Generating Data to Mitigate Spurious Correlations in Natural Language Inference Datasets

Anonymous ACL submission

Abstract

Natural language processing models often exploit spurious correlations between task-independent features and labels in datasets to perform well only within the distributions they are trained on, while not generalising to different task distributions. We propose to tackle this problem by generating a debiased version of a dataset, which can then be used to train a debiased, off-the-shelf model, by simply replacing its training data. Our approach consists of 1) a method for training *data generators* to generate high-quality, label-consistent data samples; and 2) a filtering mechanism for removing data points that contribute to spurious correlations, measured in terms of *z-statistics*. We generate debiased versions of the SNLI and MNLI datasets, and we evaluate on a large suite of debiased, out-of-distribution, and adversarial test sets. Results show that models trained on our debiased datasets generalise significantly better than those trained on the original datasets in all settings. On the majority of the datasets, our method outperforms or performs comparably to previous state-of-the-art debiasing strategies, and when combined with an orthogonal technique, product-of-experts, the performance improves further and achieves state-of-the-art results of SNLI-hard and MNLI-hard.

1 Introduction

Natural Language Processing (NLP) datasets inevitably contain biases that are unrelated to the tasks they are supposed to represent. These biases are usually artifacts of the dataset collection, annotation processes, or design decisions (Schwartz et al., 2017; Geva et al., 2019; Liu et al., 2021). Such biases often manifest as spurious correlations between simple features of the data points and their labels (Gardner et al., 2021). Trained models can exploit these spurious correlations to correctly predict the labels of the data points within the same distributions as those they are trained on, but fail to generalise to other distributions within the same tasks. Consequently, the models risk modelling the datasets, but not the tasks (Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019; Schuster et al., 2019).

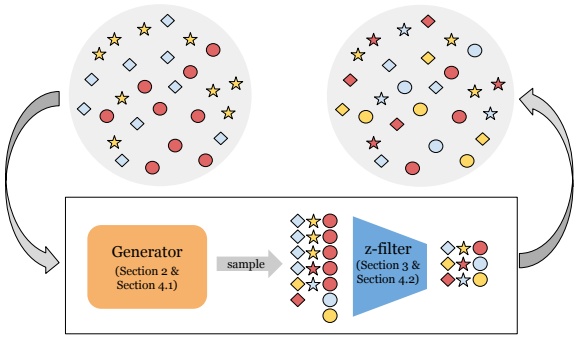

Figure 1: Overview of our dataset bias mitigation approach. We minimise spurious correlations between labels (represented by the shapes of data points) and task-independent features (represented by their colours) with our proposed data generation pipeline.

We address this issue by *adjusting* existing dataset distributions to mitigate the correlations between task-independent features and the labels. First, we train *data generators* that generate high quality data samples in the distribution of existing datasets (Section 2). Then, we identify a set of simple features that are known to be task-independent, and use the theoretical framework proposed by Gardner et al. (2021) to measure correlations (i.e., z-statistics) between those features and the labels. Finally, we use these measures of spurious correlations to adjust the distribution of samples generated by the data generator (Section 3) by post-hoc filtering (Section 3.2) to remove the data points that contribute to high z-scores with task-independent features, or finetuning the data generator (Section 4.1) to make such data points less likely. Unlike prior *model-centric* approaches for dealing with spurious correlations in datasets (Belinkov et al., 2019a,b; Clark et al., 2019; He et al., 2019; Karimi Mahabadi

et al., 2020) that define new training objectives or model architectures, our approach has the advantage of keeping the objective and the model fixed, as we only alter the training data.

To evaluate our approach, we use the task of Natural Language Inference (NLI), which offers a wide range of datasets (including challenge datasets) for various domains. We generate[1] debiased SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) distributions and evaluate the generalisability of models trained on them to out-of-distribution hard evaluation sets (Gururangan et al., 2018; McCoy et al., 2019), and the adversarial attack suite for NLI proposed by Liu et al. (2020b). Furthermore, we compare our method to strong debiasing strategies from the literature (Belinkov et al., 2019b; Stacey et al., 2020; Clark et al., 2019; Karimi Mahabadi et al., 2020; Utama et al., 2020; Sanh et al., 2021; Ghaddar et al., 2021).

Our results show that models trained on our debiased datasets generalise better than those trained on original datasets to evaluation sets targeting hypothesis-only biases (by up to 2.8 points) and syntactic biases (by up to 13 points), and a suite of adversarial tests sets (by up to 4.2 points on average). On the test sets targeting the hypotheis-only bias, they perform comparably to the best performing models tweaked specifically to handle the bias, but since our contributions are orthogonal to such improvements, we show that combining the two yields further improvements of up to 1.6 points.

2 Generating High-Quality Data Samples

First, we need to train a data generator G to generate data samples automatically. Our goal for the data generator is to model the base distribution as well as possible so that we can generate valid and high-quality data samples.

2.1 Finetuning Pretrained Language Model to Generate NLI Samples

We finetune a pretrained language model on the NLI datasets to serve as our data generator. We choose GPT-2 because it is a powerful and widely-used causal language model.

