Mind the Trade-off: Debiasing NLU Models without Degrading the In-distribution Performance

Prasetya Ajie Utama[†‡] **, Nafise Sadat Moosavi**[‡]**, Iryna Gurevych**[‡]

[†]Research Training Group AIPHES
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de

Abstract

Models for natural language understanding (NLU) tasks often rely on the idiosyncratic biases of the dataset, which make them brittle against test cases outside the training distribution. Recently, several proposed debiasing methods are shown to be very effective in improving out-of-distribution performance. However, their improvements come at the expense of performance drop when models are evaluated on the in-distribution data, which contain examples with higher diversity. This seemingly inevitable trade-off may not tell us much about the changes in the reasoning and understanding capabilities of the resulting models on broader types of examples beyond the small subset represented in the out-of-distribution data. In this paper, we address this trade-off by introducing a novel debiasing method, called *confidence regularization*, which discourage models from exploiting biases while enabling them to receive enough incentive to learn from all the training examples. We evaluate our method on three NLU tasks and show that, in contrast to its predecessors, it improves the performance on out-of-distribution datasets (e.g., 7pp gain on HANS dataset) while maintaining the original in-distribution accuracy.[1]

1 Introduction

Despite the impressive performance on many natural language understanding (NLU) benchmarks (Wang et al., 2018), recent pre-trained language models (LM) such as BERT (Devlin et al., 2019) are shown to rely heavily on idiosyncratic biases of datasets (McCoy et al., 2019b; Schuster et al., 2019; Zhang et al., 2019). These biases are commonly characterized as *surface features* of input examples that are strongly associated with the target labels, e.g., occurrences of negation words in natural language inference (NLI) datasets which are biased towards the *contradiction* label (Gururangan et al., 2018; Poliak et al., 2018). As a ramification of relying on biases, models break on the *out-of-distribution* data, in which such associative patterns between the surface features and the target labels are not present. This brittleness has, in turn, limited their practical applicability in some extrinsic use cases (Falke et al., 2019).

This problem has sparked interest among researchers in building models that are robust against *dataset biases*. Proposed methods in this direction build on previous works, which have largely explored the format of several prominent label-revealing biases on certain datasets (Belinkov et al., 2019). Two current prevailing methods, *product-of-expert* (He et al., 2019; Mahabadi and Henderson, 2019) and *learned-mixin* (Clark et al., 2019a) introduce several strategies to overcome the *known* biases by correcting the conditional distribution of the target labels given the presence of biased features. They achieve this by reducing the importance of examples that can be predicted correctly by using only biased features. As a result, models are forced to learn from harder examples in which utilizing solely superficial features is not sufficient to make correct predictions.

While these two state-of-the-art debiasing methods provide a remarkable improvement on the targeted out-of-distribution test sets, they do so at the cost of degrading the model's performance on the *in-distribution* setting, i.e., evaluation on the original test data which contains more diverse inference phenomena. It raises a question on whether these debiasing methods truly help in capturing a better notion of language understanding or simply biasing models to other directions. Ideally, if such an improvement is achieved for the right reasons (i.e., better reasoning capabilities by learning a more general feature representation), a debiased model

[1] The code is available at https://github.com/UKPLab/acl2020-confidence-regularization

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8717–8729
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

	product-of-expert	learned-mixin	**conf-reg (our)**
in-distribution	▼	▼	▲
out-of-distribution	▲	▲	▲
calibration	△	▼	▲
requires biased model	✔	✔	✔
requires hyperparameter	✖	✔	✖

Table 1: Comparison of our method against the state-of-the-art debiasing methods. Learned-mixin (Clark et al., 2019a) is a parameterized variant of Product-of-expert (He et al., 2019; Mahabadi and Henderson, 2019). Our novel confidence regularization method improves the out-of-distribution performance while optimally maintain the in-distribution accuracy.

should still be able to maintain its accuracy on previously unambiguous instances (i.e., instances that are predicted correctly by the baseline model), even when they contain biases.

In this work, we address this shortcoming by introducing a novel debiasing method that improves models' performance on the out-of-distribution examples while preserves the in-distribution accuracy. The method, called *confidence regularization*, draws a connection between the robustness against dataset biases and the overconfidence prediction problem in neural network models (Feng et al., 2018; Papernot et al., 2016). We show that by preventing models from being overconfident on biased examples, they are less likely to exploit the simple cues from these examples. The motivation of our proposed training objective is to *explicitly* encourage models to make predictions with lower *confidence* (i.e., assigning a lower probability to the predicted label) on examples that contain biased features.

Table 1 shows the comparison of our method with the existing state-of-the-art debiasing methods: *product-of-expert* and *learned-mixin*. We show that our method is highly effective in improving out-of-distribution performance while preserving the in-distribution accuracy. For example, our method achieves 7 points gain on an out-of-distribution NLI evaluation set, while slightly improves the in-distribution accuracy. Besides, we show that our method is able to improve models' calibration (Guo et al., 2017) so that the confidences of their predictions are more aligned with their accuracies. Overall, our contributions are the following:

- We present a novel *confidence regularization* method to prevent models from utilizing biased features in the dataset. We evaluate the advantage of our method over the state-of-the-art debiasing methods on three tasks, including natural language inference, fact verification, and paraphrase identification. Experimental results show that our method provides competitive out-of-distribution improvement while retaining the original in-distribution performance.

