Feature-Level Debiased Natural Language Understanding

Yougang Lyu[1], **Piji Li**[2], **Yechang Yang**[1], **Maarten de Rijke**[3], **Pengjie Ren**[1]
Yukun Zhao[1,4], **Dawei Yin**[4], **Zhaochun Ren**[1*]

[1]School of Computer Science and Technology, Shandong University, Qingdao, China
[2]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[3]University of Amsterdam, Amsterdam, The Netherlands
[4]Baidu Inc., Beijing, China
{youganglyu, yyc002}@mail.sdu.edu.cn, pjli@nuaa.edu.cn, m.derijke@uva.nl, jay.ren@outlook.com,
zhaoyukun02@baidu.com, yindawei@acm.org, zhaochun.ren@sdu.edu.cn

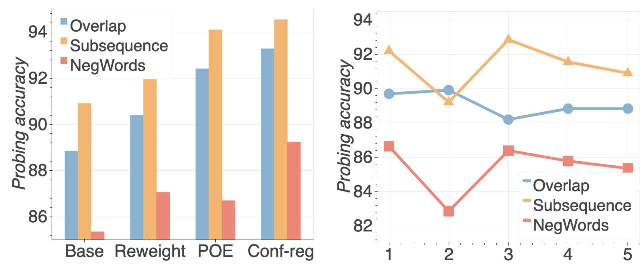
Abstract

Natural language understanding (NLU) models often rely on *dataset biases* rather than intended task-relevant features to achieve high performance on specific datasets. As a result, these models perform poorly on datasets outside the training distribution. Some recent studies address this issue by reducing the weights of biased samples during the training process. However, these methods still encode biased latent features in representations and neglect the dynamic nature of bias, which hinders model prediction. We propose an NLU debiasing method, named debiasing contrastive learning (DCT), to simultaneously alleviate the above problems based on contrastive learning. We devise a debiasing, positive sampling strategy to mitigate biased latent features by selecting the least similar biased positive samples. We also propose a dynamic negative sampling strategy to capture the dynamic influence of biases by employing a bias-only model to dynamically select the most similar biased negative samples. We conduct experiments on three NLU benchmark datasets. Experimental results show that DCT outperforms state-of-the-art baselines on out-of-distribution datasets while maintaining in-distribution performance. We also verify that DCT can reduce biased latent features from the model's representations.

1 Introduction

Pre-trained language models such as BERT (Devlin et al. 2019) have achieved impressive performance on many NLU benchmarks, such as natural language inference (NLI) (Bowman et al. 2015; Williams, Nangia, and Bowman 2018) and fact verification (Thorne et al. 2018). However, recent studies have shown that these models tend to leverage *dataset biases* instead of intended task-relevant features (McCoy, Pavlick, and Linzen 2019; Schuster et al. 2019; Du et al. 2022). For example, Gururangan et al. (2018) find that NLU models rely on the spurious association between negative words (e.g., *nobody*, *no*, *never* and *nothing*) and *contradiction* labels for prediction in NLI datasets, leading to low accuracy on out-of-distribution datasets that lack spurious associations.

To mitigate bias in training datasets, recent NLU debiasing methods attempt to train more robust models. Three prevailing debiasing methods exist in NLU: (i) example reweighting (Reweight) (Schuster et al. 2019), (ii) product-of–experts (POE) (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Mahabadi, Belinkov, and Henderson 2020), and (iii) confidence regularization (Conf-reg) (Utama, Moosavi, and Gurevych 2020a). These debiasing methods encourage the model to pay less attention to the biased examples, which forces it to learn harder samples to improve out-of-distribution (OOD) performance.



(a) Different debiasing methods (b) Different training epochs

Figure 1: Probing accuracy for three types of biased features (Overlap, Subsequence and Negwords) with different methods and different training epochs on the MNLI dataset. (a) Recent NLU debiasing methods (Reweight, POE, and Conf-reg) have a higher probing accuracy of biased features compared to BERT-base. (b) Biased features probing accuracy of BERT-base changes dynamically during the training process.

From the perspective of debiasing NLU, two main challenges remain. Both concern *biased features*, that is, features that have spurious correlations with the label, e.g., negative words in input sentences in NLU tasks. First, existing NLU debiasing methods still encode biased latent features in representations. We follow Mendelson and Belinkov (2021) to use probing tasks for several types of bias (Overlap, Subsequence and Negwords) to verify whether biased latent features have been removed from representations. The probing task for bias is to predict whether a sample is biased based on the model's representation in terms of *probing accuracy*, the accuracy of the probing task. Higher probing accuracy of bias means that the model's representation contains more biased features. Fig. 1(a) shows that existing debiasing methods have a higher probing accuracy of three types of biased latent features from

*Corresponding author.

the representations than the fine-tuned BERT-base. These results illustrate that existing debiasing methods do not improve OOD performance by reducing biased features and modeling intended task-relevant features, but by adjusting the conditional probability of labels given biased features. Since these debiasing methods only adjust the conditional probability of labels given biased features, they improve OOD performance at the cost of degrading in-distribution (ID) performance. This poses a feature-level debiasing challenge to debiasing approaches.

Second, existing NLU debiasing methods neglect the dynamic influence of bias. The number of biased latent features in a representation changes during the training process. Since the model predicts the label based on the representation, biased features dynamically influence model prediction during the training process. In Fig. 1(b) we examine three types of bias and find that the probing accuracy of biases changes during training. Different types of biased features have a different influence on model prediction during training. It is important to reduce biased features that have the greatest influence on model prediction at different training epochs. This poses a challenge in capturing the dynamic influence of bias.