Given an NLI dataset $\mathcal{D}_0$, the training objective is to minimise the following negative log-likelihood loss of generating the premise-label-hypothesis sequence, in that order:

$$\mathcal{L}_{MLE} = -\sum_{i=1}^{|\mathcal{D}_0|} \log p(P^{(i)}, H^{(i)}, l^{(i)})$$

$$= -\sum_{i=1}^{|\mathcal{D}_0|} \log p(P^{(i)}) p(l^{(i)}|P^{(i)}) p(H^{(i)}|l^{(i)}, P^{(i)}), \quad (1)$$

where $P^{(i)}$, $H^{(i)}$ and $l^{(i)}$ are the premise, hypothesis and label respectively.[2]

2.2 Improving Data Generation Quality

2.2.1 Unlikelihood Training to Improve Label Consistency

We find that a generator trained with only $\mathcal{L}_{MLE}$ has poor *label consistency*. This means that given a generated sample $(\tilde{P}, \tilde{H}, \tilde{l})$, the label $\tilde{l}$ often does not correctly describe the relationship between $\tilde{P}$ and $\tilde{H}$. We apply *unlikelihood training* (Welleck et al., 2020) to tackle this problem. First we perturb the label to construct negative samples (P, H, l') where $l' \neq l$ for each sample in the dataset. Then we apply a token-level unlikelihood objective on the hypothesis tokens:

$$\mathcal{L}_{\text{consistency}} = -\sum_{i=1}^{|\mathcal{D}_0|} \sum_{t=1}^{|H|^{(i)}} \log(1 - p(H_t^{(i)}|l'^{(i)}, P^{(i)}, H_{<t}^{(i)})).$$

This objective decreases the probability of generating H when given an incorrect label l'; hence it improves the label consistency.

We combine $\mathcal{L}_{MLE}$ and $\mathcal{L}_{\text{consistency}}$ to finetune our generator G with

$$\mathcal{L}_G = \mathcal{L}_{MLE} + \lambda \mathcal{L}_{\text{consistency}},$$

where λ is a hyper-parameter that balances the two objectives. We can randomly sample from the trained generator to obtain a large amount of the synthetic data $\mathcal{D}_G \sim G$.

2.2.2 Filtering Based on Model Confidence

To further improve the quality of the generated dataset, we use an NLI model M trained on the original dataset $\mathcal{D}_0$ to filter out samples in which M has low confidence:

$$\hat{\mathcal{D}}_G = \{(P, H, l) \in \mathcal{D}_G \mid p_M(l|P, H) > \tau\},$$

[1] We will release all our code and the generated datasets.

[2] In our preliminary study, we found the factorization order premise-label-hypothesis in Eq. (1) performs better than hypothesis-label-premise and premise-hypothesis-label.

where τ is a confidence threshold. We found that the filtered out data samples generally had incorrect labels and/or ungrammatical text.

3 Mitigating Spurious Correlations using *z-filtering*

We now define a method to reject samples that contribute to the high spurious correlations between task-independent features of the samples and their labels. Our approach is based on the theoretical framework proposed by Gardner et al. (2021) to measure these correlations, known as *z-statistics*. Our filtering method, called *z-filtering* (Section 3.2), will serve as the basis to construct debiased datasets in Section 4.

3.1 Identifying and Measuring Spurious Correlations

As a first step towards addressing spurious correlations, we need to be able to quantify them. We start by selecting a set of features that are trivially task-independent and seek to ensure that the labels are not correlated with these features. These features are listed as follows: *1*) unigrams and bigrams; *2*) hypothesis length and hypothesis-premise length ratio; *3*) lexical overlap between hypothesis and premise; *4*) the predictions of a BERT-base (Devlin et al., 2019) hypothesis-only model.[3] We choose these features as they capture various biases identified in prior work, including contradiction word biases, lexical overlap bias (McCoy et al., 2019), and hypothesis-only bias (Gururangan et al., 2018; Poliak et al., 2018). Note that our method does not rely on the specific choice of features, and one can easily add alternative features that should not correlated with the labels.

Following Gardner et al. (2021), we assume there should be no correlation between each of these features and the class labels. More formally, for any feature x from our feature set $\mathcal{X}$, $p(l|x)$ should be uniform over the class labels. We define $\hat{p}(l|x) = \frac{1}{n}\sum_{j=1}^{n} l^j$ to be the empirical expectation of $p(l|x)$ over n samples containing x. Then we compute the standardised version of *z-statistics* to quantify its deviation from the uniform distribution for each features x and label l:

$$z^*(x,l) = \frac{\hat{p}(l|x) - p_0}{\sqrt{p_0(1-p_0)/n}}, \qquad (2)$$

where p_0 is the probability of uniform distribution ($p_0 = 1/3$ in NLI tasks with three labels).

These z-statistics scores can be used to identify the most biased features for each label l – we select k features with the highest *z-statistic* to define the *biased features* set $\mathcal{B}_{\mathcal{D}}(l)$. Table 10 shows examples of these biased features on SNLI.

3.2 z-filtering

To mitigate the biases in the dataset, we propose *z-filtering*, an algorithm that iteratively selects and filters instances from a dataset $\mathcal{D}'$ to build a debiased dataset $\mathcal{Z}$. At each step, we find the set of biased features $\mathcal{B}_{\mathcal{Z}}(l)$ on the partially constructed $\mathcal{Z}$. We then select a new batch of samples from $\mathcal{D}'$ and filter out the samples that contain these biased features. This process is applied iteratively until it has exhausted all samples from $\mathcal{D}'$. It removes the samples that contribute to the spurious correlations in $\mathcal{D}'$, thus it finds a debiased subset $\mathcal{Z}(\mathcal{D}') \subset \mathcal{D}'$. We denote the removed samples as $\mathcal{Z}^{-}(\mathcal{D}')$. The full z-filtering algorithm is illustrated in Algorithm 1.

Optionally, one can initialise $\mathcal{Z}$ with a seed dataset $\mathcal{D}_{seed}$. In this case, the samples from $\mathcal{D}'$ are only added to $\mathcal{Z}$ when they do not contain the biased features of $\mathcal{D}_{seed}$. Thus it can be seen as a data-augmentation technique targeted to debias a given dataset. We refer to it as *conditional z-filtering* and denote the produced debiased dataset as $\mathcal{Z}(\mathcal{D}'|\mathcal{D}_{seed})$.

