LEARNING FROM OTHERS: SIMILARITY-BASED REGULARIZATION FOR MITIGATING ARTIFACTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Common methods for mitigating spurious correlations in natural language understanding (NLU) usually operate in the output space, encouraging a main model to behave differently from a bias model by down-weighing examples where the bias model is confident. While improving out of distribution (OOD) performance, it was recently observed that the internal representations of the presumably debiased models are actually more, rather than less biased. We propose SimgReg, a new method for debiasing internal model components via similarity-based regularization, in representation space: We encourage the model to learn representations that are either similar to an unbiased model or different from a biased model. We experiment with three NLU tasks and different kinds of biases. We find that SimReg improves OOD performance, with little in-distribution degradation. Moreover, the representations learned by SimReg are less biased than in other methods.

1 INTRODUCTION

Recent studies (McCoy et al., 2019; Geirhos et al., 2020, *inter alia*) show that in many cases neural models tend to exploit spurious correlations (a.k.a dataset biases, artifacts) in datasets and learn shortcut solutions rather than the intended function. For example, in MNLI—a popular Natural Language Understanding dataset—there is a high correlation between negation words such as “not, don’t” and the contradiction label (Gururangan et al., 2018). Thus models trained on MNLI confidently predict contradiction whenever there is a negation word in the input without considering the whole meaning of the sentence. As a result of relying on such shortcuts, models fail to generalize and perform poorly when tested on out-of-distribution data (OOD) in which such associative patterns are not present (McCoy et al., 2019) — these models are commonly known as ‘biased’ models —. Moreover, this behavior limits their practical applicability in cases where the real-world data distribution differs from the training distribution.

Recent efforts to mitigate learning spurious correlations (a.k.a debiasing methods) downweigh the importance of training samples that contain such correlations, effectively performing data reweighting Schuster et al. (2019); Utama et al. (2020a); Sanh et al. (2021); Cadene et al. (2019). Typically, a bias-only model is trained and its confidence is used to reweigh training samples. One might expect that such an *extrinsic* debiasing would lead to “suppressing the model from capturing non-robust features” Du et al. (2022). However, Mendelson & Belinkov (2021) showed a counter-intuitive trend: the more extrinsically de-biased a model is, the more biased are its representations. i.e., higher accuracy of such models on OOD challenge sets is correlated with increase of intrinsic bias. [1] Such superficial debiasing is problematic as the bias may reappear when the model is used in another setting (Orgad et al., 2022).

Inspired by this finding, we propose to perform *intrinsic* debiasing on the internal model components. We develop SimReg, a new debiasing method based on similarity-regularization, where we encourage the internal representations to be either (i) similar to representations of an *unbiased model* [2]; or (ii) dissimilar from representations of a biased model. We further apply the (dis)similarity regularization on either the model representations or its gradients.

[1] we measure representation-bias (intrinsic bias) by the *easiness* of identifying the spurious correlations in the representations (more details in Sec 6.1)

[2] In our case, unbiased model is a model that does not rely on the spurious correlations in its decision making.

Our regularization framework allows us to encourage certain constraints on how the data should be represented (representation regularization), or how the model should be sensitive to the representations of the data (gradient regularization). This is different from previous methods, where the main model usually learns to avoid the errors of bias models. Using our approach allows us to transfer knowledge from other models regarding "good" representation of the data and encourage avoiding "biased" representations.

We evaluate our approach on three tasks—natural language inference, fact checking, and paraphrase identification—and multiple spurious correlations attested in the literature: lexical overlap, partial inputs, and unknown biases from weak models (see Section 2.1). We demonstrate that our approach improves performance on out-of-distribution (OOD) challenge sets, while incurring little degradation in in-distribution (ID) performance. Finally, by measuring bias extractability, we find that SimReg representations are less biased than those obtained with competing debiasing methods.

2 Related Work

A growing body of work has revealed that models tend to exploit spurious correlations found in their training data (Geirhos et al., 2020). Spurious correlations are correlations between certain features of the input and certain labels, which are not causal. Models tend to fail when tested on out of distribution data, where said correlations do not hold. We briefly mention several relevant cases and refer to Du et al. (2022) for a recent overview of shortcut learning and its mitigation in natural language understanding.

2.1 Dataset bias

Partial-input bias. A common spurious correlation in sentence-pair classification tasks, like natural language inference (NLI), is **partial-input bias** – the association between words in one of the sentences and certain labels. For example, negation words are correlated with a 'contradiction' label when present in the hypothesis in NLI datasets (Gururangan et al., 2018; Poliak et al., 2018) and with a 'refutes' label when present in the claim in fact verification datasets (Schuster et al., 2019). A common approach for revealing the presence of such spurious correlations is to train a partial-input baseline (Feng et al., 2019). When such a model performs well despite having access only to a part of the input, it indicates that that part has spurious correlations.

Lexical overlap bias. Another common bias is when certain labels are associated with **lexical overlap** between the two input sentences. McCoy et al. (2019) found that high lexical-overlap between the premise and hypothesis correlates with 'entailment' in NLI datasets. As a result, NLI models fail when evaluated on HANS, a challenge set where that correlation does not hold. Similarly, Zhang et al. (2019) found that models trained on a paraphrase identification dataset fail to predict 'non-duplicate' questions that have high lexical-overlap.

Unknown biases. Identifying the preceding biases assumes prior knowledge of the type of bias existing in the dataset. A few studies have used weak learners to identify **unknown biases** (Sanh et al., 2021; Utama et al., 2020b). When limiting either the model capacity or its training data, it tends to exploit simple patterns.

2.2 Debiasing methods

Spurious correlation mitigation can be performed on different levels: Data-based mitigation, where the data is augmented with samples that do not align with the bias found in the dataset (Wang & Culotta, 2021; Kaushik et al., 2020, inter alia). Model/training-based mitigation, where the either the model or the training procedure is modified. A common strategy in this approach is to train a *bias model*, which latches on the bias in the dataset, and use its outputs to train the final, debiased, *main model*. He et al. (2019) and Clark et al. (2019) used variants of product-of-experts (PoE) to combine the outputs of the biased and main model during training to encourage the main model to "ignore" biased samples. Utama et al. (2020a) proposed *confidence regularization*, where they perform self-distillation with re-weighted teacher outputs using bias-weighted scaling, i.e., they induce the model to be less confident on biased samples. These methods can be viewed as data re-weighting methods, similar to Liu et al. (2021), who proposed to up-weigh examples that are miss-classified by the

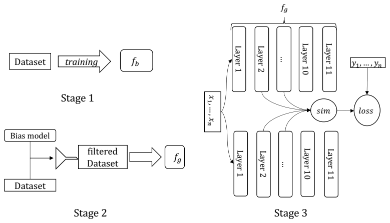

Figure 1: Illustration of SimReg: (1) train a bias model f_b; (2) use its predictions to filter the training set and train a target model f_g; (3) train a main model while regularizing its representations to be similar to f_g.

biased model, i.e., hard examples. All these methods work in the output space, while we work in representation space.