To tackle the above challenges, we propose a novel debiasing method, *debiasing contrastive learning* (DCT). The main idea of DCT is to encourage positive examples with least similar bias to be closer and negative examples with most similar bias to be apart at the feature-level. DCT consists of two strategies: (i) a debiasing, positive sampling strategy to mitigate biased latent features, and (ii) a dynamic negative sampling strategy to capture the dynamic influence of biased features. As to the first strategy, DCT's debiasing, positive sampling strategy selects the least similar biased positive samples from the debiasing dataset. We filter debiasing samples from the training set where the bias-only model has high confidence but incorrect predictions. As the bias-only model relies only on biased features to make predictions, the debiasing dataset contains biased features, but the correlation between biased features and labels differs from the dominant spurious correlation in the training dataset. As to the second strategy, DCT's dynamic negative sampling strategy uses the bias-only model to dynamically select the most similar biased negative sample during the training process. Furthermore, we adopt *momentum contrast* (He et al. 2020) to establish a massive queue for saving representations dynamically.

We conduct experiments on three NLU benchmark datasets to evaluate bias extractability and debiasing performance of our proposed method. DCT outperforms state-of-the-art baselines on OOD datasets and maintains ID performance by reducing multiple types of biased features in a model's representations.

To sum up, our contributions are as follows:

- To the best of our knowledge, we are the first to focus on feature-level debiasing and modeling the dynamic influence of bias in NLU tasks at the same time.
- We propose a novel debiasing method, named DCT, which uses contrastive learning and combines a debiasing, positive sampling strategy and a dynamic negative sampling strategy to reduce biased latent features and capture the dynamic influence of biases.
- Experiments on three NLU benchmark datasets show that DCT reduces biased latent features in the model's representation and outperforms state-of-the-art baselines on OOD datasets while maintaining ID performances.[1]

2 Related Work

2.1 Dataset Bias

Exploiting *dataset biases* seems easier for deep neural networks than learning the intended task-relevant features. (Geirhos et al. 2020; Bender and Koller 2020). For instance, models can perform better than most baselines by only partially using input in NLI without capturing the semantic relationships between premises and hypothesis sentences (Gururangan et al. 2018). Similar phenomena have been observed in other tasks, e.g., visual question-answering (Agrawal, Batra, and Parikh 2016), reading comprehension (Kaushik and Lipton 2018), and paraphrase identification (Zhang, Baldridge, and He 2019). Prior work has constructed challenge datasets consisting of "counter examples" to superficial cues that deep neural networks might adopt (Jia and Liang 2017; Glockner, Shwartz, and Goldberg 2018; Naik et al. 2018; McCoy, Pavlick, and Linzen 2019). When models are evaluated on these challenge datasets, their performance often drops to the same as the random baseline (Gururangan et al. 2018; Schuster et al. 2019). Therefore, there is a clear need for methods that are tailored to address NLU dataset biases.

2.2 Debiasing NLU Methods

Several studies aim to mitigate dataset bias by improving dataset construction techniques. For example, Zellers et al. (2019); Sakaguchi et al. (2021) reduce biased patterns in datasets with adversarial filtering; Nie et al. (2020); Kaushik, Hovy, and Lipton (2020) adopt a dynamic, human-in-the-loop data collection technique; Min et al. (2020); Schuster, Fisch, and Barzilay (2021) use adversarial samples to augment the training dataset; and Wu et al. (2022); Ross et al. (2022) train data generators to generate debiased datasets. A complementary line of work trains more robust models with alternative learning algorithms, such as product-of-experts (POE) (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Mahabadi, Belinkov, and Henderson 2020), confidence regularization (Conf-reg) (Utama, Moosavi, and Gurevych 2020a; Du et al. 2021), and example reweighting (Reweight) (Schuster et al. 2019). These algorithms can be formalized as two-stage frameworks; the first step is to train a bias-only model, either automatically (Utama, Moosavi, and Gurevych 2020b; Sanh et al. 2021; Ghaddar et al. 2021) or using prior knowledge about the bias (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Belinkov et al. 2019b,a); at the second stage, the output of the bias-only model is used to adjust the loss function of the debiased model.

However, these debiasing methods make biased features more extractable from the model representations (Mendelson and Belinkov 2021; Du et al. 2022). Instead, we aim to use contrastive learning to dynamically push the NLU

[1]The code is available at https://github.com/youganglyu/DCT

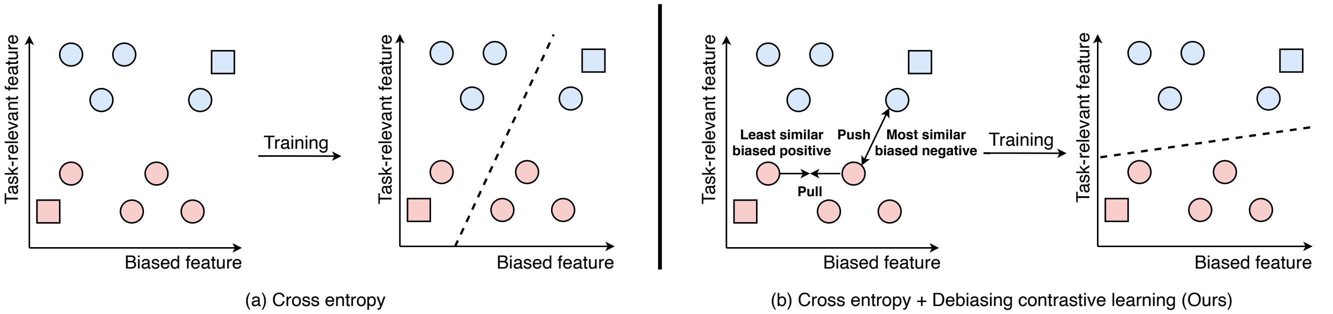


Figure 2: (a) Models fine-tuned with cross-entropy use biased features to predict. (b) Models fine-tuned with cross-entropy and DCT reduce biased features.

model to reduce biased features and capture intended task-relevant features. This enables the NLU model to improve OOD performance while maintaining ID performance.