4 Constructing Debiased NLI Datasets via Data Generation

We use z-filtering in two ways: *1*) to further finetune G with an objective that downweighs samples that should be rejected (Section 4.1); *2*) to post-hoc filter samples in $\hat{\mathcal{D}}_G$ to obtain debiased datasets (Section 4.2).

4.1 Learning to Generate Unbiased Samples

The generator G can learn to exploit task-independent features during its finetuning stage (Section 2), causing the synthetic data $\hat{\mathcal{D}}_G$ to contain many spurious correlations. While it is tempting to apply z-filtering to remove these spurious correlations from $\hat{\mathcal{D}}_G$, we find that this will lead to the removal of majority of the generated data. For example, when the generator is finetuned on SNLI,

[3] See Appendix B for detailed descriptions of the features.

Algorithm 1: z-filtering algorithm.

Data: input dataset $\mathcal{D}'$ [with optional seed dataset $\mathcal{D}_{seed}$]
Result: debiased dataset $\mathcal{Z}$ and the rejected samples $\mathcal{Z}^-$

```
Z ← ∅ (or Z ← D_seed);
Z⁻ ← ∅;
for sample batch D'_t ⊂ D' do
    compute or update z-statistics
      z*(x, l|Z), ∀x ∈ X of Z;
    find the biased features B_Z(l), ∀l ∈
      {entailment, neutral, contradiction};
    foreach instance I = (P, H, l) ∈ D'_t do
        get the features f of the instance I;
        if f ∩ B_Z(l) = ∅ then
            Z ← Z ∪ {I};
        else
            Z⁻ ← Z⁻ ∪ {I};
        end
    end
end
```

z-filtering removes around 85% of $\hat{\mathcal{D}}_{G_{SNLI}}$.[4] This leads to a very inefficient data generation process to mitigate the spurious correlations.

To alleviate this issue, we can incorporate the debiasing objectives into the training of the generator, so that the samples produced by the generator are more likely to be accepted by the z-filtering stage. More specifically, we can encourage the model to generate $\mathcal{Z}(\mathcal{D}_0)$, while discouraging it from generating $\mathcal{Z}^-(\mathcal{D}_0)$. For the latter part, we again apply an unlikelihood training objective $\mathcal{L}_{UL}$ to *unlearn* $\mathcal{Z}^-(\mathcal{D}_0)$. Hence, the overall debiasing training objective is:

$$\mathcal{L}_{debias} = \mathcal{L}_{MLE}(\mathcal{Z}(\mathcal{D}_0)) + \alpha\mathcal{L}_{UL}(\mathcal{Z}^-(\mathcal{D}_0))$$

where α is a hyperparamter.

A naive use of an unlikelihood objective on all tokens gives the model mixed signals for good tokens and leads to ungrammatical, degenerate outputs. To avoid this degeneracy, we apply the unlikelihood loss only to tokens that contribute to biased features. The unlikelihood loss applies only to the tokens that contribute to biased features. Concretely, for each token I_t^- of instance $I^- \in \mathcal{Z}^-(\mathcal{D}_0)$, we define a mask m_t as

$$m_t = \begin{cases} 0, & \text{if } I'_t \text{ contributes to } \mathcal{B}_{\mathcal{Z}}(l_{I^-}) \\ 1, & \text{otherwise.} \end{cases}$$

where $\mathcal{B}_{\mathcal{Z}}(l_{I^-})$ represent the biased features corresponding the label of I^-.

For biases towards unigram and bigram features (as defined in Section 3.1), we consider only the corresponding tokens to be relevant (i.e., $m_t = 0$ if I_t^- is part of the unigram or the bigram). For biases towards other features (e.g. length of the hypothesis), we consider all the tokens on the hypothesis to be relevant. The unlikelihood training objective is defined as follows:

$$\mathcal{L}_{UL}(\mathcal{Z}^-(\mathcal{D}_0)) = \sum_{I' \in \mathcal{Z}^-(\mathcal{D}_0)} \mathcal{L}_{UL}(I'),$$

$$\mathcal{L}_{UL}(I') = -\sum_{t=1}^{|I'|} \log(m_t p(I'_t|I'_{<t}) + (1 - m_t)(1 - p(I'_t|I'_{<t}))).$$

We further finetune G with $\mathcal{L}_{debias}$ to obtain a new generator G^*, that is trained to generate more unbiased data samples. We then randomly sample from G^* and conduct data filtering (Section 2.2.2) to obtain a large set of high-quality debiased data samples $\hat{\mathcal{D}}_{G^*}$.

4.2 Combining with z-filtering to Construct the Debiased NLI Datasets

Given the original dataset $\mathcal{D}_0$ and the synthetic dataset $\hat{\mathcal{D}}_{G^*}$, our goal is produce a large-scale unbiased dataset $\mathcal{D}^*$. There are various ways to do this given that we can either apply conditional z-filtering, or simply z-filter both $\mathcal{D}_0$ and $\hat{\mathcal{D}}_{G^*}$ and merge them. We explore the following options:

1. **Z-Augmentation (Z-Aug)** $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}|\mathcal{D}_0)$: we treat the original dataset as is, and augment it by conducting conditional z-filtering on $\hat{\mathcal{D}}_{G^*}$ using $\mathcal{D}_0$ as seed dataset.

2. **Parallel z-filter (Par-Z)** $\mathcal{Z}(\mathcal{D}_0) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$: we conduct z-filtering on $\mathcal{D}_0$ and $\hat{\mathcal{D}}_{G^*}$ separately, and then merge them.

