training strategy, i.e., we step-wisely train the bias-only model. One key question is, how many iterations do we need to obtain the best bias-boosted model? To answer this question, we design this experiment to observe the model convergence process during the bias-progressive training. Similar to Section 6.1, we verify the results on two synthesized biased datasets based on the MNLI-dev dataset. We iterate the bias-progressive training process for six steps, i.e., $K = 6$, and at each step, we evaluate the bias-only model on the two synthesized datasets and record their AUC scores. The results are shown in Figure 6. We can see that at the first three iterations, the AUC scores increase from 0.82/0.74 to 0.92/0.90 for the two evaluation sets, while just show slight fluctuation after the fourth iteration. We draw two conclusions as 1) the bias-boosted model converges through

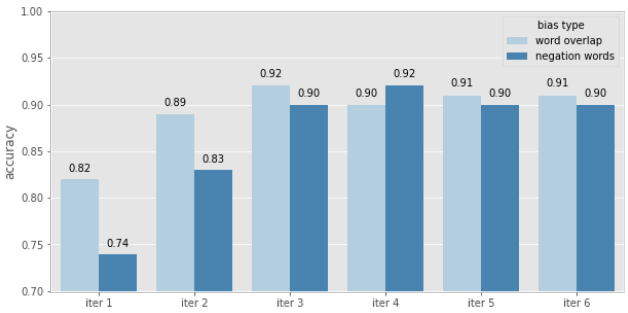

Figure 6: AUC scores for two datasets synthesized for two known biases on MNLI-dev dataset by our bias-boosted models at different iterations.

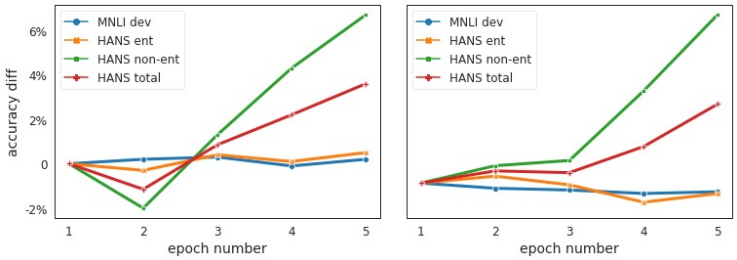

Figure 7: The x-axis indicates for the t-th epoch. **Left**: Accuracy difference of injecting cross-entropy loss at t-th epoch for only one epoch. **Right**: Accuracy difference of injecting cross-entropy loss starting at t-th epoch. For a clearer contrast, we show the difference value against the leftmost point.

the bias-progressive training progress, and 2) the model converges in the first few iterations, where we select $K = 3$ to obtain the bias-boosted model in our experiments.

6.3 Performance Tradeoff between In-distribution and Out-of-distribution Sets

We investigate the trad-off between in- and out-of-domain performance of a debiased model by setting the training objective as a multi-loss function (Sanh et al., 2020):

$$L = L_{\text{PoE}} + \alpha L_{\text{CE}}, \tag{3}$$

where L_{CE} is a normal cross-entropy loss, and α is the parameter to adjust the tradeoff. Intuitively, if fine-tuning a BERT-base model with only the cross-entropy loss, we will obtain a biased BERT-base model which is just same as the baseline BERT-base model we compared in Section 1. One advantage of introducing the cross-entropy loss (CE) is to improve the in-distribution performance since we noticed a performance drop on the in-distribution datasets with using debiasing strategies. So one question is, can we obtain a debiased model that has good performance in both in- and out-of-distribution performance by training it with the objective 3? We answer this question by observing the performance tradeoff in two strategies: 1) we insert the cross-entropy loss at t-th training epoch; or 2) we continually insert the cross-entropy loss from the t-th training epoch. Figure 7 shows the performance of the two strategies on four evaluation datasets, where MNLI-dev and HANS-ent can be regarded as in-distribution sets, and HANS-not-ent can be regarded as the out-of-distribution set. For the first strategy, we can see that adding the CE loss at a later stage improves the out-of-distribution performance while preserves the in-distribution performance. For the second strategy, we also observe a better tradeoff performance in a later training stage. Therefore, we condlude that a better tradeoff performance between in- and out-of-distribution can be achieved by adding the CE loss in a late stage during the debiased model training process.

7 Conclusion and Future Work

Reducing the impact of dataset bias plays a crucial role in natural language understanding tasks. In this paper, we propose a general bias-progressive auto-debiasing (BiasPAD) framework to obtain a strong bias-boosted model and a debiased model that are robust to dataset bias. We post and verify two assumptions as i) a bias-only model will be more biased if training on a dataset containing more biased samples and ii) a bias-only model has high confidence in predicting the biased samples, and such confidence increases as the bias-only model becomes more biased. We design a bias-progressive training strategy based on the above two observations and obtain a strong bias-boosted model. A robust debiased is then obtained by training with the bias-boosted model. The proposed BiasPAD achieves the state-of-the-art results on three challenging datasets by improving the SOTA results than two other debiasing-frameworks with 5.1%, 1.0%, and 1.0%, respectively. Our further experiments also prove the effectiveness of the proposed BiasPAD framework in many ways. In future works, we plan to investigate how the bias-only model and the debiased model can learn from the mutual knowledge for the further enhancement.

REFERENCES

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 4067–4080. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1418.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/n18-2017.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *EMNLP-IJCNLP 2019*, pp. 132, 2019.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *ICML*, 2020.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 8706–8716. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.769.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3428–3448. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/p19-1334.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In Malvina Nissim, Jonathan Berant, and Alessandro Lenci (eds.), *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pp. 180–191. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/s18-2023.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2020.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 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 (EMNLP-IJCNLP)*, pp. 3419–3425, 2019.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New*

Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pp. 809–819. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/n18-1074.

Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/786.html.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 7597–7610. Association for Computational Linguistics, 2020a. URL https://doi.org/10.18653/v1/2020.emnlp-main.613.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 8717–8729. Association for Computational Linguistics, 2020b. URL https://doi.org/10.18653/v1/2020.acl-main.770.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi (eds.), *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pp. 353–355. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/w18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3261–3275, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/n18-1101.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. Generating data to mitigate spurious correlations in natural language inference datasets. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2660–2676. Association for Computational Linguistics, 2022. URL https://doi.org/10.18653/v1/2022.acl-long.190.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 1298–1308. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/n19-1131.