- We provide insights on how the debiasing methods behave across different datasets with varying degrees of biases and show that our method is more optimal when enough bias-free examples are available in the dataset.

2 Related Work

Biases in Datasets Researchers have recently studied more closely the success of large fine-tuned LMs in many NLU tasks and found that models are simply better in leveraging biased patterns instead of capturing a better notion of language understanding for the intended task (Bender and Koller, 2020). Models' performance often drops to a random baseline when evaluated on out-of-distribution datasets which are carefully designed to be void of the biases found in the training data. Using such targeted evaluation, McCoy et al. (2019b) observe that models trained on MNLI dataset (Williams et al., 2018) leverage syntactic patterns involving word overlap to blindly predict entailment. Similarly, Schuster et al. (2019) show that the predictions of fact verification models trained for the FEVER task (Thorne et al., 2018) are largely driven by the presence of indicative words in the input claim sentences.

Following similar observations across other tasks and domains, e.g., visual question-answering (Agrawal et al., 2016), paraphrase identification (Zhang et al., 2019), and argument reasoning comprehension (Niven and Kao, 2019), researchers proposed improved data collection techniques to reduce the artifacts that result in dataset biases. While these approaches are promising, only applying them without additional efforts in the modeling part may still deliver an unsatisfactory outcome. For instance, collecting new examples by asking human annotators to conform to specific rules may be costly and thus limit the scale and diversity of the resulting data (Kaushik et al., 2020). Recently proposed adversarial filtering methods (Zellers et al., 2019; Sakaguchi et al., 2019) are more cost effective but are not guaranteed to be artifacts-free. It is, therefore, crucial to develop learning methods that can overcome biases as a complement to the data collection efforts.

Debiasing Models There exist several methods that aim to improve models' robustness and generalization by leveraging the insights from previous work about the datasets' artifacts. In the NLI task, Belinkov et al. (2019) make use of the finding that partial input information from the hypothesis sentence is sufficient to achieve reasonable accuracy. They then remove this hypothesis-only bias from the input representation using an adversarial training technique. More recently, three concurrent works (Clark et al., 2019a; He et al., 2019; Mahabadi and Henderson, 2019) introduce a model-agnostic debiasing method for NLU tasks called `product-of-expert`. Clark et al. (2019a) also propose an adaptive variant of this method called `learned-mixin`. These two methods first identify examples that can be predicted correctly based only on biased features. This step is done by using a *biased model*[2], which is a weak classifier that is trained using only features that are known to be insufficient to perform the task but work well due to biases. The output of this pre-trained biased model is then used to adjust the loss function such that it down-weights the importance of examples that the biased model can solve. While this approach prevents models from learning the task mainly using biased features, it also reduces model's ability to learn from examples that can be solved using these features. As a result, models are unable to optimize accuracy on the original training distribution, and they possibly become biased in some other ways.

Similar to these methods, our method also uses a biased model to identify examples that exhibit biased features. However, instead of using it to diminish the training signal from these examples, we use it to scale the confidence of models' predictions. This enables the model to receive enough incentive to learn from all of the training examples.

Confidence Regularization Methods for regularizing the output distribution of neural network models have been used to improve generalization. Pereyra et al. (2017) propose to penalize the entropy of the output distribution for encouraging models to be less confident in their predictions. Previously, Szegedy et al. (2016) introduce a label smoothing mechanism to reduce overfitting by preventing the model from assigning a full probability to each training example. Our method regularizes models' confidence differently: we first perform an adaptive label smoothing for the training using knowledge distillation (Hinton et al., 2015), which, by itself, is known to improve the overall performance. However, our method involves an additional bias-weighted scaling mechanism within the distillation pipelines. As we will show, our proposed scaling mechanism is crucial in leveraging the knowledge distillation technique for the purpose of overcoming the targeted bias while maintaining high accuracy in the training distribution.

Similar to our work, Feng et al. (2018) propose a regularization method that encourages the model to be uncertain on specific examples. However, the objective and the methodology are different: they apply an entropy penalty term on examples that appear nonsensical to humans with the goal of improving models' interpretability. On the contrary, we apply our confidence regularization on every training example with a varying strength (i.e., higher uncertainty on more biased examples) to improve models' performance on the out-of-distribution data.

3 Method

Overview We consider the common formulation of NLU tasks as a multi-class classification problem. Given a dataset $\mathcal{D}$ that consists of n examples $(x_i, y_i)_{i\in[1,n]}$, with $x_i \in \mathcal{X}$ as a pair of sentences, and $y_i \in \{1, 2, ..., K\}$ where K is the number of classes. The goal is to learn a robust classifier $\mathcal{F}_m$, which computes the probability distribution over target labels, i.e., $\mathcal{F}_m(x_i) = p_i$.