3. **Sequential z-filter (Seq-Z)** $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}|\mathcal{Z}(\mathcal{D}_0))$: we first conduct z-filtering on $\mathcal{D}_0$, then conduct conditional z-filtering on $\hat{\mathcal{D}}_{G^*}$ with $\mathcal{Z}(\mathcal{D}_0)$ as seed dataset.

[4] This is also strong confirmation that these biases are problematic, as the generative model easily finds them and relies on them during data generation. Conducting naive data augmentation with $\hat{\mathcal{D}}_{G_{SNLI}}$ will strengthen the spurious correlations.

5 Experiments

5.1 Experimental Setup

Source Datasets We select the two most widely used NLI datasets SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) as our original datasets. Prior work (Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019) found various annotation artifacts in them, hence they serve as good use cases for constructing debiased datasets.

Evaluation Datasets For the hypothesis-only bias, we use the challenge sets SNLI-hard (Gururangan et al., 2018) and MNLI-hard (Williams et al., 2018), which were produced by filtering the test set with a hypothesis-only model (Section 5.2). For syntactic biases, we follow previous work and use HANS (McCoy et al., 2019) for evaluation (Section 5.3). In addition, we evaluate on the adversarial test benchmark introduced by Liu et al. (2020b) (Section 5.4). This benchmark covers a wide range of adversarial attacks, which will give a more complete picture of what spurious correlations the debiasing methods tackle.

Generating Debiased Datasets We conduct debiased data generation for SNLI and MNLI *separately*. For SNLI, we use the proposed method described in Section 4.1 to train a generator G^*_{SNLI}. Then we randomly sample a large number of instances from the generator to construct $\mathcal{D}_{G^*_{\text{SNLI}}}$. The samples are filtered with a strong NLI model M trained on SNLI to obtain $\hat{\mathcal{D}}_{G^*_{\text{SNLI}}}$. Finally, different options (Section 4.2) can be adopted to merge the synthetic data with the original data $\mathcal{D}_{\text{SNLI}}$ to construct debiased versions of SNLI. The same procedure is used to produce debiased datasets for MNLI, by simply replacing the original dataset with MNLI. We choose GPT-2 large and Roberta-large as the pretrained language models for G^* and M respectively.[5] The size of the constructed debiased datasets are listed in Table 1.

NLI Model Training Since our method directly debiases the training data itself, we keep the model and training objective fixed and only replace the training data with our generated debiased datasets. For comparability with previous work (Karimi Mahabadi et al., 2020; Utama et al., 2020; Sanh et al., 2021), we train BERT-base (Devlin et al., 2019) on our debiased datasets. The NLI models are trained with ordinary *cross-entropy* classification loss, and the training hyperparameters are listed in Appendix A. We run our experiments five times and report the average and standard deviation of the scores.[6]

Options	$\mathcal{D}_0 = \mathcal{D}_{\text{SNLI}}$	$\mathcal{D}_0 = \mathcal{D}_{\text{MNLI}}$
Original $\mathcal{D}_0$	549,367	382,702
Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_0)$	1,142,475	744,326
Par-Z $\mathcal{Z}(\mathcal{D}_0) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$	933,085	740,811
Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_0))$	927,906	744,200

Table 1: Data size of the constructed debiased datasets for SNLI and MNLI.

State-of-the-art Debiasing Models We compare our method with the following three state-of-the-art debiasing models on each of our evaluation datasets. Product-of-Experts (He et al., 2019; Karimi Mahabadi et al., 2020), that ensembles a bias-only model's prediction b_i with the main model's p_i using $p'_i = softmax(\log p_i + \log b_i)$. This ensembling enforces that the main model focuses on the samples that the bias-only model does not predict well. Learned-Mixin (Clark et al., 2019), that is a variant of PoE that introduces a learnable weight for the bias-only model's prediction. The model proposed by Utama et al. (2020), that uses confidence regularisation to retain the in-distribution performance while conducting model debiasing.

Combining PoE with Our Debiased Datasets Our approach changes the training data distribution instead of the model's training objective, and hence is orthogonal to prior work method-wise. We also report the results of combining PoE with our proposed method, simply by training a PoE model on our debiased datasets. We adapt the PoE implementation[7] by Karimi Mahabadi et al. (2020), and we follow their approach to conduct hyperparameter tuning for PoE.

5.2 Hypothesis-only Bias in NLI

Gururangan et al. (2018) found that, on SNLI and MNLI, a model that has only access to the hypothesis can perform surprisingly well, which indicates that the datasets contain hypothesis-only bias.

[5] On one A100 GPU, training the generator takes around 24 hours and generating the samples takes roughly 35 hours for each dataset.

[6] With the exception of our PoE experiments which single run, as hyperparameter tuning for PoE is costlier.

[7] https://github.com/rabeehk/robust-nli

Method (model w/ data)	SNLI	SNLI-hard
Prior debiasing strategies trained on SNLI		
AdvCls (Belinkov et al., 2019a)∗	83.56	66.27
Ens. AdvCls (Stacey et al., 2020)∗	84.09	67.42
DFL (Karimi Mahabadi et al., 2020)∗	89.57	**83.01**
PoE (Karimi Mahabadi et al., 2020)∗	90.11	82.15
BERT-base w/ $\mathcal{D}_{\text{SNLI}}$ *baseline*	90.45	$80.34_{\pm 0.46}$
Models trained on our debiased datasets		
BERT-base w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{SNLI}})$	**90.67**	$81.78_{\pm 0.53}$
BERT-base w/ Par-Z $\mathcal{Z}(\mathcal{D}_{\text{SNLI}}) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$	88.11	$82.81_{\pm 0.37}$
BERT-base w/ Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{SNLI}}))$	88.08	$\mathbf{82.82}_{\pm 0.15}$
Combining PoE with our debiased datasets		
BERT-base + PoE w/ $\mathcal{D}_{\text{SNLI}}$	90.25	82.92
BERT-base + PoE w/ Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{SNLI}}))$	87.65	**84.48**

Table 2: Results on SNLI and SNLI-hard. ∗ are reported results.