2.3 Contrastive Learning

The main idea of contrastive learning is to encourage the representation of similar samples to be close and different samples to be apart (Hadsell, Chopra, and LeCun 2006; Chen et al. 2020). Contrastive learning has been used to improve improve in-distribution performance in computer vision (CV; He et al. 2020; Chen et al. 2020) and natural language processing (NLP; Giorgi et al. 2021; Wang et al. 2021). In a self-supervised framework, positive (i.e., similar) samples can be generated by data augmentation of the anchor sample, and negative (i.e., different) samples can be obtained from the same batch (Gao, Yao, and Chen 2021) or from a memory bank/queue that saves the representation of previous samples (Wu et al. 2018; Chen et al. 2020). In a supervised framework, positive samples belong to the same class while negative samples belong to a different class (Khosla et al. 2020; Gunel et al. 2021; Li et al. 2021).

These methods focus on improving ID performance, while we aim to use contrastive learning to reduce biased latent features and improve OOD performance.

3 Method

In this section, we detail the DCT method. First, we formulate our research problem. Then, we introduce the overall framework of debiasing contrastive learning. Next, we introduce the debiasing, positive sampling strategy and describe the dynamic negative sampling strategy. Finally, a training process with momentum contrast for DCT is explained.

3.1 Problem Formulation

Following (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Mahabadi, Belinkov, and Henderson 2020; Utama, Moosavi, and Gurevych 2020a), we formulate NLU tasks as a general classification problem. We denote a training dataset as $\mathcal{D}$ consisting of N examples $\{x_i, y_i\}_{i=1}^{N}$, where $x_i \in \mathcal{X}$ is the input data, $y_i \in \mathcal{Y}$ is the target label, $|\mathcal{Y}| = K$ is the number of the classes. For each input instance x, we assume that the features of x can be divided into intended task-relevant features x^t and biased features x^b, where x^t have invariant relations with the label y and x^b have spurious relations with the label y. The random variables of x, y, x^b and x^t are respectively denoted as X, Y, X^b and X^t. Our goal is to train a debiasing model f_d to capture $\mathbb{P}_D(Y|X^t)$ by reducing biased features X^b, which performs better on OOD datasets and maintains ID performance.

3.2 Debiasing Contrastive Learning

DCT aims to pull the least similar biased positive samples closer to each other and push the most similar biased negative samples apart at the feature-level, as illustrated in Fig. 2(b). To accomplish this, we rewrite the contrastive loss to obtain the *debiasing contrastive learning* loss as follows:

$$\mathcal{L}_{DCT} = -\frac{1}{|S_i^p|} \sum_{x_j \in S_i^p} \left(\log \frac{\exp\big(\Phi_d(x_i) \cdot \Phi_{d'}(x_j)/\tau\big)}{\sum\limits_{x_k \in S_i^n / \{x_j\}} \exp(\Phi_d(x_i) \cdot \Phi_{d'}(x_k)/\tau)} \right), \quad (1)$$

where $\Phi_d(\cdot)$ refers to the debias encoder, $\Phi_d(\cdot)$ denotes to the momentum encoder, $|S_i^p|$ is the size of the debiasing, positive sample set S_i^p for x_i, S_i^n denotes the negative sample set for x_i, and τ is a scalar temperature parameter.

In the following subsections, we detail the debiasing, positive sampling strategy and the dynamic negative sampling strategy.

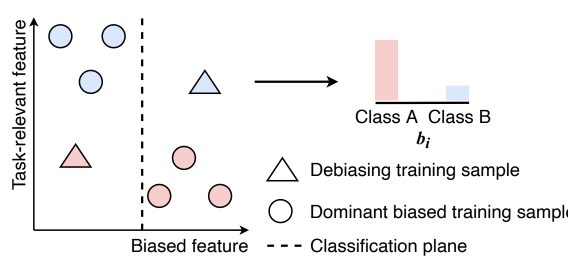


Figure 3: The bias-only model is used to filter the debiasing dataset from the original training dataset.

3.3 Debiasing Positive Sampling

To mitigate biased latent features, we employ a bias-only model to filter the debias dataset from the training dataset and sample least similar biased positive samples from the debias dataset, i.e., we train a bias-only model f_b to approximate $\mathbb{P}_D(Y|X^b)$ which makes predictions only based on biased features. Following Sanh et al. (2021), we use a limited capacity weak learner as the bias-only model, which is trained on the full training dataset. As sketched in Fig. 3, if the bias-only model is particularly confident in predicting a sample but predicts incorrectly, it is likely that the sample contains biased features, but the correlation between biased features and labels differs from the dominant spurious correlation in the training dataset. Given a training example $\{x_i, y_i\}$, we assume the output of bias-only model f_b to be $b_i = \langle b_{i,1}, b_{i,2}, \ldots, b_{i,K}\rangle$. Based on the probabilistic distributions b_i, we filter the debiasing dataset from the training dataset, so we have:

$$\mathcal{D}_{debias} = \{\{x_i, y_i\}|b_{i,c} \geq \lambda \wedge y_{i,c} = 0\}, \tag{2}$$

where c is the predicted class by the bias-only model, $b_{i,c}$ denotes the scalar probability value of the class c, $y_{i,c}$ refers to the ground truth of class c for x_i, and λ is a scalar threshold.