The key idea of our method is to *explicitly* train $\mathcal{F}_m$ to compute *lower probability*, i.e., less confidence, on the predicted label when the input example exhibits a bias. This form of confidence regularization can be done by computing the loss function with the "soft" target labels that are obtained through our proposed smoothing mechanism. The use of soft targets as the training objective is motivated by the observation that the probability distribution of labels for each sample provides valuable information about the underlying task (Hinton et al., 2015; Pereyra et al., 2017). When the soft targets of certain examples have higher entropy, models can be explicitly taught that some labels are more likely to be correct than the others. Based on this intuition, we argue that adjusting the con-

[2] We follow the terminology used by He et al. (2019).

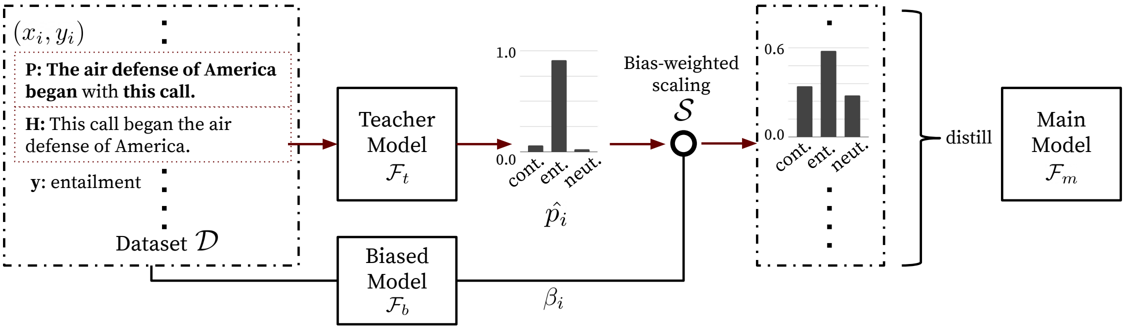

(x_i, y_i)
P: The air defense of America began with this call.
H: This call began the air defense of America.
y: entailment
Dataset $\mathcal{D}$
Teacher Model $\mathcal{F}_t$
Biased Model $\mathcal{F}_b$
1.0
0.0
cont.
ent.
neut.
$\hat{p_i}$
β_i
Bias-weighted scaling $\mathcal{S}$
0.6
0.0
cont.
ent.
neut.
distill
Main Model $\mathcal{F}_m$

Figure 1: An overview of our debiasing strategy when applied to the MNLI dataset. An input example that contains lexical-overlap bias is predicted as entailment by the teacher model with a high confidence. When biased model predicts this example well, the output distribution of the teacher will be re-scaled to indicate higher uncertainty (lower confidence). The re-scaled output distributions are then used to distill the main model.

fidence on soft labels can better inform the model about the true conditional distribution of the labels given the presence of the biased features.

We first produce a meaningful softened target distribution for each training example by performing *knowledge distillation* (Hinton et al., 2015). In this learning framework, a "teacher" model $\mathcal{F}_t$, which we parameterize identically to the main model $\mathcal{F}_m$, is trained on the dataset $\mathcal{D}$ using a standard classification loss. We then use $\mathcal{F}_t$ to compute output probability distribution $\hat{p_i}$, where $\mathcal{F}_t(x_i) = \hat{p_i}$. In the original knowledge distillation approach, the output of the teacher model $\hat{p_i}$ is then used to train $\mathcal{F}_m$. We extend this approach by adding a novel scaling procedure before we distill the teacher model into $\mathcal{F}_m$. We define a scaling function $\mathcal{S}$ that takes the probability distribution $\hat{p_i}$ and scale it such that the probability assigned to its predicted label is lowered when the example can be predicted well by only relying on the biased features.

Training the biased model For several NLU tasks, biased features are known a-priori, e.g., the word overlapping features in NLI datasets are highly correlated with the *entailment* label (McCoy et al., 2019b). We leverage this a-priori knowledge to design a measure of how well an example can be predicted given only the biased features. We refer to this measure as *bias weight*, denoted as β_i for every example x_i.

Similar to previous debiasing methods (Clark et al., 2019a), we compute bias weights using a *biased model*. This biased model, denoted as $\mathcal{F}_b$, predicts the probability distribution b_i, where $\mathcal{F}_b(x_i) = b_i = \langle b_{i,1}, b_{i,2}, ..., b_{i,K} \rangle$. We define the bias weight β_i as the scalar value of the assigned probability by $\mathcal{F}_b$ to the ground truth label: $\beta_i = b_{i,c}$ (c-th label is the ground truth).

Bias-weighted scaling As illustrated in Figure 1, our method involves scaling the teacher output $\hat{p_i}$ using β_i. We do this by defining a scaling function $\mathcal{S} : \mathbb{R}^K \rightarrow \mathbb{R}^K$:

$$\mathcal{S}(\hat{p_i}, \beta_i)_j = \frac{\hat{p_{i,j}}^{(1-\beta_i)}}{\sum_{k=1}^{K} \hat{p_{i,k}}^{(1-\beta_i)}}$$

for $j = 1, ..., K$. The value of β_i controls the strength of the scaling: as $\beta_i \rightarrow 1$, the scaled probability assigned to each label approaches $\frac{1}{K}$, which presents a minimum confidence. Conversely, when $\beta_i \rightarrow 0$, the teacher's probability distribution remains unchanged, i.e., $\mathcal{S}(\hat{p_i}, 0) = \hat{p_i}$.