To alleviate this problem, SNLI-hard and MNLI-hard (Gururangan et al., 2018) subsets were constructed by filtering the test set with a hypothesis-only model and only accepting those that the hypothesis-only model predicts incorrectly. We examine whether our method successfully mitigates the hypothesis-only bias in NLI, by evaluating the models trained with our debiased datasets on SNLI-hard and MNLI-hard.

Results on SNLI Table 2 shows the results of our method on SNLI. The results show that, compared to training on SNLI, training with our debiased datasets significantly improves the performance on SNLI-hard. The debiased dataset produced by Seq-Z achieves a 2.48% gain in accuracy on SNLI-hard compared to the SNLI baseline, whereas Z-Aug improves both SNLI and SNLI-hard accuracy.

Results on MNLI Table 3 shows the results of our method on MNLI-matched (MNLI-m) and MNLI-mismatched (MNLI-mm), and their corresponding hard sets. We use the development sets of MNLI-hard reconstructed by (Karimi Mahabadi et al., 2020) to develop our methods. To comply with the submission limit of MNLI submission system, we select the best checkpoint among the five runs using the development set, and report its test set performance in Table 3.

The results show that BERT-base models trained on our debiased MNLI datasets outperform the models trained on the original MNLI by a large margin on the MNLI-hard sets. In particular, the Z-Aug version of the debiased datasets gives a 2.72% and 2.76% gain in accuracy on MNLI-m hard and MNLI-mm hard respectively, and outperforms the previous state-of-the-art on MNLI-m, MNLI-mm, and MNLI MNLI-mm hard.

Combining PoE with Our Debiased Datasets We investigate the combination of our method and PoE, to see if the two orthogonal techniques can work together to achieve better performance. Since hyperparameter tuning of PoE is costly, we choose the best version of the debiased dataset (Seq-Z on SNLI and Z-Aug on MNLI) using the development set accuracy, and train PoE with it. The results are listed in the last rows of Table 2 and Table 3. We can find that, on both SNLI and MNLI, combining PoE with our debiased dataset yields further improvements on SNLI-hard, MNLI-m hard, and MNLI-mm hard, achieving new state-of-the-art results on all three datasets.

5.3 Syntactic Bias in NLI

McCoy et al. (2019) show that NLI models trained on MNLI can exploit syntactic heuristics present in the data, such as lexical overlap, subsequence, and constituent features. They introduce HANS, an evaluation dataset that contains examples where the syntactic heuristics fail. To test whether our method mitigates the syntactic biases in NLI, we evaluate models trained on our debiased datasets on HANS. If our debiased dataset contains less syntactic bias than the original dataset, the model would not exploit the syntactic heuristics and thus perform better on HANS. Due to the high variance of the scores on HANS, we run five times for each experiment (except PoE), and report the average and standard deviation of the scores.

Results on HANS Table 4 shows the results on HANS. The results are categorised into three sections according to the original data: SNLI, MNLI, and our debiased datasets. With SNLI as the original dataset, we compare with TAILOR (Ross et al., 2021), a semantically controlled data augmentation method that uses heuristics specifically designed to tackle syntactic biases. Following TAILOR, we train Roberta-base models with our debiased SNLI dataset (Seq-Z). The results show that the performance of our debiased dataset outperforms the SNLI baseline, and is also slightly better than TAILOR. This is surprising because TAILOR relies on specifically designed heuristics to generate samples, whereas our method does not require such manual heuristics.

The results of models train on our debiased MNLI datasets also show strong improvements: compared to the original MNLI, our debiased

Method (model w/ data)	MNLI-m		MNLI-mm		MNLI-m hard		MNLI-mm hard	
	dev	test	dev	test	dev	test	dev	test
Prior debiasing strategies trained on MNLI								
PoE (Karimi Mahabadi et al., 2020)∗	84.58	84.11	84.85	83.47	78.02	76.81	79.23	76.83
Learned-Mixin (Clark et al., 2019)∗	80.5	79.5	81.2	80.4	-	**79.2**	-	78.2
Regularized-conf (Utama et al., 2020)∗	84.6	84.1	85.0	84.2	-	78.3	-	77.3
BERT-base Main PoE+CE (Sanh et al., 2021)∗	83.32	-	83.54	-	-	77.63	-	76.39
BERT-base w/ $\mathcal{D}_{\text{MNLI}}$ *baseline*	83.87	84.11	84.22	83.51	$76.39_{\pm 0.64}$	75.88	$77.75_{\pm 0.45}$	75.75
Models trained on our debiased datasets								
BERT-base w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	**84.72**	**85.12**	**85.14**	**84.09**	$\mathbf{78.95}_{\pm 0.76}$	78.60	$\mathbf{80.29}_{\pm 0.54}$	**78.51**
BERT-base w/ Par-Z $\mathcal{Z}(\mathcal{D}_{\text{MNLI}}) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$	82.48	83.27	82.95	82.95	$78.88_{\pm 0.80}$	**79.19**	$80.02_{\pm 0.62}$	78.49
BERT-base w/ Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{MNLI}}))$	82.55	83.41	82.70	83.17	$78.88_{\pm 0.83}$	**79.19**	$79.65_{\pm 0.44}$	78.44
Combining PoE with our debiased dataset								
BERT-base + PoE w/ $\mathcal{D}_{\text{MNLI}}$	84.39	-	84.25	-	78.37	77.54	79.45	78.33
BERT-base + PoE w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	**85.22**	**85.38**	**85.72**	**84.53**	**80.49**	**80.03**	**81.52**	**79.28**

Table 3: Results on MNLI-matched (MNLI-m), MNLI-mismatched (MNLI-mm), MNLI-matched hard, and MNLI-mismatched hard. ∗ are reported results.