To construct the positive sample set S_i^p for x_i in Eq. 1, we select positives that are least similar to x_i in the debiasing dataset by L2 distance. Additionally, we incorporate the debiasing dataset into the training dataset to generate a sufficient number of positive pairs.

3.4 Dynamic Negative Sampling

To capture the dynamic influence of bias, we employ the bias-only model to dynamically select the most similar biased negative sample that is closest to the anchor sample x_i. Based on the checkpoints of the bias-only model, we apply the encoder of the bias-only model of each epoch to all the samples and dynamically retrieve the most similar biased negative sample set, that is:

$$\mathcal{S}_{i,k}^{dn} = \{x_j|y_j \neq y_i \wedge \arg\min_j(d(\Phi_b^k(x_i), \Phi_b^k(x_j)))\}, \tag{3}$$

where $\mathcal{S}_{i,k}^{dn}$ denotes the set of most similar biased negative samples of x_i for the k-th epoch, $\Phi_b^k(\cdot)$ is the bias-only model encoder for the k-th epoch, and $d(\cdot)$ refers to the L2 distance function.

To construct the negative sample set S_i^n for x_i in Eq. 1, we combine $\mathcal{S}_{i,k}^{dn}$ and other negative samples in the momentum contrast queue.

3.5 Training with Momentum Contrast

To leverage a large number of positive and negative samples, we adopt momentum contrast (He et al. 2020) to build a massive queue for dynamically saving representations. In order to maintain the representation consistency in the queue, the momentum contrast framework requires two encoders, a debias encoder and a momentum encoder. During training with the momentum contrast framework, the parameter θ_d in the debias encoder is updated by training samples and then the $\theta_{d'}$ in the momentum encoder is updated by:

$$\theta_{d'} \leftarrow m\theta_d + (1-m)\,\theta_{d'}, \tag{4}$$

where $m \in [0, 1)$ is a momentum coefficient, which keeps the consistency of sample representations in the queue. The sample representations in the queue are gradually replaced. Specifically, the sample representations encoded by the momentum encoder are added to the queue, and the oldest sample representations are removed. In each training iteration, only the parameter θ_d in the debias encoder is updated by back-propagation.

To directly use the label information, we adopt cross entropy as part of the overall loss for training the main model:

$$\mathcal{L}_{CE} = -y_i \cdot \log f_d\,(x_i)\,. \tag{5}$$

The overall loss function is formally computed as:

$$\mathcal{L} = (1-\alpha)\,\mathcal{L}_{CE} + \alpha\mathcal{L}_{DCT}, \tag{6}$$

where α is a scalar weighting hyperparameter.

4 Experiments

4.1 Research Questions

We conduct experiments on different NLU tasks to answer the following research questions: (RQ1) Does the proposed DCT method reduce multiple types of biased features simultaneously? (RQ2) How does the proposed DCT perform on ID and OOD datasets compared to state-of-the-art baselines? (RQ3) How do strategies and hyperparameters affect ID and OOD performances of DCT?

4.2 Datasets

We use three NLU benchmark datasets in our experiments:

- **MNLI** – The MNLI dataset (Williams, Nangia, and Bowman 2018) contains pairs of premise and hypothesis sentences labeled as *entailment*, *neutral*, and *contradiction*. We test models trained on MNLI against the challenge dataset HANS (McCoy, Pavlick, and Linzen 2019). It contains examples of high overlap between premises and hypothetical sentences but are labeled as *contradiction*. Since the overlapping feature is correlated with label *entailment* in MNLI, models trained directly on MNLI tend to perform poorly on HANS.
- **SNLI** – The SNLI dataset (Bowman et al. 2015) contains pairs of premise and hypothesis sentences labeled as *entailment*, *neutral*, and *contradiction*. Following Utama et al. (2021), we evaluate models trained on SNLI against the long and short subsets of the Scramble Test challenge set (Dasgupta et al. 2018). It contains samples that are changed the word order against the overlap bias in SNLI dataset.
- **FEVER** – The FEVER dataset (Thorne et al. 2018) contains pairs of claim and evidence sentences labeled as either *support*, *not-enough-information*, or *refute*. We follow Schuster et al. (2019) to process and split the dataset.[2] FEVER models rely on the claim-only bias, where specific words in the claim are often associated with target label. We evaluate models trained on FEVER against Fever-Symmetric datasets (Schuster et al. 2019) (version 1 and 2), which were manually constructed to reduce claim-only bias.

[2]https://github.com/TalSchuster/FeverSymmetric

Method	Overlap		Subsequence		NegWords	
	Compression	Acc.	Compression	Acc.	Compression	Acc.
BERT-base	3.29 ± 0.16	88.84 ± 1.22	3.21 ± 0.24	90.91 ± 2.53	2.44 ± 0.12	85.35 ± 0.93
Reweight	3.68 ± 0.12	90.39 ± 0.79	3.66 ± 0.11	92.21 ± 2.30	2.53 ± 0.11	87.06 ± 0.22
POE	3.68 ± 0.10	92.41 ± 1.27	3.82 ± 0.17	94.10 ± 1.97	2.47 ± 0.09	86.70 ± 0.85
Conf-reg	4.34 ± 0.20	93.28 ± 0.69	4.03 ± 0.18	94.54 ± 1.82	2.70 ± 0.11	89.24 ± 0.27
DCT	**2.79 ± 0.15**	**85.35 ± 0.84**	**2.76 ± 0.13**	**88.96 ± 1.77**	**1.88 ± 0.12**	**80.31 ± 0.35**

Table 1: Results of probing for Overlap, Subsequence, and NegWords on MNLI. Acc is the probing accuracy of biases. Note that lower compression scores and probing accuracy represent lower extractability of biased features in the model representation.