Most relevant to our work, Bahng et al. (2020) debias vision models by learning representations that are statistically independent from those of a biased model, by minimizing a statistical independence measure (HSIC) in a min-max optimization objective. We propose a simpler objective function, based on similarity regulariation, which can easily be trained by SGD. Additionally, while they focus only on learning representations independent of a biased model, we propose learning representations that are either dissimilar from biased models or similar to unbiased ones.

2.3 Knowledge Distillation

Our approach have some similarity with Knowledge-Distillation (KD) methods, where we transfer knowledge from teacher model to a student model (usually to a smaller model to 'compress' the teacher model). In our framework we utilize this transfer of knowledge to improve robustness of a model. Aguilar et al. (2020) perform KD using internal representations, by minimizing the cosine similarity between the representations of the two models. They compare the similarity of the classification token *(CLS)* whereas we compare all of the tokens. Additionally we use second-order isomorphism methods to compare the models whereas they perform first-order methods.

3 Methodology

The key idea behind our approach is to encourage our model to be either similar to an unbiased model or dissimilar from a biased model. To achieve this, we design a three-stage procedure (Figure 1):

1. We train a bias model, f_b, on the original training set, $\mathcal{D}$. This model is meant to capture dataset biases, as explained in Section 3.1. In the case of decreasing similarity, we use f_b as our target model, f_g, and continue directly to Stage 3.
2. In order to obtain an unbiased model, we filter the training set based on the predictions of f_b and train a target model f_g on the unbiased part of the training set, $\mathcal{D}^{\mathcal{U}}$ (Section 3.2).
3. We train the main model on $\mathcal{D}$ while encouraging its representations to be (dis)similar to those of f_g (Section 3.3).

3.1 Training a biased model

To mitigate a specific bias, we use a bias-specific model, f_b, which is designed to capture the intended bias. To mitigate lexical-overlap bias, we use the model proposed in (Clark et al., 2019): an MLP whose input features are the ratio of overlap between the two parts of the input, and the average of the minimum cosine similarity between the embeddings of each word from the two sentences. For partial-input bias, we use a partial-input model (Gururangan et al., 2018; Belinkov et al., 2019), i.e., we train a model only on the hypothesis/claim part of the input for MNLI/FEVER partial-input biases, respectively. To mitigate unknown biases, we use TinyBert (Turc et al., 2020) as a our bias model (f_b);Sanh et al. (2021) showed that limited capacity models recover previously-known biases in the dataset without explicitly modeling them.

In the case of decreasing dissimilarity from a biased model, we use this f_b as the target model, i.e., $f_g = f_b$, and proceed to Stage 3 (Section 3.3). In the case of increasing similarity to an unbiased model, we cannot use f_b as we need an unbiased model; the next section describes how to obtain it.

3.2 Obtaining an unbiased model

To obtain an unbiased model, we run f_b on the training set, $\mathcal{D}$, and exclude samples on which f_b is correct and confident. The remaining samples compose our unbiased dataset, $\mathcal{D}^{\mathcal{U}}$:

$$\mathcal{D}^{\mathcal{B}} = \{x_i | x_i \in \mathcal{D} \wedge f_b(x) == y_i \ \wedge \ c(f_b(x_i)) > c_t\} \tag{1}$$

$$\mathcal{D}^{\mathcal{U}} = \mathcal{D} \setminus \mathcal{D}^{\mathcal{B}} \tag{2}$$

where c_t is a confidence threshold and $c(\cdot)$ is the models' confidence, i.e., the highest probability it assigns among the predictions. Our unbiased model, f_g, is obtained by training a new model on $\mathcal{D}^{\mathcal{U}}$.

Choosing the threshold c_t is performed manually by plotting the confidence of the bias model over the training set. When there is a significant bias signal in the dataset, we see a spike in the number of biased samples. Figure 2 shows an example for claim-only bias in FEVER.

A natural question is the following: *What is the advantage of our framework if we already have an unbiased model?* We emphasize that the unbiased model was trained on $\mathcal{D}^{\mathcal{U}}$, a subset of $\mathcal{D}$, and argue that other samples in $\mathcal{D}$ could also be useful. Indeed, we show experimentally that training a model on the full training set while regularizing it to be similar to the unbiased model leads to a better IID–OOD tradeoff.

3.3 Training the main model

The final step is to train the main model, f_m. We propose two approaches. The first is to encourage the model during training to learn different representations than a biased model, by penalizing its similarity to said biased model, f_b (in this case, $f_g = f_b$). Thus the model would learn different decision boundaries than the biased model. The second approach is to increase the similarity of the learned representations to an unbiased model, f_g. Thus, our model will encode the data in an unbiased manner and its predictions will be less biased.

In both cases, we need to compute the the similarity between the representations of the main model and those of the target model, f_g. Directly comparing the representations of the models on a single example is not possible, since each model might learn a different latent space for representing the data. Furthermore, the two models might have different architectures and dimensionalities. For instance, in some of our experiments we compare BERT-base (768 dimensions) with TinyBERT (128 dimensions) or with an MLP of 7 dimensions. To overcome these challenges, we use second-order similarity measures, which operate at the batch level (Section 3.3.1).

Formally, we add a similarity regularization term to the batch training loss to promote the similarity/dis-similarity. Given a batch $\mathcal{B}$, we minimize the following objective:

$$\mathcal{L} = \sum_{i \in \mathcal{B}} \mathcal{L}_{CE}(f_m(x_i), y_i) + \lambda \cdot sim(Z, H) \tag{3}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, λ is a trade-off hyper-parameter, Z and H are respectively the main and target model representations of the batch, $f(x)$ is the prediction of the model on input x, and sim is a similarity measure. To increase the similarity, we use $\lambda < 0$. Unless otherwise noted, we set $\lambda = 10$ (-10) for decreasing (increasing) similarity.