Method	HANS
Methods trained on SNLI	
Roberta-base w/ $\mathcal{D}_{\text{SNLI}}$ (Ross et al., 2021)∗	64.72
Roberta-base w/ TAILOR (Ross et al., 2021)∗	66.45
Methods trained on MNLI	
Learned-Mixin (Clark et al., 2019)∗	64.00
Learned-Mixin+H (Clark et al., 2019)∗	66.15
PoE (Karimi Mahabadi et al., 2020)∗	$66.31_{\pm 0.6}$
DFL (Karimi Mahabadi et al., 2020)∗	$69.26_{\pm 0.2}$
PoE+CE (Sanh et al., 2021)∗	67.9
Regularized-conf (Utama et al., 2020)∗	$69.1_{\pm 1.2}$
E2E Self-debias (Ghaddar et al., 2021)∗	$\mathbf{71.2}_{\pm 0.2}$
Models trained on our debiased datasets	
Roberta-base w/ $\mathcal{D}_{\text{SNLI}}$ (ours)	$65.32_{\pm 2.22}$
Roberta-base w/ $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{SNLI}}))$ (ours)	$\mathbf{66.87}_{\pm 1.47}$
BERT-base w/ $\mathcal{D}_{\text{MNLI}}$ (baseline)	$54.36_{\pm 2.56}$
BERT-base w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	$62.57_{\pm 5.91}$
BERT-base w/ Par-Z $\mathcal{Z}(\mathcal{D}_{\text{MNLI}}) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$	$65.11_{\pm 5.62}$
BERT-base w/ Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{MNLI}}))$	$\mathbf{67.69}_{\pm 3.53}$
BERT-base + PoE w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	**68.75**
Roberta-large w/ $\mathcal{D}_{\text{MNLI}}$ (baseline)	$75.74_{\pm 2.82}$
Roberta-large w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	$\mathbf{78.65}_{\pm 2.26}$

Table 4: Results on HANS (McCoy et al., 2019). ∗ are reported results.

MNLI datasets obtain up to a 13.33% gain in HANS accuracy. Our Seq-Z variant achieves 67.69% accuracy, which is comparable with strong baselines PoE (Karimi Mahabadi et al., 2020) and (Sanh et al., 2021).

Additionally, we train a Roberta-large model on our debiased MNLI dataset. Training on our debiased dataset introduces 2.9 points accuracy gain on HANS, indicating that the performance gain by our debiased dataset generalises to larger and stronger model.

5.4 Adversarial Tests for Combating Distinct Biases in NLI

Liu et al. (2020b) find that debiasing methods often tie to one particular known bias and it is nontrivial to mitigate multiple NLI biases at the same time. They introduce a suite of test datasets for NLI models that targets various aspects of robustness, including partial input heuristics (PI), logical inference ability (LI), and stress test (ST).[8]

Several data augmentation strategies were investigated by Liu et al. (2020b): 1) text swap: swapping the premise and hypothesis in the original data; 2) word substitution: replacing words in the hypothesis with synonyms or generations from a masked language model; 3) paraphrase: using back translation to paraphrase the hypothesis.

We compare the our approach with their data-augmentation heuristics, and the results are shown in Table 5. Comparing with the MNLI baseline, our debiased MNLI datasets lead to better performance across all categories, which indicates that our method successfully mitigates various distinct biases simultaneously. All three variants of our debiased datasets outperform the data augmentation heuristics by Liu et al. (2021), which again demonstrates the efficacy of our method when compared against manually designed heuristics.

[8]Refer to Appendix C for detailed description of the subcategories.

	PI-CD	PI-SP	IS-SD	IS-CS	LI-LI	LI-TS	ST	Avg.
Data-augmentation heuristics proposed by Liu et al. (2020b)								
Text Swap∗	71.7	72.8	63.5	67.4	86.3	**86.8**	66.5	73.6
Sub (synonym)∗	69.8	72.0	62.4	65.8	85.2	82.8	64.3	71.8
Sub (MLM)∗	71.0	72.8	64.4	65.9	85.6	83.3	64.9	72.6
Paraphrase∗	72.1	74.6	66.5	66.4	85.7	83.1	64.8	73.3
BERT-base w/ $\mathcal{D}_{\text{MNLI}}$ *baseline*	$70.3_{\pm 0.5}$	$73.7_{\pm 1.4}$	$53.5_{\pm 2.3}$	$64.8_{\pm 1.4}$	$85.5_{\pm 0.9}$	$81.6_{\pm 1.4}$	$69.2_{\pm 0.8}$	$71.2_{\pm 0.8}$
Models trained on our debiased datasets								
BERT-base w/ Z-Aug $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{D}_{\text{MNLI}})$	$\mathbf{73.1}_{\pm 0.9}$	$76.1_{\pm 1.2}$	$61.8_{\pm 6.1}$	$69.1_{\pm 1.3}$	$86.9_{\pm 0.6}$	$83.1_{\pm 0.9}$	$\mathbf{70.1}_{\pm 0.5}$	$74.3_{\pm 1.3}$
BERT-base w/ Par-Z $\mathcal{Z}(\mathcal{D}_{\text{MNLI}}) \cup \mathcal{Z}(\hat{\mathcal{D}}_{G^*})$	$72.0_{\pm 0.9}$	$\mathbf{78.7}_{\pm 1.2}$	$64.5_{\pm 5.8}$	$70.7_{\pm 1.7}$	$88.5_{\pm 0.7}$	$82.6_{\pm 0.3}$	$69.6_{\pm 1.0}$	$75.2_{\pm 1.4}$
BERT-base w/ Seq-Z $\mathcal{Z}(\hat{\mathcal{D}}_{G^*}\|\mathcal{Z}(\mathcal{D}_{\text{MNLI}}))$	$71.7_{\pm 0.9}$	$77.8_{\pm 1.2}$	$\mathbf{66.9}_{\pm 3.7}$	$\mathbf{71.1}_{\pm 0.7}$	$\mathbf{89.1}_{\pm 1.0}$	$82.3_{\pm 0.9}$	$69.3_{\pm 0.8}$	$\mathbf{75.4}_{\pm 0.8}$