Method	Overlap		Subsequence		NegWords	
	Compression	Acc.	Compression	Acc.	Compression	Acc.
BERT-base	4.39 ± 0.17	93.82 ± 0.99	5.10 ± 0.23	94.96 ± 1.44	4.08 ± 0.36	93.26 ± 0.66
Reweight	4.72 ± 0.28	92.79 ± 1.05	5.44 ± 0.50	95.22 ± 1.62	4.17 ± 0.29	93.33 ± 0.53
POE	4.78 ± 0.14	93.22 ± 0.47	5.14 ± 0.15	94.34 ± 1.18	4.28 ± 0.22	94.12 ± 0.52
Conf-reg	5.20 ± 0.19	94.81 ± 0.46	5.67 ± 0.16	93.45 ± 2.86	4.78 ± 0.29	95.70 ± 0.49
DCT	**3.14 ± 0.10**	**90.63 ± 0.51**	**3.44 ± 0.15**	**91.86 ± 2.19**	**2.25 ± 0.14**	**85.17 ± 0.75**

Table 2: Results of probing for Overlap, Subsequence and NegWords on SNLI. Notational conventions are the same as in Table 1.

Method	Compression	Acc.
BERT-base	2.57 ± 0.08	81.82 ± 0.75
Reweight	2.68 ± 0.12	83.72 ± 1.88
POE	2.77 ± 0.05	84.60 ± 1.08
Conf-reg	2.82 ± 0.06	82.47 ± 0.90
DCT	**2.21 ± 0.08**	**78.27 ± 1.04**

Table 3: Results of probing for NegWords on FEVER. The notation here is consistent with Table 1.

4.3 Baselines and Evaluation Metrics

We compare DCT with three state-of-the-art debiasing methods: (i) Example reweighting (Reweight) (Schuster et al. 2019) adjusts the importance of each training instance by computing the importance weight. The weight scalar for each training instance x_i is computed as $1 - b_{i,g}$, where $b_{i,g}$ is the probability of the bias-only model predicting the gold label. (ii) Product-of-experts (POE) (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Mahabadi, Belinkov, and Henderson 2020) trains a debiased model by ensembling with the bias-only model. (iii) Confidence regularization (Conf-reg) (Utama, Moosavi, and Gurevych 2020a) regularizes model confidence on biased training examples. Conf-reg uses a self-distillation training objective and scales the teacher model output by the bias-only model's output.

To measure the extractability of biased features in the model's representation, we follow Mendelson and Belinkov (2021) to use compression scores and probing accuracy.[3] The compression score is defined as $compression = \frac{L_{unif}}{L_{online}}$, where $L_{unif} = |D| \log K$ denotes the uniform distribution over the K labels and L_{online} is the online coding proposed by Voita and Titov (2020). The probing accuracy is the accuracy (Acc.) of the probing task.

[3]https://github.com/technion-cs-nlp/bias-probing

To evaluate the ID and OOD performance of models, we follow existing work (Schuster et al. 2019; He, Zha, and Wang 2019; Utama et al. 2021) and employ accuracy (Acc.) on the ID and corresponding OOD datasets.

4.4 Implementation Details

For the MNLI, SNLI and FEVER datasets, we train all models for 5 epochs; all models converge. The base model uses BERT-base (Devlin et al. 2019) and combines with cross-entropy to fine-tune on three datasets. For debiased models, the first step is to train a bias-only model, where we follow Sanh et al. (2021) using TinyBERT (Micheli, d'Hoffschmidt, and Fleuret 2020) for modeling unknown bias. Based on the same bias-only model, we train all the above debiased models. In the training process, we adopt the AdamW (Loshchilov and Hutter 2019) optimizer as the optimizer with initial learning rate $3 \cdot 10^{-5}$. Meanwhile, the temperature parameter τ, threshold λ, momentum coefficient m and scalar weighting hyperparameter α are set to 0.04, 0.6, 0.999, and 0.1. The sizes of the least similar positive samples S^p and the most similar negative samples S^{dn} are set to 150 and 1.

5 Experimental Results and Analysis

To answer our research questions we conduct bias extractability experiments, ID and OOD experiments, and ablation studies are conducted. To directly explore the effectiveness of DCT in reducing biased latent features, we conducted visualization experiments.

5.1 Bias Extractability

For RQ1, we analyze three types of bias and three datasets.

- **MNLI** – Table 1 shows results for the Overlap, Subsequence and Negwords probing tasks on MNLI. Compared to fine-tuned baseline (BERT-base), all debiasing methods except DCT increase the extractability of multiple types

Method	MNLI (Acc.)		SNLI (Acc.)		FEVER (Acc.)		
	dev	HANS	dev	Scramble	dev	Symm. v1	Symm. v2
BERT-base	84.16 ± 0.23	61.22 ± 1.17	90.61 ± 0.15	72.74 ± 6.87	87.06 ± 0.57	56.53 ± 0.78	63.84 ± 0.83
Reweight	82.56 ± 0.31	66.18 ± 1.04	86.44 ± 0.24	80.30 ± 6.99	83.45 ± 0.36	61.56 ± 1.19	67.33 ± 1.04
POE	81.62 ± 0.18	67.27 ± 1.21	83.69 ± 0.33	79.51 ± 6.42	82.23 ± 0.52	62.19 ± 1.65	67.36 ± 1.54
Conf-reg	84.15 ± 0.21	64.89 ± 1.08	90.56 ± 0.11	83.21 ± 4.26	85.31 ± 0.29	59.69 ± 1.35	64.75 ± 1.28
DCT	**84.19 ± 0.17**	**68.30 ± 0.85**	**90.64 ± 0.33**	**86.40 ± 4.64**	**87.12 ± 0.34**	**63.27 ± 1.62**	**68.45 ± 1.09**

Table 4: Classification accuracy on MNLI, SNLI and FEVER.

of biases, which is demonstrated by higher compression values and higher probing accuracy of biases. Compared to the baselines, DCT has the lowest compression value and probing accuracy for multiple biases, indicating that our method DCT reduces the extractability of multiple types of biased features simultaneously on MNLI.

