Decorrelate Irrelevant, Purify Relevant: Overcome Textual Spurious Correlations from a Feature Perspective

Shihan Dou[1*], **Rui Zheng**[1*], **Ting Wu**[1], **Songyang Gao**[1], **Junjie Shan**[3], **Qi Zhang**[12], **Yueming Wu**[4], **Xuanjing Huang**[1†]
[1] School of Computer Science, Fudan University, Shanghai, China
[2] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[3] KTH Royal Institute of Technology, Stockholm, Sweden
[4] Nanyang Technological University, Singapore
{shdou21,tingwu21,gaosy21}@m.fudan.edu.cn
{rzheng20,qz,xjhuang}@fudan.edu.cn

Abstract

Natural language understanding (NLU) models tend to rely on spurious correlations (*i.e.*, dataset bias) to achieve high performance on in-distribution datasets but poor performance on out-of-distribution ones. Most of the existing debiasing methods often identify and weaken these samples with biased features (*i.e.*, superficial surface features that cause such spurious correlations). However, down-weighting these samples obstructs the model in learning from the non-biased parts of these samples. To tackle this challenge, in this paper, we propose to eliminate spurious correlations in a fine-grained manner from a feature space perspective. Specifically, we introduce Random Fourier Features and weighted re-sampling to decorrelate the dependencies between features to mitigate spurious correlations. After obtaining decorrelated features, we further design a mutual-information-based method to purify them, which forces the model to learn features that are more relevant to tasks. Extensive experiments on two well-studied NLU tasks demonstrate that our method is superior to other comparative approaches.

1 Introduction

Recently, researchers have found that the main reason why large-scale pre-trained language models perform well on NLU tasks is that they rely on *spurious correlations*, rather than capturing the language understanding for the intended task (Bender and Koller, 2020). These spurious correlations are also denoted as *dataset bias* in previous work (He et al., 2019; Clark et al., 2019): prediction rules that work for training examples but do not hold in general. In reality, a variety of spurious correlations appear in widely-used NLU benchmark datasets. For example, in natural language inference (NLI) tasks, McCoy et al. (2019) observe that models on the MNLI dataset (Williams et al., 2018) rely heavily on the features of word overlap to predict the entailment label blindly. Consequently, these models perform poorly on out-of-distribution (OOD) datasets where such correlations no longer hold (Nie et al., 2019).

To mitigate these spurious correlations, some existing debiasing works (Clark et al., 2019; He et al., 2019) prefer to train a *bias model* with known spurious correlations as prior knowledge to identify the samples without biased features. This trained *bias model* is used in the later stage to force the *main model* to learn from these samples. For better transferability, Utama et al. (2020b); Sanh et al. (2020) relax this basic assumption that spurious correlation is apriori by using a small part of the training dataset in the training phase of *bias model*. However, these methods are not end-to-end and their training procedures are complicated. Moreover, not all features in the samples with biased features are insignificant (Wen et al., 2021). These samples may still contain features that generalize to the real-world dataset, and weakening these samples obstructs the model in learning from the non-biased parts of these samples (Wen et al., 2021).

In this paper, unlike the above-mentioned methods, we propose an end-to-end method that can eliminate the spurious correlations in a fine-grained way[1]. Recently, some works (Marcus, 2018; Arjovsky et al., 2019) have demonstrated that spurious correlations are essentially caused by the subtle dependencies between irrelevant features (*i.e.*, the features that are irrelevant to a given label) and relevant features. According to this observation, we intend to eliminate spurious correlations by decor-

[*] Equal contribution.
[†] Corresponding author.