Training the main model The final step is to train $\mathcal{F}_m$ by distilling from the scaled teacher model's outputs. Since the main model is parameterized identically to the teacher model, we refer to this step as self-distillation (Furlanello et al., 2018). Self-distillation is performed by training $\mathcal{F}_m$ on pairs of input and the obtained soft target labels $(x_i, \mathcal{S}(\hat{p_i}, \beta_i))$. Specifically, $\mathcal{F}_m$ is learned by minimizing a standard cross-entropy loss between the scaled teacher's output $\mathcal{S}(\hat{p_i}, \beta_i)$ and the current prediction of the main model:

$$\mathcal{L}(x_i, \mathcal{S}(\hat{p_i}, \beta_i)) = -\mathcal{S}(\hat{p_i}, \beta_i) \cdot \log \mathcal{F}_m(x_i)$$

In practice, each $\mathcal{S}(\hat{p_i}, \beta_i)$ is computed only once as a preprocessing step. Our method *does not require hyperparameters*, which can be an advantage since most out-of-distribution datasets do not provide a development set for tuning the hyperparameters.

4 Experimental Setup

In this section, we describe the datasets, models, and training details used in our experiments.

4.1 Natural Language Inference

We use the MNLI dataset (Williams et al., 2018) for training. The dataset consists of pairs of premise and hypothesis sentences along with their inference labels (i.e., entailment, neutral, and contradiction). MNLI has two in-distribution development and test sets, one that matches domains of the training data (MNLI-m), and one with mismatching domains (MNLI-mm). We consider two out-of-distribution datasets for NLI: HANS (Heuristic Analysis for NLI Systems) (McCoy et al., 2019b) and MNLI-hard test sets (Gururangan et al., 2018).

HANS The dataset is constructed based on the finding that the word overlapping between premise and hypothesis in NLI datasets is strongly correlated with the *entailment* label. HANS consists of examples in which such correlation does not exist, i.e., hypotheses are *not entailed* by their word-overlapping premises. HANS is split into three test cases: (a) Lexical overlap (e.g., "*The doctor was paid by the actor*" $\nRightarrow$ "*The doctor paid the actor*"), (b) Subsequence (e.g., "*The doctor near the actor danced*" $\nRightarrow$ "*The actor danced*"), and (c) Constituent (e.g., "*If the artist slept, the actor ran*" $\nRightarrow$ "*The artist slept*"). Each category contains both entailment and non-entailment examples.

MNLI-hard Hypothesis sentences in NLI datasets often contain words that are highly indicative of target labels (Gururangan et al., 2018; Poliak et al., 2018). It allows a simple model that predicts based on the hypothesis-only input to perform much better than the random baseline. Gururangan et al. (2018) presents a "hard" split of the MNLI test sets, in which examples cannot be predicted correctly by the simple hypothesis-only model.

4.2 Fact Verification

For this task, we use the training dataset provided by the FEVER challenge (Thorne et al., 2018). The task concerns about assessing the validity of a claim sentence in the context of a given evidence sentence, which can be labeled as either *support*, *refutes*, and *not enough information*. We use the Fever-Symmetric dataset (Schuster et al., 2019) for the out-of-distribution evaluation.

Fever-Symmetric Schuster et al. (2019) introduce this dataset to demonstrate that FEVER models mostly rely on the claim-only bias, i.e., the occurrence of words and phrases in the claim that are biased toward certain labels. The dataset is manually constructed such that relying on cues of the claim can lead to incorrect predictions. We evaluate the models on the two versions (version 1 and 2) of their test sets.[3]

4.3 Paraphrase Identification

We use the Quora Question Pairs (QQP) dataset for training. QQP consists of pairs of questions which are labeled as *duplicate* if they are paraphrased, and *non-duplicate* otherwise. We evaluate the out-of-distribution performance of QQP models on the QQP subset of PAWS (Paraphrase Adversaries from Word Scrambling) (Zhang et al., 2019).

PAWS The QQP subset of PAWS consists of question pairs that are highly overlapping in words. The majority of these question pairs are labeled as non-duplicate. Models trained on QQP are shown to perform worse than the random baseline on this dataset. This partly indicates that models largely rely on lexical-overlap features to perform well on QQP. We report models' performance on the duplicate and non-duplicate examples separately.

4.4 Models

Baseline Model We apply all of the debiasing methods across our experiments on the BERT base model (Devlin et al., 2019), which has shown impressive in-distribution performance on the three tasks. In our method, BERT base is used for both $\mathcal{F}_t$ and $\mathcal{F}_m$. We follow the standard setup for sentence pair classification tasks, in which the two sentences are concatenated into a single input and the special token `[CLF]` is used for classification.