- **SNLI** – Table 2 shows results for the Overlap, Subsequence and Negwords probing tasks on SNLI. Compared to the fine-tuned baseline (BERT-base), all debiasing methods except DCT increase the extractability of multiple biases. Compared to the baselines, DCT has the lowest compression value and bias probing accuracy for multiple biases, indicating that it reduces the extractability of multiple types of biased features simultaneously on SNLI.
- **FEVER** – Table 3 shows results for the Negwords probing task on FEVER. Compared to fine-tuned baseline (BERT-base), all debiasing methods except DCT increase the extractability of Negwords bias. Compared to the baselines, DCT has the lowest compression value and bias probing accuracy for Negwords bias, indicating that our method DCT reduces the extractability of Negwords bias on FEVER.

5.2 ID and OOD Performance

Next, we turn to RQ2 and evaluate the in-distribution and out-of-distribution performance of models on the development set and the corresponding challenge set of each dataset.

In-Distribution Performance. We evaluate the performance of models on the development sets of MNLI, SNLI and FEVER as the in-distribution (ID) performance. From the ID performance of models, we have the following observations: (i) Compared to debiasing baselines, our method DCT performs best on the ID development sets of MNLI, SNLI and FEVER. (ii) Compared to the fine-tuned BERT-base, Reweight and POE have substantially lower ID performance, Conf-reg can maintain ID performance on MNLI and SNLI, while DCT can maintain ID performance on MNLI, SNLI, and FEVER. (iii) Although Conf-reg also maintains the ID performance, it has the highest extractability of biased features. In contrast to Conf-reg, DCT has the lowest extractability of biased feature and maintains ID performance.

Out-of-Distribution Performance. We evaluate the performance of models on corresponding challenge sets of MNLI, SNLI and FEVER as the out-of-distribution (OOD) performance. We observe that: (i) Compared to the baselines, DCT performs well, both on the ID dataset and OOD dataset. For instance, on the MNLI, POE improves the average HANS accuracy from 61.22 to 67.27 but sacrifices 2.54 points of MNLI in-distribution accuracy; Conf-reg maintains in-distribution accuracy but only improves 3.67 points on HANS. (ii) Compared to the fine-tuned BERT-base, all debiased baselines improve the OOD performance, and Conf-reg even achieves a trade-off between ID and OOD performance, but all debiased baselines improve the extractability of biased features. In contrast, DCT reduces the extractability of biased features while improving ID and OOD performances, indicating that our method DCT reduces biased latent features and learns intended task-relevant features.

Method	MNLI (Acc.)		
	dev	HANS	avg.
DCT	84.19	**68.30**	**76.25**
-debiasing positive sampling	83.94	65.88	74.91
-dynamic negative sampling	83.54	65.79	74.67
-all	**84.57**	63.24	73.91

Table 5: Ablation studies with different strategies on MNLI.

5.3 Ablation Studies

For RQ3, we perform ablation experiments with respect to strategies, threshold λ, the number of debiasing positive samples, and the number of dynamic negative samples.

Impact of Different Strategies. Table 5 lists our ablation experiments on MNLI and HANS to explore the effectiveness of strategies. (i) -debiasing positive sampling: we built the DCT model without debiasing dataset and positive samples are randomly sampled from the original training dataset. (ii) -dynamic negative sampling: we built DCT without the bias-only model and negative samples are randomly sampled from the original training dataset. (iii) -all: we remove the debiasing positive sampling strategy and dynamic negative sampling strategy simultaneously. DCT is degraded to the original supervised contrastive learning (Gunel et al. 2021).

The results in Table 5 show that both strategies (debiasing positive sampling strategy and dynamic negative sampling strategy) enhance the OOD performances of DCT. The ID performance will be improved after removing all strategies compared to DCT. The reason is that supervised contrastive learning aims to improve the ID performance. However, the original supervised contrastive learning is not designed for solving dataset bias, so it has the lowest OOD performance.

Impact of Threshold λ. The threshold λ is defined in Eq. 2 to filter the debiased samples that are incorrectly predicted

	MNLI (Acc.)		
Threshold	dev	HANS	avg.
0.4	84.12	66.27	75.20
0.5	84.15	67.13	75.64
0.6	84.19	**68.30**	**76.25**
0.7	84.17	66.81	75.49
0.8	**84.25**	64.91	74.58

Table 6: Parameter analysis for threshold λ.

by the bias-only model but with a high confidence above the threshold. We conduct an ablation study on the threshold by changing it from 0.4 to 0.8. From Table 6, when λ is set to 0.6, we can get the best OOD performance. When the value of λ is too small, some of the filtered debias samples are not misclassified by containing biased features, thus introducing noise into the debiasing process. Conversely, when λ is too large, the number and diversity of filtered debias samples are insufficient, thus affecting the debiasing process.