[1] Our code is available at https://github.com/Coling2022-DePro/DePro.

relating the dependencies between features in the feature space. However, those irrelevant features still exist in the feature space and may confuse the analysis capability of deep models. To achieve better performance, we further design another component to purify the decorrelated features in the feature space, which forces the model to learn useful local features (*i.e.*, features that are more relevant to tasks (Wang et al., 2020)). Specifically, we address two main challenges:

- *Challenge 1: How to eliminate dependencies among features in the feature space?*
- *Challenge 2: How to find the useful local features and purify the decorrelated global features with them?*

To address the first challenge, some previous works (Shen et al., 2020) try to decorrelate features under linear frameworks. However, these linear frameworks are not capable of dealing with nonlinear dependencies between features in the feature space. To further enhance the effectiveness of methods on decorrelating nonlinear dependencies, an ideal candidate is to use kernel methods to remap the original features to high-dimensional feature space. In this way, both linear and nonlinear dependencies can be decorrelated. Nevertheless, the mapping operator of the kernel function cannot be given explicitly. Therefore, we use Random Fourier Features (RFF) (Rahimi and Recht, 2007) to approximate the kernel method for the sake of computability. After completing high-dimensional feature reconstruction, we introduce weighted resampling to remove the dependencies between reconstructed features in the reconstructed feature space. To tackle the second challenge, we introduce a saliency-map-based method to identify the useful local features in the samples and design a mutual-information-based strategy to purify the decorrelated global features (*i.e.*, sentence representation) with these useful local features.

We evaluate our framework over two NLU tasks including Natural Language Inference and Fact Verification. Through the experimental results, we observe that feature decorrelation and feature purification are both useful for improving the generalization ability of deep neural models. Moreover, our method can achieve state-of-the-art performance on predicting out-of-distribution datasets compared with existing approaches. In summary, this paper makes the following contributions:

- We introduce a novel end-to-end framework that combines feature decorrelation with feature purification to strengthen the generalization ability of NLU models. The feature decorrelation phase is used to eliminate spurious correlations of features while the feature purification component is used to force the model to learn features that are more relevant to tasks.
- We conduct extensive experiments over several widely used benchmark datasets. The experimental results report that feature decorrelation and feature purification can both enhance the generalization ability of deep models. Also, the results suggest the synergistic effect between decorrelation and purification. After combining them, our proposed method outperforms the state-of-the-art methods.

2 Related Work

2.1 Spurious Correlations and Debiasing Methods

The performance of machine learning models on multiple natural language understanding benchmarks has achieved remarkable results. However, due to the presence of spurious surface lexical-syntactic features in the training phase, deep models perform poorly on out-of-distribution examples. These spurious properties are also known as spurious correlations or dataset biases. For example, McCoy et al. (2019) reports that models on the MNLI dataset (Williams et al., 2018) rely heavily on high word overlap to predict the entailment label. In fact, spurious correlations also exist in datasets of other NLU tasks such as multi-hop QA datasets (Wen et al., 2021). Deep models' excessive dependence on these spurious correlations can affect their generalization ability when testing on more challenging datasets.

In response to the problem of spurious correlations in datasets, many methods have been proposed to mitigate the impact. For example, Clark et al. (2019); He et al. (2019) propose a two-stage-based framework to reduce the model's dependence on known spurious correlations. They first train a bias-only model using known spurious correlations and then leverage it to guide the main model to distinguish biased examples. However, these approaches suffer from low transferability since they require prior knowledge about the spurious correla-