Since we wish the main model, f_m, to resemble or differ from f_g only on biased samples, we apply regularization only on the biased subset, $\mathcal{D}^{\mathcal{B}}$: We stochastically sample a batch either from $\mathcal{D}^{\mathcal{U}}$ and optimize regular cross-entropy, or from $\mathcal{D}^{\mathcal{B}}$ minimizing the objective in Eq. 3. Appendix A.4) shows that regularizing only $\mathcal{D}^{\mathcal{B}}$ results in better OOD performance, supporting our intuition.

Finally, we examine two techniques to increase similarity/dis-similarity of the main and target models. The first works directly at the representations level, i.e., we regularize the similarity of the representations (activations). The second approach is indirect: We regularize the similarity of the *gradients* of the main and target models, inducing similar (different) changes to model weights, which should make the model more similar (different) to an unbiased (biased) model. In this case, we replace the second term in Eq. 3 with $sim(\nabla Z, \nabla H)$.

3.3.1 Choice of similarity

Instead of directly comparing two vectors that originate from different models, we use second-order isomorphism methods. In particular, as in representational similarity analysis (RSA) Kriegeskorte et al. (2008), we compute a representation similarity matrix (RSM) for each model, which is a matrix where each entry is the similarity between two representations in the batch: $RSM(Z) = \hat{Z}^T\hat{Z}$, where $\hat{Z}$ is column-wise L_2 normalization. Then, we compare two RSMs via their correlation:

$$CosCor(Z, H) = |\frac{(RSM(Z) - \overline{RSM(Z)}) \cdot (RSM(H) - \overline{RSM(H)})}{|\overline{RSM(Z)}|_f |\overline{RSM(H)}|_f}| \tag{4}$$

where $\overline{Z}$ is the mean of Z entries and $|Z|_f$ is the Frobenius norm.

As an alternative, we consider a popular RSA-based measure called centered kernel alignment (CKA; Kornblith et al. 2019). CKA is computed in a similar manner to CosCor, replacing the cosine distance with dot-product in the RSM computation and using a different centring transformation.

Experimentally, we found that $CosCor$ works better, so we report its results in the main paper, and compare with CKA in Appendix A.2.

4 Experimental Setup

4.1 Datasets

4.1.1 Natural Language Inference

We train models on MNLI, a popular NLI dataset consisting of $\sim 400k$ English examples in multiple genres (Williams et al., 2018). Each example is a pair of premise and hypothesis sentences, and the task is to predict whether the hypothesis is entailed, contradicted, or neutral w.r.t the premise. MNLI contains several spurious correlations as discussed in Section 2, such as lexical overlap and hypothesis-only biases. We train on the MNLI training set and report IID results on dev-matched.

As OOD test set, we use HANS (McCoy et al., 2019) for evaluation against **lexical overlap bias**. HANS is constructed using structured templates that obey bias heuristics, e.g., the hypothesis overlaps with premise, but with half of the examples having non-entailment labels, as opposed to the bias in MNLI. For **hypothesis-only bias** we use MNLI-hard, a subset of MNLI's dev-mismatched set where a hypothesis-only model failed to classify correctly (Gururangan et al., 2018).

4.1.2 Synthetic MNLI

As a sanity test, we introduce synthetic spurious correlations to MNLI (Synthetic-MNLI), following prior work (He et al., 2019; Sanh et al., 2021; Dranker et al., 2021). We prepend the input with a 'label-token' that correlates highly with the label. We used tokens <*0*>, <*1*>, and <*2*>, corresponding to *entailment*, *neutral*, and *contradiction*. Following (Dranker et al., 2021), we denote the probability of injecting a token to the input as the $prevalence$ of the bias, and the probability of the prepended token being correct as the $strength$ of the bias. Through all our experiments, we used prevalence of 1.0 and strength of 0.95. The subsets of examples containing bias token with wrong and right correlations are denoted *anti_bias* and *bias* subsets, respectively.

The goal of this setting is to demonstrate the viability of the proposed approaches. Thus we use an *oracle unbiased* model as f_g for the case of increasing similarity, i.e., a model trained on regular MNLI (without synthetic bias). For the bias model, f_b, we train a model for a small enough number of steps to capture the bias, judging by the rapid drop of training loss; we found 1k steps sufficient.

4.1.3 Fact Verification

Fact Extraction and VERification (FEVER) (Thorne et al., 2018) is a dataset for fact verification against textual sources. Given evidence and claim sentences, the task is to predict the relation between them: SUPPORTED, REFUTED, or NOT ENOUGH INFO. We train on the FEVER training set and evaluate IID on the development set.

We use FEVER-Symmetric Schuster et al. (2019) for OOD evaluation against **claim-only bias**. The construction of FEVER-Symmetric ensures that there is no correlation between partial input and labels, thus it enables us to evaluate the extent of debiasing on this type of bias.

4.1.4 QQP

Quora Question Pairs (QQP) is a collection of >400K question pairs from the Quora platform. Given a pair of questions, the task is to predict whether they are *duplicate* (paraphrase) or *non-duplicate*. QQP is biased in that question pairs with low **lexical-overlap** between them are correlated with the *non-duplicate* label. We train on the QQP training set and evaluate IID on the development set.

Paraphrase Adversaries from Word Scrambling (PAWS) Zhang et al. (2019) is a dataset for paraphrase identification that is built in a adversarial manner to lexical-overlap bias. The authors scramble the words of a sentence to generate samples with high lexical-overlap that are not a paraphrase. We use the QQP subset of PAWS as our OOD evaluation set for **lexical-overlap bias**. For IID, we report F1-score of 'duplicate' label, and for OOD we report F1 score of 'non-duplicate' label.

4.2 Models

We evaluate our approach using BERT (Devlin et al., 2018) as the main and target model. We repeat some of the experiments using DeBERTa (He et al., 2021) to verify that our method is not specific to BERT. For full training details see Appendix A.1.

5 Results

5.1 Synthetic bias

Table 1: Results on Synthetic-MNLI.