Biased Model ($\mathcal{F}_b$) We consider the biased features of each of the examined out-of-distribution datasets to train the biased models. For HANS and PAWS, we use hand-crafted features that indicate how words are shared between the two input sentences. Following Clark et al. (2019a), these features include the percentage of hypothesis words that also occur in the premise and the average of cosine distances between word embedding in the premise and hypothesis.[4] We then train a simple

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

[4] We include the detailed description in the appendix.

Method	MNLI-m		MNLI-mm		HANS				Hard subset	
	dev	test	dev	test	lex.	subseq.	const.	*avg.*	MNLI-m	MNLI-mm
BERT-base	84.3 ± 0.3	84.6	84.7 ± 0.1	83.3	72.4	52.7	57.9	61.1 ± 1.1	76.8	75.9
Learned-mixin $_{\text{hans}}$	84.0 ± 0.2	84.3	84.4 ± 0.3	83.3	**77.5**	54.1	63.2	64.9 ± 2.4	-	-
Product-of-expert $_{\text{hans}}$	82.8 ± 0.2	83.0	83.1 ± 0.3	82.1	72.9	65.3	**69.6**	**69.2** ± 2.6	-	-
Regularized-conf $_{\textbf{hans}}$	84.3 ± 0.1	**84.7**	84.8 ± 0.2	83.4	73.3	**66.5**	67.2	**69.1** ± 1.2	-	-
Learned-mixin $_{\text{hypo}}$	80.5 ± 0.4	79.5	81.2 ± 0.4	80.4	-	-	-	-	79.2	78.2
Product-of-expert $_{\text{hypo}}$	83.5 ± 0.4	82.8	83.8 ± 0.2	84.1	-	-	-	-	**79.8**	**78.7**
Regularized-conf $_{\textbf{hypo}}$	**84.6** ± 0.2	84.1	**85.0** ± 0.2	**84.2**	-	-	-	-	78.3	77.3

Table 2: The in-distribution accuracy (in percentage point) of the NLI models along with their accuracy on out-of-distribution test sets: HANS and MNLI hard subsets. Models are only evaluated against their targeted out-of-distribution dataset.

nonlinear classifier using these features. We refer to this biased model as the *hans* model.

For MNLI-hard and Fever-Symmetric, we train a biased model on only hypothesis sentences and claim sentences for MNLI and FEVER, respectively. The biased model is a nonlinear classifier trained on top of the vector representation of the input sentence. We obtain this vector representation by max-pooling word embeddings into a single vector for FEVER, and by learning an LSTM-based sentence encoder for MNLI.

State-of-the-art Debiasing Models We compare our method against existing state-of-the-art debiasing methods: *product-of-expert* (He et al., 2019; Mahabadi and Henderson, 2019) and its variant *learned-mixin* (Clark et al., 2019a). *product-of-expert* ensembles the prediction of the main model (p_i) with the prediction of the biased model (b_i) using $p'_i = softmax(\log p_i + \log b_i)$, where p'_i is the ensembled output distribution. This ensembling enables the main model to focus on learning from examples that are not predicted well by the biased model. *Learned-mixin* improves this method by parameterizing the ensembling operation to let the model learn when to incorporate or ignore the output of the biased model for the ensembled prediction.

On FEVER, we also compare our method against the *example-reweighting* method by Schuster et al. (2019). They compute the importance weight of each example based on the correlation of the n-grams within the claim sentences with the target labels. These weights are then used to compute the loss of each training batch.

Training Details As observed by McCoy et al. (2019a), models can show high variance in their out-of-distribution performance. Therefore, we run each experiment five times and report both average and standard deviation of the scores.[5] We also use training configurations that are known to work well for each task.[6] For each experiment, we train our *confidence regularization* method as well as *product-of-expert* and *learned-mixin* using the same biased-model. Since the challenge datasets often do not provide a development set, we could not tune the hyperparameter of learned-mixin. We, therefore, use their default weight for the entropy penalty term.[7]

5 Results

The results for the tasks of NLI, fact verification, and paraphrase identification are reported in Table 2, Table 3, and Table 4, respectively.

5.1 In-distribution Performance

The results on the original development and test sets of each task represent the in-distribution performance. Since we examine two types of biases in NLI, we have two debiased NLI models, i.e., *Regularized-conf*$_{\text{hans}}$ and *Regularized-conf*$_{\text{hypo}}$ which are trained for debiasing HANS and hypothesis-only biases, respectively.

We make the following observations from the results: (1) Our method outperforms *product-of-expert* and *learned-mixin* when evaluated on the corresponding in-distribution data of all the three tasks; (2) *Product-of-expert* and *learned-mixin* drop the original BERT baseline accuracy on most

[5] Due to the limited number of possible submissions, we report the MNLI test scores only from a model that holds the median out-of-distribution performance.

[6] We set a learning rate of $5e^{-5}$ for MNLI and $2e^{-5}$ for FEVER and QQP.

[7] E.g., $w = 0.03$ for training on MNLI.