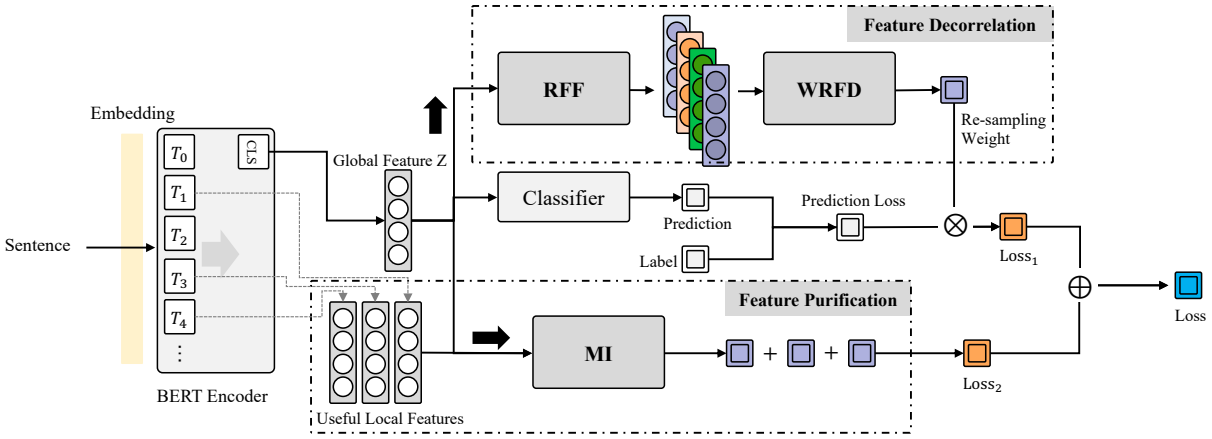

Figure 1: System architecture of *DePro*. RFF, WRFD, and MI refer to Random Fourier Features, Weighted Re-sampling for Feature Decorrelation, and Mutual Information, respectively.

tions in a dataset. To mitigate the issue, Utama et al. (2020b); Clark et al. (2020) tend to train a weak or shadow model as the bias-only model to provide guidance on discriminating biased data. However, these methods are not end-to-end and the training procedures of these methods are complicated.

2.2 Feature Decorrelation

Since the correlation between features can affect or even damage model predictions, several studies focus on eliminating this correlation during the training process. Zhang et al. (2017) propose a strategy that selects uncorrelated features in groups to decorrelate features. Shen et al. (2020) address this issue by re-weighting samples. However, these two methods can only remove the linear dependence between features which cannot tackle the complex nonlinear dependence between features. Bahng et al. (2020) propose to use the biased representations to generate a debiased representation. Although this method can decorrelate the nonlinear and linear dependence between features, it needs to artificially design the biased representation based on the known spurious correlations in the dataset. On the contrary, our method can remove all kinds of dependencies between the features and does not need to rely on prior knowledge.

3 Method

In this section, we introduce our proposed end-to-end framework namely *DePro*. Figure 1 presents the system architecture of *DePro* which mainly consists of two phases: feature decorrelation and feature purification. In the first phase, we introduce Random Fourier Features (RFF) (Rahimi and Recht, 2007) to map features from the original feature space to the reconstruction space. Then we use weighted re-sampling to remove the dependencies between reconstructed features. In the later phase, we purify the global sample features from an information theoretic perspective to further improve the generalization ability.

Notations $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ denote the space of samples (*i.e.*, sentences), the space of labels, and the feature space, respectively. We use $f : \mathcal{X} \rightarrow \mathcal{Z}$ to denote the encoder function which can encode a sample into the feature space. The classifier function is denoted as $c : \mathcal{Z} \rightarrow \mathcal{Y}$, which can predict the sample to the corresponding label. Given a dataset $\mathcal{D}$ that consists of n pairs of sentences and labels $(X_i, Y_i)_{i\in[1,n]}$, with $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$, the representation of X_i is denoted as $Z_i \in \mathcal{Z}$, and Z^i denotes the i-th variable in the feature space. For an input sentence $X_i = [X_i^1; X_i^2; \ldots; X_i^k]$, w_i denotes the re-sampling weight of this sentence X_i and we use $T_i = [T_i^1; T_i^2; \ldots; T_i^k]$ to denote the local feature of X_i in the encoder (*e.g.*, the output of BERT embedding layer).

3.1 Decorrelate Features of Feature Space

In this subsection, we mainly introduce our method of removing both the nonlinear and linear dependencies between features by using RFF and weighted re-sampling.

High-dimensional Feature Reconstruction via RFF

The kernel method can obtain mutually independent features by mapping them from the original feature space to Reproducing Kernel Hilbert Space (RKHS) (Alvarez et al., 2012) as follows:

$$\mathcal{K}(x,\cdot)=\sum_{i=1}^{\infty}\lambda_i\varphi_i(x)\varphi_i(\cdot)=\left(\sqrt{\lambda}_i\varphi_i(x),\cdots\right)_{\mathcal{H}} \quad (1)$$

Here $\mathcal{K}(\cdot,\cdot)$ is the mapping operator of a measurable, symmetric positive definite kernel function and $(\cdot)_{\mathcal{H}}$ is Hilbert-Schmidt space. However, the mapping operator $\mathcal{K}(x,\cdot)$ is implicit. In other words, the reconstructed features cannot be obtained explicitly. To mitigate this issue, we use Random Fourier Feature (RFF) (Rahimi and Recht, 2007), inspired by Zhang et al. (2021), to approximate the kernel function. The function space of Random Fourier Features is denoted as $\mathcal{H}$ with the following form:

$$\begin{aligned}\mathcal{H}=\{h:x\rightarrow\sqrt{2}\cos(\omega x+\phi)\mid\\ \omega\sim N(0,1),\phi\sim \mathrm{U}(0,2\pi)\}\end{aligned} \quad (2)$$

where ω and ϕ are sampled from any distribution.