Model	Biased	Anti-biased
BERT-base	**98.5 ± 0.1**	41.8 ± 1.1
Oracle	83.8	82.1
SimReg ↑	96.7 ± 0.1	61.00 ± 0.9
SimReg ↓	97.0 ± 0.0	49.0 ± 2.4
∇ SimReg ↑	82.8 ± 0.0	**72.1 ± 0.0**
∇ SimReg ↓	94.7 ± 0.4	57.1 ± 2.4

The results on Synthetic-MNLI are in Table 1. All of the SimReg approaches resulted in a large increase compared to the baseline on the *anti-biased* subset, where the synthetic token is mis-aligned with the label. Increasing similarity (↑) performed better than decreasing it (↓). The improvement comes at a cost of a small decrease on the biased subset, with the exception of increasing gradient similarity (∇ SimReg ↑), where the drop is large. Compared to an oracle model, which was trained without the synthetic bias, the regularized models perform worse, indicating that they were not able to completely discard the bias.

5.2 Known bias

Tables 2 and 3 show the results on known bias cases. Focusing first on partial-input bias (Table 2), all our SimReg models outperform the baseline on the OOD sets (MNLI hard and FEVER Symmetric), except for ∇ SimReg ↓, which is on par. Increasing similarity (↑) seems to work better than decreasing similarity (↓). The improvements are comparable to or better than those of competitive approaches (POE and ConfReg).

Turning to lexical-overlap bias (Table 3), we see a similar pattern: SimReg performs much better than the baseline on HANS non-entailed and PAWS (the OOD sets), with little or no degradation on the corresponding IID dev sets. In this bias type, increasing similarity works much better than decreasing it. Compared to competitive methods, ConfReg performs a bit better on HANS non-entailed. However, SimReg ↑ is much better on average on HANS.

A telling comparison is between SimReg and the guidance model, which is a model that was trained only on unbiased examples (f_g, Section 3.2). In almost all cases, when we increase similarity to this model (rows with ↑), we get models that perform better than it, on both IID and OOD sets. Interestingly, in QQP f_g performed poorly on PAWS, but using it as a guidance model still improved

Table 2: Known bias: Mitigating partial-input bias.

	MNLI		FEVER	
	dev	hard	dev	Symmetric
BERT-base	83.9 ± 0.1	76.9 ± 0.2	**85.4 ± 0.1**	58.2 ± 0.6
Guidance	79.2	78.0	70.6	59.7
POE	84.1 ± 0.1	78.02 ± 0.4	81.44 ± 0.1	59.2 ± 0.1
ConfReg	84.3 ± 0.1	**78.4 ± 0.6**	85.2 ± 0.3	61.0 ± 1.7
SimReg ↑	**84.4 ± 0.2**	78.2 ± 0.1	84.5 ± 0.2	**61.2 ± 0.4**
SimReg ↓	83.0 ± 0.1	77.9 ± 0.5	84.1 ± 0.9	60.3 ± 1.1
∇ SimReg ↑	83.1 ± 0.1	78.2 ± 0.1	80.9 ± 0.1	60.1 ± 0.6
∇ SimReg ↓	81.7 ± 0.2	77.8 ± 0.2	79.9 ± 0.6	58.3 ± 0.1

Table 3: Known bias: Mitigating lexical-overlap bias.

	MNLI	HANS			QQP	
	dev	non-entailment	entailment	avg	dev	PAWS
BERT-base	83.9 ±0.1	30.1 ±1.3	**98.1** ±0.7	64.1	**88.4** ±0.1	28.2 ±2.3
Guidance	83.0	24.3	89.2	56.7	87.9	29.3
PoE	83.9 ±0.4	41.8 ±5.3	93.3 ±3.4	67.6	75.4 ±0.3	**73.6*** ±1.7
ConfReg	**84.3** ±0.2	60.9 ±6.6	72.3 ±8.5	66.6	85.4 ±0.5	28.7 ±3.6
SimReg ↑	83.6 ±0.1	57.6 ±6.0	82.3 ±2.6	**70.0**	86.9 ±0.0	36.9 ±1.2
SimReg ↓	83.9 ±0.1	40.1 ±1.7	89.4 ±3.6	64.8	87.7 ±0.3	32.6 ±1.2
∇ SimReg ↑	83.4 ±0.0	57.9 ±1.4	73.0 ±2.6	65.5	87.3 ±0.1	33.2 ±1.7
∇ SimReg ↓	84.0 ±0.1	48.8 ±6.4	89.8 ±5.5	69.3	87.5 ±0.2	31.6 ±3.5

performance. These results support our hypothesis that increasing similarity to an unbiased model can lead to better representations than those of the unbiased model itself.

5.3 Unknown bias

The results of unknown bias mitigation are in Table 4. Debiasing from unknown bias seems to be more challenging than known bias mitigation, with smaller improvements of all debiasing methods compared to the baseline. This is consistent with prior work on unknown biases (Sanh et al., 2021; Utama et al., 2020b). On MNLI hard, there is practically no improvement, but in other cases we find improvements of up to 6–7 points (on HANS and PAWS).

Comparing the different SimReg models, increasing similarity (↑) works better than decreasing it (↓), and regularizing representations is generally better than regularizing gradients, except for PAWS, where increasing gradient similarity works very well. When compared to competitive approaches, SimReg works better on HANS and PAWS, but ConfReg is better on MNLI hard and FEVER Symmetric.

5.4 Results with Stronger Models

In this section we investigate whether our approach improves the performance of stronger models than BERT. While most work tends to compare with BERT as the baseline, it is important to demonstrate that a new debiasing method is effective also when applied to stronger models.[3] We experiment with DeBERTa-V1 (He et al., 2021) and DeBERTa-V3 (Federer et al., 2019). As Table 5 shows, SimReg still leads to improvements above the strong DeBERTa-V3. Notably, ConfReg does

[3] Bowman (2022) made such a claim about *analyzing* stronger models; we believe it is similarly important to work on *robustifying* stronger models.

Table 4: Results for unknown bias mitigation.