Method	FEVER $_{\text{dev}}$	Symm. $_{\text{v1}}$	Symm. $_{\text{v2}}$
BERT-base	85.8 ± 0.1	57.9 ± 1.1	64.4 ± 0.6
Learned-mixin $_{\text{claim}}$	83.1 ± 0.7	60.4 ± 2.4	64.9 ± 1.6
Product-of-expert $_{\text{claim}}$	83.3 ± 0.3	61.7 ± 1.5	65.5 ± 0.7
Reweighting $_{\text{bigrams}}$	85.5 ± 0.3	**61.7** ± 1.1	**66.5** ± 1.3
Regularized-conf $_{\text{claim}}$	**86.4** ± 0.2	60.5 ± 0.4	66.2 ± 0.6

Table 3: Accuracy on the FEVER dataset and the corresponding challenge datasets.

Method	QQP dev		PAWS test	
	dupl	¬dupl	dupl	¬dupl
BERT-base	88.4 ± 0.3	92.5 ± 0.3	96.9 ± 0.3	9.8 ± 0.4
LMixin $_{\text{hans}}$	77.5 ± 0.7	91.9 ± 0.2	69.7 ± 4.3	**51.7** ± 4.3
Prod-exp $_{\text{hans}}$	80.8 ± 0.2	**93.5** ± 0.1	71.0 ± 2.3	49.9 ± 2.3
Reg-conf $_{\text{hans}}$	**85.0** ± 0.7	91.5 ± 0.4	**91.0** ± 1.8	19.8 ± 1.3

Table 4: Results of the evaluation on the QQP task.

of the in-distribution experiments; (3) Regardless of the type of bias, our method preserves the in-distribution performance. However, it is not the case for the other two methods, e.g., *learned-mixin* only results in a mild decrease in the accuracy when it is debiased for HANS, but suffers from substantial drop when it is used to address the hypothesis-only bias; (4) Our method results in a slight in-distribution improvement in some cases, e.g., on FEVER, it gains 0.6pp over BERT baseline. The models produced by *Regularized-conf* $_{\text{hans}}$ also gain 0.1 points to both MNLI-m and MNLI-mm test sets; (5) All methods, including ours decrease the in-distribution performance on QQP, particularly on its duplicate examples subset. We will discuss this performance drop in Section 6.

5.2 Out-of-distribution Performance

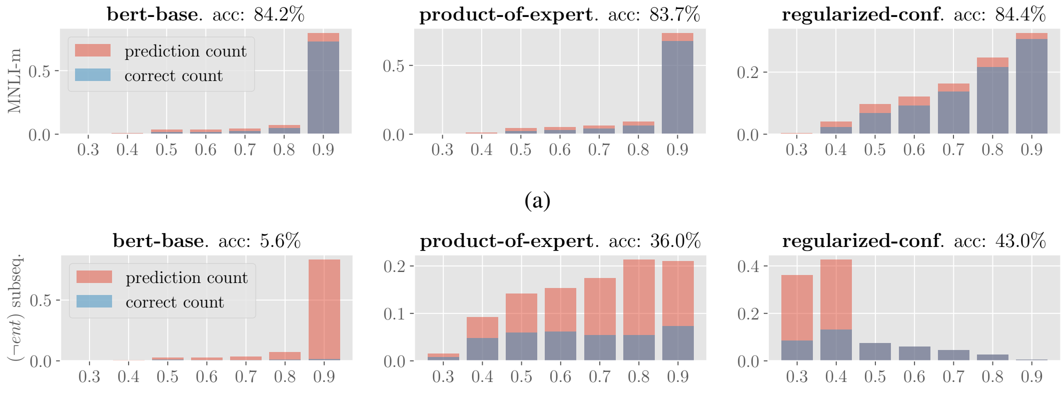
The rightmost columns of each table report the evaluation results on the out-of-distribution datasets for each task. Based on our out-of-distribution evaluations, we observe that: (1) Our method minimizes the trade-off between the in-distribution and out-of-distribution performance compared to the other methods. For example, on HANS, *learned-mixin* maintains the in-distribution performance but only improves the average HANS accuracy from 61.1% to 64.9%. *product-of-expert* gains 7 points improvement over the BERT baseline while reducing the MNLI-m test accuracy by 1.6 points. On the other hand, our method achieves the competitive 7 points gain without dropping the in-distribution performance; (2) The performance trade-off is stronger on some datasets. On PAWS, the two compared methods improve the accuracy on the *non-duplicate* subset while reducing models' ability to detect the *duplicate* examples. Our method, on the other hand, finds a balance point, in which the non-duplicate accuracy can no longer be improved without reducing the duplicate accuracy; (3) depending on the use of hyperparameters, *learned-mixin* can make a lower out-of-distribution improvement compared to ours, even after substantially degrading in-distribution performance, e.g., on FEVER-symmetric$_{\text{v2}}$, it only gains 0.5 points while dropping 3 points on the FEVER development set.

6 Discussions and Analysis

Ablation studies In this section, we show that the resulting improvements from our method come from the combination of both self-distillation and our scaling mechanism. We perform ablation studies to examine the impact of each of the components including (1) *self-distillation*: we train a model using the standard self-distillation without bias-weighted scaling, and (2) *example-reweighting*: we train a model with the standard cross-entropy loss with an example reweighting method to adjust the importance of individual examples to the loss. The weight of each example is obtained from the (scaled) probability that is assigned by the teacher model to the ground truth label.[8] The aim of the second setting is to exclude the effect of self-distillation while keeping the effect of our scaling mechanism.