	MNLI (Acc.)		
$\|S^p\|$	dev	HANS	avg.
75	83.96	66.94	75.45
100	**84.22**	67.15	75.69
150	84.19	**68.30**	**76.25**
500	84.18	66.14	75.16
1,000	84.12	66.82	75.47

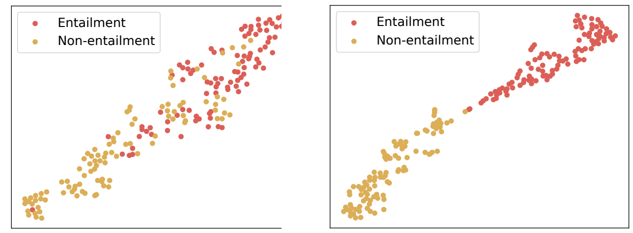
Table 7: Parameter analysis for the number of debiasing positive samples.

Impact of the Number of Debiasing Positive Samples. Several experiments are conducted to explore the impact of the number of debiasing positive samples. The results are shown in Table 7. More positive samples for DCT do indeed lead to better OOD performance. In addition, with the growth of the number of positive samples, OOD performance is slightly degraded which is probably due to some positive samples contain similar biased features to the anchor sample.

	MNLI (Acc.)		
$\|S^{dn}\|$	dev	HANS	avg.
0	83.54	65.79	74.67
1	**84.19**	**68.30**	**76.25**
5	83.66	68.25	75.96
10	84.07	68.06	76.06
20	84.11	68.15	76.13

Table 8: Parameter analysis for the number of dynamic negative samples.

Impact of the Number of Dynamic Negative Samples. To explore the impact of the number of dynamic negative samples, sufficient experiments are conducted. As shown in Table 8, the most similar biased negative sample improve the ID and OOD performances considerably, while adding more similar biased negative samples has little influence on the ID



(a) Encoder of BERT-base. (b) Encoder of DCT.

Figure 4: t-SNE plots of the learned [CLS] embeddings on 200 samples from the HANS dataset, comparing BERT-base fine-tuned with cross-entropy only (a) and with our proposed DCT (b) for the NLI task. Red: samples contain overlap bias and are labeled as entailment; Yellow: samples contain overlap bias and are labeled as non-entailment.

and OOD performance. We can observe that the best ID and OOD performances can be achieved when we set $|S^{dn}|$ to 1.

5.4 Visualizations

As mentioned before, the key idea of our proposed method DCT is to encourage positive examples with least similar bias to be closer and negative examples with most similar bias to be apart at the feature-level. To describe this insight intuitively, we use t-SNE to plot the [CLS] representations of the original BERT finetuned model and our debiased model using 200 data points sampled from the HANS dataset. The 100 data points sampled from HANS containing overlap bias are labeled as entailment and the other 100 data points containing overlap bias are labeled as non-entailment.

As shown in Fig. 4(a), the encoder trained with only cross-entropy is biased at the feature-level and thus cannot distinguish between samples with the same bias but different classes. In contrast, in Fig. 4(b), the encoder trained with DCT pushes away samples with the same overlap bias at the feature-level, so that samples with the same bias but different classes are more easier to distinguish.

6 Conclusions

We have focused on reducing biased latent features in an NLU model's representation and on capturing the dynamic influence of biased features. To tackle these challenges, we have proposed an NLU debiasing method, namely DCT. To mitigate biased latent features, we have proposed a debiasing, positive sampling strategy. To capture the dynamic influence of biased features, we have devised a dynamic negative sampling strategy to use the bias-only model to dynamically select the most similar biased negative sample during the training process. Experiments have shown that DCT improves the OOD performance while maintaining ID performance. In addition, our method reduces the extractability of multiple types of bias from an NLU model's representations. A limitation of DCT is that it is implemented only for the classification task. Our future work is to extend the proposed method to other NLU tasks that are impacted by dataset bias, e.g., named entity recognition and question answering.

Acknowledgments

This work was supported by the National Key R&D Program of China with grant No. 2020YFB1406704, the Natural Science Foundation of China (62272274, 62202271, 61902219, 61972234, 62072279, 62102234, 62106105), the Natural Science Foundation of Shandong Province (ZR2021QF129), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Agrawal, A.; Batra, D.; and Parikh, D. 2016. Analyzing the Behavior of Visual Question Answering Models. In *Proceedings of EMNLP*, 1955–1960.

Belinkov, Y.; Poliak, A.; Shieber, S. M.; Durme, B. V.; and Rush, A. M. 2019a. Don't Take the Premise for Granted: Mitigating Artifacts in Natural Language Inference. In *Proceedings of ACL*, 877–891.

Belinkov, Y.; Poliak, A.; Shieber, S. M.; Durme, B. V.; and Rush, A. M. 2019b. On Adversarial Removal of Hypothesis-only Bias in Natural Language Inference. In *Proceedings of SEM*, 256–262.

Bender, E. M.; and Koller, A. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of ACL*, 5185–5198.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of EMNLP*, 632–642.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of ICML*, 1597–1607.

Clark, C.; Yatskar, M.; and Zettlemoyer, L. 2019. Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases. In *Proceedings of EMNLP*, 4067–4080.

Dasgupta, I.; Guo, D.; Stuhlmüller, A.; Gershman, S.; and Goodman, N. D. 2018. Evaluating Compositionality in Sentence Embeddings. In *Proceedings of CogSci*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, 4171–4186.

Du, M.; He, F.; Zou, N.; Tao, D.; and Hu, X. 2022. Shortcut Learning of Large Language Models in Natural Language Understanding: A Survey. *CoRR*, abs/2208.11857.