	MNLI		HANS	FEVER		QQP	
	dev	hard	avg	dev	Symm.	dev	PAWS
BERT	**83.9** ±0.1	76.9±0.2	64.1	85.4±0.1	58.2±0.6	**88.4 ±0.1**	28.2 ±2.2
Biased	64.6±0.1	51.4±0.2	50.0	71.3±0.1	45.3±0.4	80.6 ±0.1	30.4 ±0.2
Guidance	80.9	74.9	61.9	64.9	50.9	73.13	55.4
POE	83.1±0.1	75.8±0.5	65.4	84.2±0.1	57.7±1.7	87.6 ±0.4	34.7 ±6.1
ConfReg	83.4±0.4	**77.0** ±0.5	63.2	**86.0** ±0.2	**60.0** ±1.6	86.6 ±0.3	16.6 ±3.0
BERT + $\mathcal{F}_{bow}$	83.0 ±0.4	76.6	70.4	87.1 ±0.2	61.0 ±1.4	-	-
SimReg↑	82.9±0.1	76.1±0.1	**68.2**	85.6±0.3	59.1±0.2	84.4 ±0.1	40.6 ±0.6
SimReg↓	82.9±0.3	75.8±0.6	63.5	84.1±0.5	57.5±2.2	86.6 ±0.7	32.4 ±4.6
∇SimReg↑	83.4±0.0	75.9±0.3	64.8	81.1±0.0	59.1±0.1	82.2 ±0.3	45.5 ±1.5
∇SimReg↓	82.9±0.3	75.3±0.4	62.7	83.8±1.1	57.2±1.5	84.0 ±2.0	36.4 ±13.8

Table 5: Results with DeBERTa-V3.

	MNLI		HANS	QQP	
	dev	hard	non-ent	dev	PAWS
baseline	89.9 ±0.1	85.2 ±0.1	56.7 ±2.2	89.9 ±0.1	55.7 ±5.6
ConfReg	90.1 ±0.1	86.0 ±0.1	54.5 ±2.0	88.8 ±0.1	61.1 ±2.0
SimReg ↑	89.1 ±0.2	85.1 ±0.3	66.8 ±0.7	86.3 ±0.2	67.0 ±2.0
∇SimReg ↑	87.9 ±1.6	83.9 ±1.1	74.8 ±1.3	89.3 ±0.1	58.2 ±1.5

not improve the baseline in this case, while SimReg does. Similar improvements are obtained with DeBERTa-V1 (Table 13, App. A.8).

6 Analysis

6.1 Probing for Bias Extractability

Recall that Mendelson & Belinkov (2021) found that, counter-intuitively, extrinsic debiasing methods, like POE and ConfReg, increase the bias in models' internal representations. Their finding motivated us to propose SimReg, which regularizes the representations themselves. Their discovery was made by measuring the *extractability* of bias from model representations, for which they used compression, a measure from minimum description length probing (Voita & Titov, 2020). We perform a similar analysis of our models. Figure 2 shows the increase in compression of debiased models compared to the baseline, against the increase of performance on OOD challenge set HANS (non-entailed). Consistent with Mendelson & Belinkov (2021), ConfReg leads to a large performance increase on HANS, but suffers from a large increase in bias extractability. In contrast, SimReg is able to improve performance on OOD challenge sets with little increase in bias extractability, supporting our motivation to perform debiasing at the representation level.

6.2 Bias recovery

We demonstrate the importance and effectiveness of debiasing the representations of a model. We retrained the classification layer of the resulted models after debiasing. In Table 6 we present the results on lexical-overlap bias; in App.A.6 we present other configurations. In all approaches there was a drop on HANS non-entailment accuracy. In SimReg↑ we got the highest performance on the OOD challenge set. This indicates that the representations produced by SimReg↑ have the least signal of lexical-overlap spurious correlation.

Table 6: lexical-overlap bias recovery

	IID	HANS -
POE	83.4 ±0.2	38.2 ±4.2
ConfReg	**84.8** ±0.1	20.5 ±6.2
BERT + $\mathcal{F}_{bow}$	83.7 ±0.2	42.2 ±1.2
SimReg↑	84.3 ±0.3	**44.2 ±4.4**

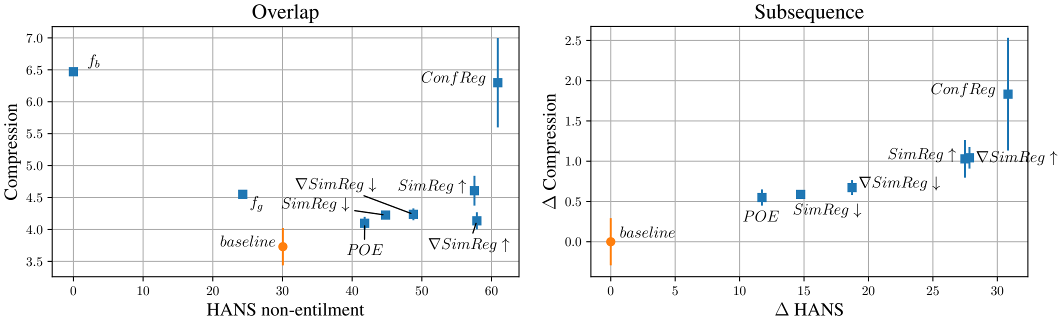

Figure 2: Compression increment relative to $HANS$ non-entailment subset. Left: Overlap bias. Right: Subsequence bias. SimReg leads to lower increase in intrinsic bias than ConfReg.

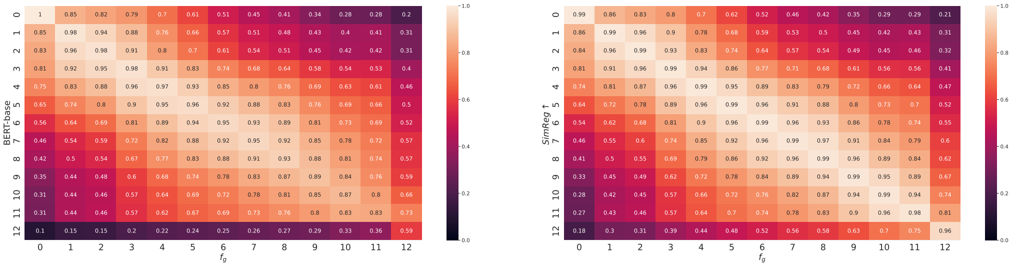

Figure 3: Similarity of an unbiased model, f_g, to either a baseline (Left) or a SimReg model (Right). Similarity regularization makes top layers more similar to the unbiased model, as desired.

6.3 Similarity Heat-map Analysis

To investigate whether our similarity-based regularization achieves its goal, we compute the similarity between every layer in the main model and every layer in the (unbiased) guidance model, and likewise the similarity between layers of the baseline model and layers of the guidance model. We expect our similarity regularization to increase the similarity of the main model to the guidance model, compared to that of the baseline model.