Table 5 presents the results of these experiments on MNLI and HANS. We observe that each component individually still gains substantial improvements on HANS over the baseline, albeit not as strong as the full method. The results from the *self-distillation* suggest that the improvement from our method partly comes from the regularization effect of the distillation objective (Clark et al., 2019b; Furlanello et al., 2018). In the *example-reweighting* experiment, we exclude the effect of all the scaled teacher's output except for the probability assigned to the ground truth label. Compared to *self-distillation*, the proposed *example-reweighting* has a higher impact on improving the performance in both in-distribution and out-of-distribution eval-

[8] Details of the ablation experiments are included in the supplementary materials.



(a)

(b)

Figure 2: Distribution of models' confidence on their predicted labels. The blue areas indicate the fraction of each bin that are correct. (a) Distribution on MNLI-m dev by models trained using hypothesis-only biased model. (b) Distribution on non-entailment subsequence subset of HANS by models trained using *hans* biased-model.

Method	MNLI	HANS
BERT-base	84.3	61.1
Full method	84.3	69.1
self-distillation	84.6	64.4
example-reweighting	84.7	65.3

Table 5: Results of the ablation experiments. The MNLI column refers to the MNLI-m dev set.

	BERT-baseline	product-of-expert	learned-mixin	**conf-reg (our)**
MNLI-m	9.0	7.7	9.9	**5.4**
MNLI-mm	8.5	7.6	9.5	**5.6**

Table 6: The calibration scores of models measured by ECE (lower is better).

uations. However, both components are necessary for the overall improvements.

In-distribution performance drop of product-of-expert The difference between our method with *product-of-expert* and its variants is the use of biased examples during training. *Product-of-expert* in practice scales down the gradients on the biased training examples to allow the model to focus on learning from the harder examples (He et al., 2019). As a result, models often receive little to no incentive to solve these examples throughout the training, which can effectively reduce the training data size. Our further examination on a *product-of-expert* model (trained on MNLI for HANS) shows that its degradation of in-distribution performance largely comes from the aforementioned examples. Ensembling back the *biased-model* to the main model can indeed bring the in-distribution accuracy back to the BERT baseline. However, this also leads to the original poor performance on HANS, which is counterproductive to the goal of improving the out-of-distribution generalization.

Impact on Models' Calibration We expect the training objective used in our method to discourage models from making overconfident predictions, i.e., assigning high probability to the predicted labels even when they are incorrect. We investigate the changes in models' behavior in terms of their confidence using the measure of *calibration*, which quantifies how aligned the confidence of the predicted labels with their actual accuracy are (Guo et al., 2017). We compute the *expected calibration error* (ECE) (Naeini et al., 2015) as a scalar summary statistic of calibration. Results in Table 6 show that our method improves model's calibration on MNLI-m and MNLI-mm dev sets, with the reduction of ECE ranging from 3.0 to 3.6. The histograms in figure 2 show the distribution of models' confidences in their predictions. Figure 2a demonstrates that the prediction confidences of our resulting model on MNLI-m are more smoothly distributed. In figure 2b, we observe that our debiased model predicts examples that contain lexical overlap features with lower confidence, and when the confidence is higher, the prediction is more likely to be correct.

Impact of biased examples ratio To investigate the slight in-distribution drop by our method in QQP (Table 4), we examine the ratio of biased examples in the QQP training data by evaluating the

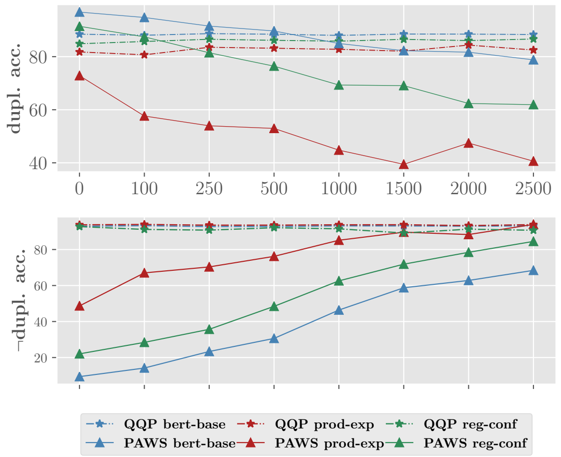


Figure 3: Results on the PAWS-augmented QQP dataset.

performance of the biased model on the dataset. We find that almost 80% of the training examples can be solved using the lexical overlap features alone, which indicates a severe lexical overlap bias in QQP.[9] Moreover, in 53% of all examples, the biased model makes correct predictions with a very high confidence ($\beta_i > 0.8$). For comparison, the same biased model predicts only 12% of the MNLI examples with confidence above 0.8 (more comparisons are shown in the supplementary material. As a result, there are not enough unbiased examples in QQP and the resulting soft target labels in this dataset are mostly close to a uniform distribution, which in turn may provide insufficient training signal to maximize the accuracy on the training distribution.