Du, M.; Manjunatha, V.; Jain, R.; Deshpande, R.; Dernoncourt, F.; Gu, J.; Sun, T.; and Hu, X. 2021. Towards Interpreting and Mitigating Shortcut Learning Behavior of NLU models. In *Proceedings of NAACL*, 915–929.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of EMNLP*, 6894–6910.

Geirhos, R.; Jacobsen, J.; Michaelis, C.; Zemel, R. S.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut Learning in Deep Neural Networks. *Nat. Mach. Intell.*, 2(11): 665–673.

Ghaddar, A.; Langlais, P.; Rezagholizadeh, M.; and Rashid, A. 2021. End-to-End Self-Debiasing Framework for Robust NLU Training. In *Proceedings of ACL*, 1923–1929.

Giorgi, J. M.; Nitski, O.; Wang, B.; and Bader, G. D. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of ACL*, 879–895.

Glockner, M.; Shwartz, V.; and Goldberg, Y. 2018. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of ACL*, 650–655.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *Proceedings of ICLR*.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S. R.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of NAACL*, 107–112.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *Proceedings of CVPR*, 1735–1742.

He, H.; Zha, S.; and Wang, H. 2019. Unlearn Dataset Bias in Natural Language Inference by Fitting the Residual. In *Proceedings of EMNLP*, 132–142.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of CVPR*, 9726–9735.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of EMNLP*, 2021–2031.

Kaushik, D.; Hovy, E. H.; and Lipton, Z. C. 2020. Learning the Difference that Makes a Difference With Counterfactually-Augmented Data. In *Proceedings of ICLR*.

Kaushik, D.; and Lipton, Z. C. 2018. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. In *Proceedings of EMNLP*, 5010–5015.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Proceedings of NeurIPS*.

Li, L.; Song, D.; Ma, R.; Qiu, X.; and Huang, X. 2021. KNN-BERT: Fine-Tuning Pre-Trained Models with KNN Classifier. *arXiv preprint arXiv:2110.02523 (2021)*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*.

Mahabadi, R. K.; Belinkov, Y.; and Henderson, J. 2020. End-to-End Bias Mitigation by Modelling Biases in Corpora. In *Proceedings of ACL*, 8706–8716.

McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of ACL*, 3428–3448.

Mendelson, M.; and Belinkov, Y. 2021. Debiasing Methods in Natural Language Understanding Make Bias More Accessible. In *Proceedings of EMNLP*, 1545–1557.

Micheli, V.; d'Hoffschmidt, M.; and Fleuret, F. 2020. On the Importance of Pre-training Data Volume for Compact Language Models. In *Proceedings of EMNLP*, 7853–7858.

Min, J.; McCoy, R. T.; Das, D.; Pitler, E.; and Linzen, T. 2020. Syntactic Data Augmentation Increases Robustness to Inference Heuristics. In *Proceedings of ACL*, 2339–2352.

Naik, A.; Ravichander, A.; Sadeh, N. M.; Rosé, C. P.; and Neubig, G. 2018. Stress Test Evaluation for Natural Language Inference. In *Proceedings of COLING*, 2340–2353.

Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; and Kiela, D. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of ACL*, 4885–4901.

Ross, A.; Wu, T.; Peng, H.; Peters, M. E.; and Gardner, M. 2022. Tailor: Generating and Perturbing Text with Semantic Controls. In *Proceedings of ACL*, 3194–3213.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Commun. ACM*, 64(9): 99–106.

Sanh, V.; Wolf, T.; Belinkov, Y.; and Rush, A. M. 2021. Learning from Others' Mistakes: Avoiding Dataset Biases without Modeling Them. In *Proceedings of ICLR*.

Schuster, T.; Fisch, A.; and Barzilay, R. 2021. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. In *Proceedings of NAACL*, 624–643.

Schuster, T.; Shah, D. J.; Yeo, Y. J. S.; Filizzola, D.; Santus, E.; and Barzilay, R. 2019. Towards Debiasing Fact Verification Models. In *Proceedings of EMNLP*, 3417–3423.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of NAACL*, 809–819.

Utama, P. A.; Moosavi, N. S.; and Gurevych, I. 2020a. Mind the Trade-off: Debiasing NLU Models without Degrading the In-distribution Performance. In *Proceedings of ACL*, 8717–8729.

Utama, P. A.; Moosavi, N. S.; and Gurevych, I. 2020b. Towards Debiasing NLU Models from Unknown Biases. In *Proceedings of EMNLP*, 7597–7610.

Utama, P. A.; Moosavi, N. S.; Sanh, V.; and Gurevych, I. 2021. Avoiding Inference Heuristics in Few-shot Prompt-based Finetuning. In *Proceedings of EMNLP*, 9063–9074.

Voita, E.; and Titov, I. 2020. Information-Theoretic Probing with Minimum Description Length. In *Proceedings of EMNLP*, 183–196.

Wang, D.; Ding, N.; Li, P.; and Zheng, H. 2021. CLINE: Contrastive Learning with Semantic Negative Examples for Natural Language Understanding. In *Proceedings of ACL*, 2332–2342.

Williams, A.; Nangia, N.; and Bowman, S. R. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of NAACL*, 1112–1122.

Wu, Y.; Gardner, M.; Stenetorp, P.; and Dasigi, P. 2022. Generating Data to Mitigate Spurious Correlations in Natural Language Inference Datasets. In *Proceedings of ACL*, 2660–2676.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Proceedings of CVPR*, 3733–3742.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of ACL*, 4791–4800.

Zhang, Y.; Baldridge, J.; and He, L. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proceedings of NAACL*, 1298–1308.