Figure 3 (Left) shows that, without similarity-based regularization, the bottom layers of the baseline and guidance model are already similar, but the top layers are rather different. This is consistent with findings on how fine-tuning affects mostly the top layers (Mosbach et al., 2020; Merchant et al., 2020), as both models started from a pre-trained BERT. Figure 3 (Right) shows that after our similarity-based regularization, the top layers of the main and guidance models become very similar, as desired. Moreover, the regularization also indirectly affects lower layers (bottom row of the heatmap). We conclude that the similarity regularization is successful and affects large parts of the model even when applied only on a few layers.

6.4 Ablations

Our main reported results were with CosCor as the similarity method (Section 3.3.1). In Appendix A.2 we report experiments with CKA as an alternative measure, showing it also leads to consistent improvements compared to the baseline, albeit not as good as CosCor. Additionally, in the main experiments we reported results when regularizing only on batched from a biased subset, $\mathcal{D}^{\mathcal{B}}$. In Appendix A.4, we compare with results when regularizing all samples, finding that regularizing only on the biased samples is better.

7 Conclusion

In this work, we have proposed SimReg, a new debiasing approach using similarity-based regularization. We have explored several variants of this approach, regularizing by increasing or decreasing similarity of a model to biased or unbiased models, respectively, at the level of either the representations or the gradients. We found SimReg to improve performance on OOD challenge sets on multiple bias types and NLU tasks, with little decrease in IID performance. Future work may investigate the effect of simultaneously learning from unbiased and biased models. Another interesting direction is to extend our approach to generation tasks, which would require different similarity measures.

Ethics Statement

Our work develops a new approach to mitigate spurious correlations in NLU tasks. These are also known as dataset biases, but are different from social biases such as gender or racial bias. One could use our approach to debias against social biases. However, a malicious actor could use our basic approach to increase such social bias, rather than decrease it, by reversing the optimization. In such a case, probing for bias extractability as we performed in this work may expose biased representations.

Reproducibility Statement

We provide detailed configurations in Appendix A.1. Code in PyTorch (Paszke et al., 2019) for reproducing our results is available at `https://github.com/simreg/SimReg`. Gradient regularization experiments require twice the memory of representation regularization. For gradients experiments we used an NVIDIA A40 GPU; a typical training time was $\sim$ 7 hours. For representations experiments, we used an NVIDIA A4000 RTX GPU, and training time was $\sim$ 6 hours.

References

URL https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs.

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7350–7357, Apr. 2020. doi: 10.1609/aaai.v34i05.6229. URL https://ojs.aaai.org/index.php/AAAI/article/view/6229.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning debiased representations with biased representations, 2020. URL https://openreview.net/forum?id=SJeHwJSYvH.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 877–891, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1084. URL https://aclanthology.org/P19-1084.

Samuel Bowman. The dangers of underclaiming: Reasons for caution when reporting how NLP systems fail. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7484–7499, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.516. URL https://aclanthology.org/2022.acl-long.516.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in Neural Information Processing Systems*, 32: 841–852, 2019.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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)*, pp. 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1418. URL https://aclanthology.org/D19-1418.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. URL https://arxiv.org/abs/1810.04805.

Yana Dranker, He He, and Yonatan Belinkov. IRM—when it works and when it doesn't: A test case of natural language inference. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=KtvHbjCF4v.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding: A survey. *arXiv preprint arXiv:2208.11857*, 2022.

Callie Federer, Haoyan Xu, Alona Fyshe, and Joel Zylberberg. Training neural networks to have brain-like representations improves object recognition performance. *CoRR*, abs/1905.10679, 2019. URL http://arxiv.org/abs/1905.10679.

Shi Feng, Eric Wallace, and Jordan Boyd-Graber. Misleading failures of partial-input baselines. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5533–5538, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1554. URL https://aclanthology.org/P19-1554.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

He He, Sheng Zha, and Haohan Wang. 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 2019)*, pp. 132–142, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6115. URL https://aclanthology.org/D19-6115.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *2021 International Conference on Learning Representations*, May 2021. URL https://www.microsoft.com/en-us/research/publication/deberta-decoding-enhanced-bert-with-disentangled-attention-2/. Under review.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008. URL https://www.frontiersin.org/articles/10.3389/neuro.06.004.2008.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

Michael Mendelson and Yonatan Belinkov. Debiasing methods in natural language understanding make bias more accessible. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1545–1557, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.116. URL https://aclanthology.org/2021.emnlp-main.116.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 33–44, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.4. URL https://aclanthology.org/2020.blackboxnlp-1.4.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 68–82, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.7. URL https://aclanthology.org/2020.blackboxnlp-1.7.

Hadas Orgad, Seraphina Goldfarb-Tarrant, and Yonatan Belinkov. How gender debiasing affects internal model representations, and why it matters. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2602–2628, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.188. URL https://aclanthology.org/2022.naacl-main.188.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL https://aclanthology.org/S18-2023.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Hf3qXoiNkR.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3419–3425, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1341. URL https://aclanthology.org/D19-1341.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models, 2020. URL https://openreview.net/forum?id=BJg7x1HFvB.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8717–8729, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.770. URL https://aclanthology.org/2020.acl-main.770.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7597–7610, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.613. URL `https://aclanthology.org/2020.emnlp-main.613`.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL `https://aclanthology.org/2020.emnlp-main.14`.

Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031, May 2021. doi: 10.1609/aaai.v35i16.17651. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17651`.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Yuan Zhang, Jason Baldridge, and Luheng He. 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)*, pp. 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL `https://aclanthology.org/N19-1131`.

A Appendix

A.1 Training parameters

We used pre-trained *bert-base-uncased* from HuggingFace models (Wolf et al., 2019), trained for 10 epochs with a batch size of 64 and a learning rate of $2\mathrm{e}{-5}$ that warms up on the first 10% of the training steps and decays linearly. The results reported in the tables are the mean and standard deviation of 3 different random seeds (changes the initialization of the model and the training batches).

We performed the similarity regularization on multiple layers, aggregating the similarities by summing it across the layers. When increasing similarity f_g is the same architecture as f_m. Experimentally we found that upper layers are most effective for this case. We regularize layers $10, 11,$ and 12, each layer in f_g paired with its parallel layer in f_m. For gradient regularization, we choose layers from across the model: embedding layer and layers 1, 7, 11, and 12, and use this set of layers for increasing and decreasing gradient similarity whenever f_g has the same architecture as f_m (partial-input biases for $\downarrow$).