Impact of adding bias-free examples Finally, we investigate how changing the ratio of biased examples affects the behavior of debiasing methods. To this end, we split PAWS data into training and test sets. The training set consists of 2500 examples, and we use the remaining 10K examples as a test set. We train the model on QQP that is gradually augmented with fractions of this PAWS training split and evaluate on a constant PAWS test set. Figure 3 shows the results of this experiment. When more PAWS examples are added to the training data, the accuracy of the BERT baseline gradually improves on the non-duplicate subset while its accuracy slowly drops on the duplicate subset. We observe that *product-of-expert* exaggerates this effect: it reduces the duplicate accuracy up to 40% to obtain the 93% non-duplicate accuracy. We note that our method is the most effective when the entire 2500 PAWS examples are included in the training, obtaining the overall accuracy of 77.05% compared to the 71.63% from the baseline BERT.

7 Conclusion

Existing debiasing methods improve the performance of NLU models on out-of-distribution datasets. However, this improvement comes at the cost of strongly diminishing the training signal from a subset of the original dataset, which in turn reduces the in-distribution accuracy. In this paper, we address this issue by introducing a novel method that regularizes models' confidence on biased examples. This method allows models to still learn from all training examples without exploiting the biases. Our experiments on four out-of-distribution datasets across three NLU tasks show that our method provides a competitive out-of-distribution performance while preserves the original accuracy.

Our debiasing framework is general and can be extended to other task setups where the biases leveraged by models are correctly identified. Several challenges in this direction of research may include extending the debiasing methods to overcome multiple biases at once or to automatically identify the format of those biases which simulate a setting where the prior knowledge is unavailable.

Acknowledgments

We thank Leonardo Ribeiro and Max Glockner for the thoughtful discussion on the earlier version of this work and the anonymous reviewers for their constructive comments. We also thank Tal Schuster for the support in using the Fever-Symmetric dataset. This work is supported by the German Research Foundation through the research training group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES, GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessian State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question an-

[9] The random baseline is 50% for QQP.

swering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander M. Rush. 2019. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, pages 256–262. Association for Computational Linguistics.

Emily Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page *to appear*, virtual conference. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019a. 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 (EMNLP-IJCNLP)*, pages 4067–4080, Hong Kong, China. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019b. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel 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: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 132–142. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Rabeeh Karimi Mahabadi and James Henderson. 2019. Simple but effective techniques to reduce biases. *CoRR*, abs/1909.06321.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. 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, Florence, Italy. Association for Computational Linguistics.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597. IEEE Computer Society.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WINOGRANDE: an adversarial winograd schema challenge at scale. *CoRR*, abs/1907.10641.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, 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 (EMNLP-IJCNLP)*, pages 3417–3423, Hong Kong, China. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Ablation Details

For the second setting of our ablation studies, we perform an example reweighting using the scaled probability of the teacher model $\mathcal{F}_t$ on the ground truth label. Specifically, the cross entropy loss assigned to each batch of size m is computed by the following:

$$-\sum_{s=1}^{b} \frac{\hat{p_{s,c}}}{\sum_{u=1}^{b} \hat{p_{u,c}}} \cdot \log(p_{s,c})$$

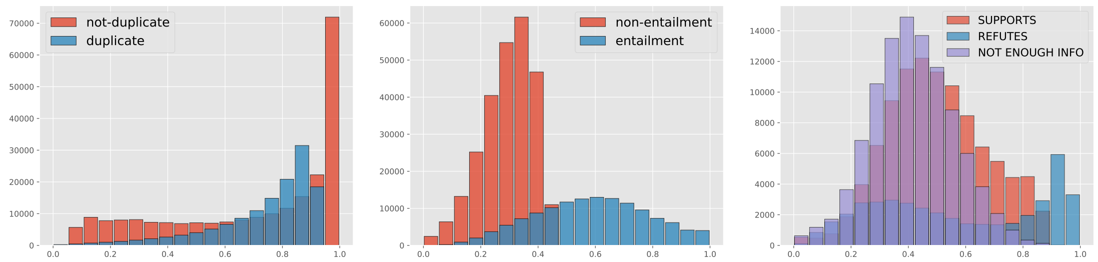
where we assume that cth label is the ground truth label. The probability assigned to the correct label by the teacher model is then denoted as $\hat{p_{s,c}}$. The currect predicted probability of the main model is denoted as $p_{s,c}$.

B Bias Weights Distribution

Figure 4 shows the performance of biased models on QQP, MNLI, and FEVER. For QQP and MNLI we show the results of biased model trained using lexical overlap features. For FEVER, the biased model is trained with claim-only partial input. We show that on PAWS (figure 4a), a large portion of examples can be predicted with a very high confidence by the biased model.

C HANS Biased Model

We use the hand-crafted HANS-based features proposed by Clark et al. (2019a). These features include: (1) whether all words in the hypothesis exist in the premise; (2) whether the hypothesis is a contiguous subsequence of the premise; (3) the fraction of hypothesis words that exist in the premise; (4) the average and the max of cosine distances between word vectors in the premise and the hypothesis.



(a)

(b)

(c)

Figure 4: The distribution of biased model confidence on three training datasets of QQP, MNLI, and FEVER.