Table 5: Results on the NLI adversarial test benchmark (Liu et al., 2020b). We compare with the data augmentation techniques investigated by Liu et al. (2020b) and ∗ are reported results.

6 Related Work

Spurious Correlations in Datasets The issue of spurious correlations in datasets between labels and simple input features has recently received significant attention (Gururangan et al., 2018; Poliak et al., 2018; Belinkov et al., 2019a; Karimi Mahabadi et al., 2020). It has been shown that this issue is often inherent in the data annotation process, caused by biases in the framing of the task (Schwartz et al., 2017), noisy annotations (Chen et al., 2016), or personal (Geva et al., 2019) or group-level (Liu et al., 2021) annotator biases. Gardner et al. (2021) provide a theoretical framework for analyzing spurious correlations in language understanding problems, which we use to define our filtering mechanism in Section 3.2.

Debiasing NLI Models Much prior work follows a *model-centric* approach towards mitigating biases in NLI models—they propose novel model architectures or training objectives to ensure that the models do not exploit the shortcuts presented by the dataset biases. At the representation level, Belinkov et al. (2019a,b) introduce an adversarial architecture to debias hypothesis representations to tackle hypothesis-only bias (Gururangan et al., 2018), and Stacey et al. (2020) strengthen the debiasing by using multiple adversarial classifiers. Zhou and Bansal (2020) use HEX projection to project the representation to the space orthogonal to the biased features to debias the model. At the model level, Clark et al. (2019); He et al. (2019); Karimi Mahabadi et al. (2020) propose methods based on Product-of-Expert (PoE) (Hinton, 2002) for mitigating biases by ensembling a biased-only model with a main model. Utama et al. (2020) propose the use of confidence regularization to improve out-of-distribution performance while retaining in-distribution accuracy.

Debiasing NLI Datasets Prior work towards debiasing NLI datasets *perturb* individual instances as a data augmentation strategy. Ross et al. (2021) introduce TAILOR, a semantically-controlled perturbation method for data augmentation based on a small number of manually defined perturbation strategies. Lee et al. (2021) train a generator to generate new claims and evidence for debiasing fact verification datasets. Unlike their approach, our method does not require manually-written heuristics or additional datasets to construct augmented datasets, and hence is more generally applicable.

7 Conclusions

To address the issue of spurious correlations between task-independent features and labels in NLI datasets, we propose methods to generate label-consistent data and then filter out instances from existing datasets that contribute to those spurious correlations; thereby generating debiased datasets. Models trained on our debiased versions of the SNLI and MNLI datasets generalise better than the equivalent model trained on the original datasets to a large suite of test sets focusing on various kinds of known biases, even though our method does not explicitly counteract those biases. Future work in this direction includes investigating whether our techniques are applicable to tasks beyond NLI.

References

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019a. 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, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019b. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

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

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT (2)*, pages 107–112. 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 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

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

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3181–3185.

Haochen Liu, Joseph Thekinen, Sinem Mollaoglu, Da Tang, Ji Yang, Youlong Cheng, Hui Liu, and Jiliang Tang. 2021. Toward annotator group bias in crowdsourcing. *arXiv preprint arXiv:2110.08038*.

Tianyu Liu, Zheng Xin, Baobao Chang, and Zhifang Sui. 2020a. HypoNLI: Exploring the artificial patterns of hypothesis-only bias in natural language inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6852–6860, Marseille, France. European Language Resources Association.

Tianyu Liu, Zheng Xin, Xiaoan Ding, Baobao Chang, and Zhifang Sui. 2020b. An empirical study on model-agnostic debiasing strategies for robust natural language inference. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 596–608, Online. Association for Computational Linguistics.

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

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In **SEM@NAACL-HLT*, pages 180–191. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.

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

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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. 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*, pages 8717–8729, Online. Association for Computational Linguistics.

Haohan Wang, Da Sun, and Eric P Xing. 2019. What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7136–7143.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

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.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

A Hyper-parameters

Hyper-parameter	Value
learning rate	1e-5
batch size	24
epoch	5
optimiser	Adam
Adam ϵ	1e-6
Adam (β_1, β_2)	(0.9, 0.999)
learning rate scheduler	constant
max sequence length	128
pretrained model	GPT-2 large
device	Nvidia A100
λ	0.5
α	1.0

Table 6: Hyper-parameters for the generator G^*.

Hyper-parameter	Value
number of samples from G^*_{SNLI}	5,000,000
number of samples from G^*_{MNLI}	4,000,000
data filtering threshold τ	0.95
data filtering model	Roberta-large
z-filtering number of biased features	20

Table 7: Hyper-parameters for the data generation pipeline.