For decreasing similarity from models with a different architecture, using a combination of layers yielded the best results, and we used the upper two layers and the embeddings layers. For gradient dis-similarity, regularizing the gradient of lower layers was the major factor as validated in Appendix A.3, which provides additional results when regularizing different layers.

A.2 Results with CKA

In Table 7 we repeat experiments from the main paper, but using CKA as the similarity measure instead of CosCor. This approach also leads to improvements compared to the baseline, but not as large as using CosCor.

Table 7: Results of SimReg with CKA as the similarity measure.

	MNLI		HANS	
	dev	hard	non-entailment	entailment
Unknown bias				
SimReg $\uparrow$	84.1 ± 0.2	77.1 ± 0.4	35.5 ± 1.8	96.4 ± 0.2
SimReg $\downarrow$	82.4 ± 0.3	75.0 ± 0.3	43.7 ± 3.5	79.0 ± 2.0
∇SimReg $\uparrow$	76.5 ± 0.1	70.7 ± 0.3	87.9 ± 0.5	25.5 ± 1.6
∇SimReg $\downarrow$	81.8 ± 0.5	75.2 ± 0.7	38.8 ± 3.1	84.4 ± 3.0
Lexical bias				
SimReg $\uparrow$	83.6 ± 0.1	–	57.5 ± 6.0	82.3 ± 2.7
SimReg $\downarrow$	83.9 ± 0.1	–	36.7 ± 5.8	96.2 ± 1.4
∇SimReg $\uparrow$	82.9 ± 0.2	–	59.0 ± 4.5	68.7 ± 5.5
∇SimReg $\downarrow$	83.8 ± 0.1	–	41.8 ± 4.8	93.6 ± 1.1
Hypothesis-only				
SimReg $\uparrow$	82.4 ± 0.2	78.8 ± 0.2	–	–
SimReg $\downarrow$	82.3 ± 0.1	76.3 ± 0.7	–	–
∇SimReg $\uparrow$	77.0 ± 0.1	79.1 ± 0.2	–	–
∇SimReg $\downarrow$	81.6 ± 0.3	77.9 ± 0.7	–	–

A.3 Layers

In the main experiments, we regularized multiple layers together, as described in Appendix A.1. In Tables 8 and 9 we present the results when regularizing different layers, where we regularize only one layer at a time. We see several patterns. In the case of increasing representation similarity, deeper layers work better. In the case of regularizing gradients, early layers work better. In decreasing representation similarity, individual layers are not effective, as opposed to regularizing multiple layers as in the main experiments. To simplify the choice of layers, we reported results with a set of layers as described in Appendix A.1.

Table 8: Layers search.

	SimReg↑		∇SimReg ↑	
	bias	anti-bias	bias	anti-bias
Embeddings	98.7 ±0.1	39.3 ±0.9	96.1 ±0.2	63.8 ±0.7
Layer-2	98.2 ±0.0	45.6 ±1.2	96.9 ±0.0	59.9 ±0.0
Layer-4	98.7 ±1.0	38.7 ±12.4	97.3 ±0.1	56.6 ±0.3
Layer-6	97.3 ±0.0	56.2 ±0.9	98.4 ±1.3	42.6 ±20.4
Layer-8	96.3 ±0.0	59.8 ±1.6	98.5 ±1.0	45.1 ±11.4
Layer-10	94.9 ±0.1	68.1 ±0.1	98.6 ±0.5	41.2 ±9.9
Layer-12	96.0 ±0.1	62.7 ±1.2	99.0 ±0.1	35.0 ±2.9

Table 9: dis-similarity Layers search.

	SimReg↓		∇SimReg ↓	
	bias	anti-bias	bias	anti-bias
Embeddings	98.9 ±0.0	35.7 ±0.1	94.7 ±0.4	57.1 ±2.4
Layer-2	98.8 ±0.1	37.1 ±0.3	77.2 ±14.2	40.1 ±34.9
Layer-7	99.0 ±0.0	38.4 ±0.1	91.1 ±10.0	43.6 ±21.3
Layer-12	98.7 ±0.0	36.4 ±1.0	100.0 ±0.0	0.5 ±0.7

A.4 Regularization set

In the main body we have reported results when performing similarity-based regularization only on the biased samples, $\mathcal{D}^{\mathcal{B}}$, as described in Section 3.3. In Table 10 we compare to the case of regularizing on all the training samples. Clearly, it is important to perform regularization only on $\mathcal{D}^{\mathcal{B}}$. This is especially true for the dissimilarity case, where the model regularized on all samples sometimes fails to converge, leading to random performance on MNLI dev and constant predictions on HANS (see last row).

Table 10: Comparing regularizing only on $\mathcal{D}^{\mathcal{B}}$ to regularizing on all samples.

Bias		MNLI	HANS	
		dev	non-entailment	entailment
Lexical-overlap	SimReg ↑	83.6 ± 0.1	57.5 ± 6.0	82.3 ± 2.7
	SimReg↑ (all)	84.4 ±0.0	7.4 ±1.4	98.4 ±0.4
	SimReg ↓	83.9 ± 0.1	40.1 ± 1.7	89.4 ± 3.6
	SimReg ↓ (all)	83.8 ± 0.2	18.5 ± 4.3	96.8 ± 1.3
	∇SimReg↑	83.3 ±0.4	56.4 ±2.8	75.6 ±0.8
	∇SimReg↑ (all)	46.1 ±52.8	61.7 ±54.1	44.4 ±62.7
	∇SimReg↓	84.0 ±0.1	48.8 ±6.4	89.8 ±5.5
	∇SimReg↓ (all)	32.8 ±0.2	100.0 ±0.0	0.0 ±0.0
Unknown	SimReg ↑	82.9 ± 0.2	54.1 ± 1.1	82.4 ± 2.5
	SimReg ↑ (all)	84.3 ± 0.1	7.3 ± 0.7	98.3 ± 0.2
	SimReg ↓	82.9 ± 0.3	41.4 ± 2.5	85.6 ± 4.3
	SimReg ↓ (all)	83.1 ± 0.2	3.6 ± 0.4	98.9 ± 0.2
	∇SimReg↑	82.7 ±0.1	54.4 ±2.5	75.0 ±2.4
	∇SimReg↑ (all)	82.8 ±0.0	32.7 ±0.1	82.7 ±0.0
	∇SimReg ↓	82.8 ± 0.2	40.0 ± 2.0	86.2 ± 6.3
	∇SimReg ↓ (all)	32.3 ± 0.6	100.0 ± 0.0	0.0 ± 0.0

A.5 Synthetic Bias

In this section we present more detailed results for synthetic-MNLI. In Table 11 we show wider range of configuration for the case of increasing similarity. Note that λ values for resulted in models with better performance on the anti-biased set. Bert-base is BERT trained on synthetic-MNLI, while BERT (Oracle) is trained on MNLI. In Table 12 we present a partial list of decreasing similarity experiments.