Hyper-parameter	Value
learning rate	1e-5
batch size	32
epoch	5
optimiser	Adam
Adam ϵ	1e-6
Adam (β_1, β_2)	(0.9, 0.999)
learning rate scheduler	constant with warmup
warm up steps	2000
max sequence length	128
pretrained model	BERT-base
device	Nvidia A100
early stop patience	3 epochs

Table 8: Hyper-parameters for training NLI models.

B Biased Features

We list the set of task-independent features that we mitigate in this work in Table 9. Table 10 shows the most salient biased features identified on SNLI and our debiased SNLI dataset.

C Description of Adversarial Test (Liu et al., 2020b) Subcategories

The adversarial test benchmark (Liu et al., 2020b) includes the following subcategories from various sources:

- PI-CD: classifier detected partial-input (Gururangan et al., 2018).
- PI-SP: HypoNLI (Liu et al., 2020a) dataset that tackles surface patterns heuristics.
- IS-SD: syntactic diagnostic dataset HANS (McCoy et al., 2019).
- IS-CS: lexically misleading instances constructed by Nie et al. (2019).
- LI-LI: lexical inference test by (Naik et al., 2018; Glockner et al., 2018).
- LI-TS: text-fragment swap test by swapping the premise and hypothesis (Wang et al., 2019; Minervini and Riedel, 2018).
- ST: an aggregation of word-overlap (ST-WO), negation (ST-NE), length mismatch (ST-LM), and spelling errors (ST-SE) tests in (Naik et al., 2018).

D Visualisation of z-statistics

Following Gardner et al. (2021), we visualise the statistics of the features on both SNLI and our debiased SNLI (Seq-Z) dataset in Fig. 2.[9] Comparing the two plots, it confirms that our method successfully suppresses the spurious correlations in the dataset.

[9] it may be slow to load the figures because they are large.

Feature	Description
Unigrams & Bigrams	All unigrams and bigrams. The n-grams from premise and hypothesis are treated separately.
Hypothesis length	Number of tokens in the hypothesis.
Hypothesis-premise length ratio	Number of tokens in hypothesis divided by number of tokens in the premise.
Lexical overlap	Ratio of tokens in the hypothesis that overlap with the premise.
Hypothesis-only model's prediction	We train a hypothesis-only model on the original dataset and use its prediction as a feature.
Null feature	A dummy feature added for *all* instances to avoid skewed label distribution.

Table 9: Descriptions of the features used to debias the datasets in Section 3.

SNLI		Debiased SNLI (Seq-Z)	
Biased feature	**z-statistics**	**Biased feature**	**z-statistics**
Entailment			
hypo-only-pred=0	422.1	theres@hypothesis	17.5
lex-overlap> 0.8	123.3	hypo-len< 5	17.4
full-lex-overlap	117.3	full-lex-overlap	17.4
outside@hypothesis	102.2	politician@hypothesis	17.4
lex-overlap> 0.9	90.4	speaking@hypothesis	17.4
Neutral			
hypo-only-pred=1	436.1	championship@hypothesis	15.3
for a@hypothesis	63.6	living room@hypothesis	15.2
his@hypothesis	56.8	many men@hypothesis	15.2
friends@hypothesis	55.6	green suit@hypothesis	15.2
tall@hypothesis	52.7	are wearing@hypothesis	15.2
Contradiction			
hypo-only-pred=2	433.9	nothing@hypothesis	17.0
sleeping@hypothesis	92.9	hypo-only-pred=2	16.9
is sleeping@hypothesis	68.7	at home@hypothesis	16.9
nobody@hypothesis	68.4	is no@hypothesis	16.9
no@hypothesis	62.7	york yankees@hypothesis	16.9

Table 10: Top-5 biased features with the highest z-statistics on SNLI (left) and debiased SNLI Seq-Z (right) for each label class.

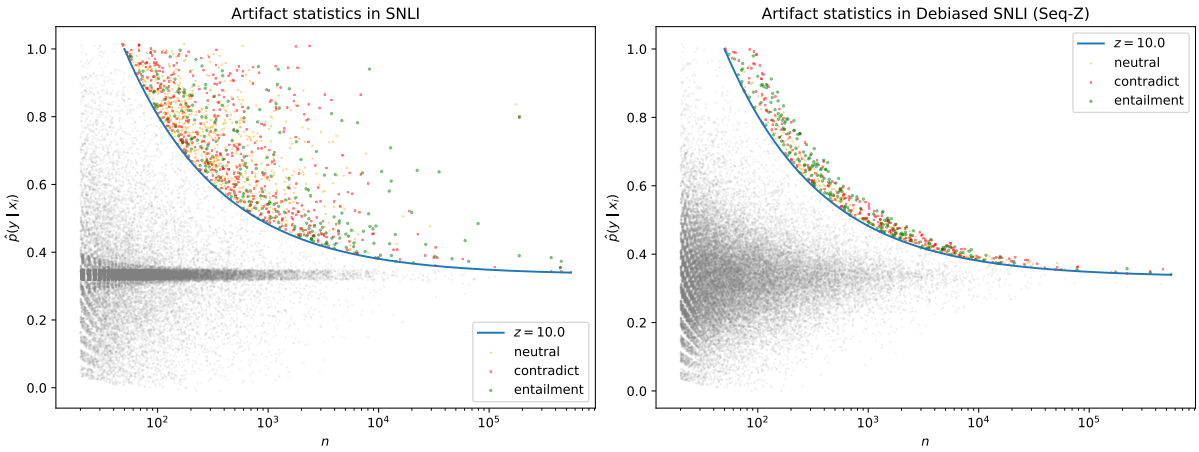

Figure 2: Statistics of the features on SNLI and our debiased SNLI (Seq-Z).