Table 11: Increasing similarity: Synthetic-MNLI evaluation for prevalence=1 and strength=0.95, ∇ denotes regularizing gradients.

	Biased	Anti-biased	Unbiased
Bert-base	98.5 ± 0.1	41.8 ± 1.1	78.0 ± 0.4
BERT (Oracle)	83.5 ± 0.3	82.0 ± 0.9	84.0 ± 0.3
CosCor			
SimReg $\uparrow$ ($\lambda = 1$)	97.5 ± 0.0	57.0 ± 0.6	82.6 ± 0.2
SimReg $\uparrow$ ($\lambda = 10$)	96.9 ± 0.0	59.3 ± 0.8	83.0 ± 0.1
SimReg $\uparrow$ ($\lambda = 100$)	96.8 ± 0.1	60.3 ± 0.6	82.7 ± 0.1
∇SimReg $\uparrow$ ($\lambda = 1$)	96.8 ± 0.1	60.8 ± 0.6	82.7 ± 0.5
∇SimReg $\uparrow$ ($\lambda = 10$)	95.5 ± 0.2	56.7 ± 2.6	81.1 ± 1.0
∇SimReg $\uparrow$ ($\lambda = 100$)	82.8	72.1	79.1
Linear-CKA			
SimReg $\uparrow$ ($\lambda = 1$)	97.3 ± 0.0	57.7 ± 0.1	83.3 ± 0.2
SimReg $\uparrow$ ($\lambda = 10$)	96.8 ± 0.1	60.2 ± 0.4	83.6 ± 0.3
SimReg $\uparrow$ ($\lambda = 100$)	96.7 ± 0.1	61.0 ± 0.9	83.2 ± 0.2
∇SimReg $\uparrow$ ($\lambda = 1$)	98.1 ± 0.1	31.3 ± 1.0	70.4 ± 1.7
∇SimReg $\uparrow$ ($\lambda = 10$)	96.7 ± 0.1	17.5 ± 0.2	51.3 ± 0.7
∇SimReg $\uparrow$ ($\lambda = 100$)	32.8 ± 0.0	32.2 ± 0.0	32.7 ± 0.0

Table 12: Decreasing similarity: Synthetic-MNLI evaluation for prevalence=1 and strength=0.95.

Model	Biased	Anti-biased	Unbiased
Bert-base	98.5 ± 0.1	41.8 ± 1.1	78.0 ± 0.4
Biased-BERT	99.9	05.7	64.3
CosCor			
SimReg$\downarrow$ ($\lambda = 1$)	99.6 ± 0.6	12.2 ± 17.3	53.4 ± 25.9
SimReg $\downarrow$ ($\lambda = 10$)	97.0 ± 0.1	49.0 ± 2.4	76.1 ± 2.0
SimReg$\downarrow$ ($\lambda = 100$)	33.5 ± 1.6	32.3 ± 0.4	33.3 ± 1.6
Linear-CKA			
SimReg $\downarrow$ ($\lambda = 1$)	99.2 ± 0.1	21.9 ± 6.8	69.0 ± 0.4
SimReg $\downarrow$ ($\lambda = 10$)	32.3 ± 0.7	32.2 ± 0.0	32.3 ± 0.6
SimReg $\downarrow$ ($\lambda = 100$)	32.3 ± 0.7	32.6 ± 0.6	32.3 ± 0.7

A.6 Bias recovery

Here we present the full results of our bias-recovery experiments.

A.7 Threshold choosing

A.8 Results with DeBERTa-V1

Table 13 provides results with DeBERTa-V1. SimReg consistently improves the baseline, demonstrating its success with stronger models than BERT.

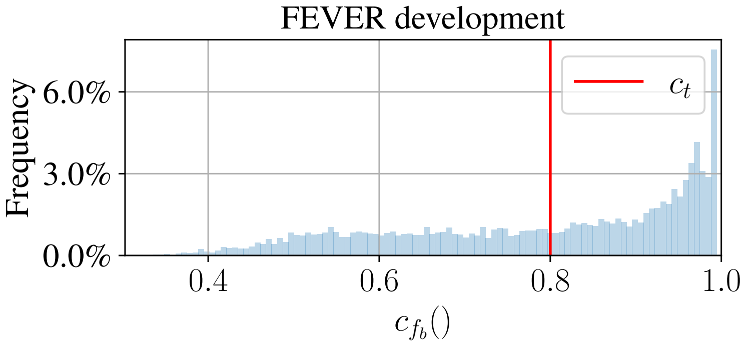

Figure 4: Confidence distribution of a claim-only model (f_b) on FEVER; here $c_t = 0.8$.

Table 13: Results with stronger models - DeBERTa V1.

	Hypothesis		Lexical	
	IID	OOD	IID	OOD
baseline	88.2 ±0.2	82.3 ±0.1	88.2 ±0.2	75.2 ±0.9
Known bias				
SimReg ↑	88.2 ±0.0	83.1 ±0.0	87.5 ±0.0	80.5 ±0.6
SimReg ↓	87.0 ±0.6	82.1 ±0.3	87.8 ±0.1	77.3 ±1.5
∇SimReg ↑	87.2 ±0.1	82.7 ±0.2	87.5 ±0.2	79.8 ±0.6
∇SimReg ↓	85.0 ±1.1	81.7 ±2.7	87.8 ±0.0	78.4 ±1.3
Unknown bias				
SimReg ↑	88.3 ±0.1	83.0 ±0.3	88.3 ±0.1	74.6 ±1.5
SimReg ↓	87.7 ±0.0	82.3 ±0.6	87.7 ±0.0	76.5 ±0.3
∇SimReg ↑	85.4 ±0.2	80.8 ±0.4	85.4 ±0.2	73.7 ±0.7
∇SimReg ↓	87.3 ±0.1	81.8 ±0.4	87.3 ±0.1	78.5 ±1.6