For the i-th variable Z^i and the j-th variable Z^j of the feature space (Z^i and Z^j are represented by $\mathcal{A}$ and $\mathcal{B}$ for simplicity), we sample $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ mapping functions from $\mathcal{H}$ and denote them as $u=\{u_k\}_{k\in[1,n_{\mathcal{A}}]}$ and $v=\{v_k\}_{k\in[1,n_{\mathcal{B}}]}$. Thus, the reconstructed features $u(\mathcal{A})$ of feature A can be represented as Eq. (3) and $v(\mathcal{B})$ of feature B follows the same rule.

$$u(\mathcal{A})=\left(u_1(\mathcal{A}),\ldots,u_{n_{\mathcal{A}}}(\mathcal{A})\right),u_k(\cdot)\in\mathcal{H}_{\mathrm{RFF}},\forall k, \quad (3)$$

By mapping the two features A and B to the reconstructed space through RFF, only linear dependencies between $u(\mathcal{A})$ and $v(\mathcal{B})$ remain.

Weighted Re-sampling for Feature Decorrelation

We use cross-covariance operator Σ_{XY} to measure the independence between features as follows:

$$\begin{aligned}\langle g,\Sigma_{YX}f\rangle_{\mathcal{H}_2}=E_{XY}[f(X)g(Y)]-\\ E_X[f(X)]E_Y[g(Y)]\end{aligned} \quad (4)$$

Specifically, for $u(\mathcal{A})$ and $u(\mathcal{B})$, the cross-covariance Σ_{AB} between the distributions can be calculated by their unbiased empirical estimation with the following form:

$$\begin{aligned}\Sigma_{\mathcal{AB}}=\frac{1}{n-1}\sum_{i=1}^{n}\Big[\Big(u\left(\mathcal{A}_i\right)-\frac{1}{n}\sum_{j=1}^{n}u\left(\mathcal{A}_j\right)\Big)^T\cdot\\ \Big(v\left(\mathcal{B}_i\right)-\frac{1}{n}\sum_{j=1}^{n}v\left(B_j\right)\Big)\Big]\end{aligned} \quad (5)$$

Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2007) uses the squared Hilbert-Schmidt norm of $\Sigma_{\mathcal{AB}}$ to test the independence of random variables. In the Euclidean space which the reconstructed space belongs to, Hilbert-Schmidt norm degenerates to the equivalent Frobenius norm (Zhang et al., 2021). Thus, we use Frobenius norm to calculate the linear correlation between the reconstructed features.

Suppose $P(\mathcal{A},\mathcal{B})$ is denoted as the joint distribution of features $\mathcal{A}$ and $\mathcal{B}$. Due to the complicated correlation between $\mathcal{A}$ and $\mathcal{B}$, $P(\mathcal{A},\mathcal{B})$ cannot be obtained by their respective marginal distributions, which means $P(\mathcal{A},\mathcal{B})\neq P(\mathcal{A})\cdot P(\mathcal{B})$. Inspired by the Acceptance-Rejection Sampling method (Naesseth et al., 2017) which reparameterizes the target distribution function from the standard normal distribution by introducing the proposal distribution, we use the normalized weight function instead of the rejection process to obtain a linearly independent weighted marginal distribution from the original complex joint distribution. Specifically, consider a probability density function with the independent marginal distributions of $\mathcal{A}$ and $\mathcal{B}$ as $\mathcal{Q}(\mathcal{A},\mathcal{B})=\mathcal{Q}(\mathcal{A})\cdot\mathcal{Q}(\mathcal{B})$, the $\mathcal{Q}$ can be fitted by the proposal distribution $\mathcal{P}$ and the normalized sampling weight is denoted as follows:

$$w(x)=\frac{\mathcal{Q}(\S)}{\tau\mathcal{P}(x)} \quad (6)$$

where $x\in\mathcal{H}(\mathcal{A},\mathcal{B})$ and τ is a normalization constant with the following form:

$$\tau^{-1}=\int_{x\in\mathcal{H}(\mathcal{A},\mathcal{B})}w(x)dx \quad (7)$$

Thus, the linear dependencies between reconstructed features can be removed by the normalized weight function as follows:

$$w(x)\cdot x(\mathcal{A},\mathcal{B})\sim\tau w(x)\cdot P(x)=\mathcal{Q}(\mathcal{A})\cdot\mathcal{Q}(\mathcal{B}) \quad (8)$$

In practice, we use the training dataset to learn the optimal sampling weights. Through Eq. (5) and Eq. (8), the cross-covariance with weighted re-sampling can be estimated as:

$$\begin{aligned}\tilde{\Sigma}_{\mathcal{AB};w}=\frac{1}{n-1}\sum_{i=1}^{n}\Big[\Big(w_iu\left(\mathcal{A}_i\right)-\frac{1}{n}\sum_{j=1}^{n}w_ju\left(\mathcal{A}_j\right)\Big)^T\cdot\\ \Big(w_iv\left(\mathcal{B}_i\right)-\frac{1}{n}\sum_{j=1}^{n}w_jv\left(\mathcal{B}_j\right)\Big)\Big]\end{aligned} \quad (9)$$

As aforementioned, we use Frobenius norm to measure the correlation between features (*i.e.*, $\left\|\tilde{\Sigma}_{\mathcal{AB};w}\right\|_F^2$). Thus, by optimizing w in the training process, both nonlinear and linear dependencies between features of the feature space can be eliminated. Specifically, the correlation between the two

variables Z^i and Z^j of the feature space is represented as $\left\|\tilde{\Sigma}_{Z^iZ^j;w}\right\|_F^2$. Therefore, the re-sampling weight w can be optimized as follows:

$$w^* = \arg\min_{w \in \mathcal{W}} \sum_{1 \le i < j \le m_Z} \left\|\tilde{\Sigma}_{Z^iZ^j;w}\right\|_F^2 \quad (10)$$

where $\mathcal{W} = \left\{w \in \mathbb{R}_+^n \mid \sum_{i=1}^n w_i = n\right\}$ and m_Z denotes the dimension of space $\mathcal{Z}$. We use a mini-batch to update the global weight repeatedly during the optimization process. Moreover, the optimization objective function for encoder f and classifier c can be expressed as:

$$f^*, c^* = \arg\min_{f,c} \sum_{i=1}^n w_i \mathcal{L}(c(f(X_i)), y_i) \quad (11)$$

where $\mathcal{L}(\cdot,\cdot)$ is the cross entropy loss function.

3.2 Feature Purification via Local Information

For better generalization, we propose to purify the decorrelated global features from an information theoretic perspective. Specifically, we find the useful local features by a saliency-map-based method and purify the decorrelated global features with these local features by mutual information (MI).

Inspired by Han et al. (2020), we measure the significance of all local features of the sentence by computing the absolute value of the partial derivative of loss w.r.t. these local features. The gradient of each local feature can be calculated as:

$$\mathcal{G}(T^i) = \nabla_{T^i} \ell\left(f(T), y\right) \quad (12)$$

where T^i is the i-th feature of the local features T. We consider the part of the local features with the smallest values as the useless local features (*e.g.*, stopwords and punctuation) which carry limited information and cannot be used to make predictions (Wang et al., 2020). Therefore, the information of such useless features should not be encoded into the global features of a sentence.

After feature filtering, we treat these remaining local features as useful local features that are significant to the label (Wang et al., 2020), and use them to purify the decorrelated sentence representation by mutual information. Specifically, by maximizing the mutual information between the useful local features and the decorrelated sentence representation, the useful features are retained and the useless features are compressed. In practice, we simply examine the ℓ_2 norm of the gradient $\mathcal{G}(T^i)$ of each local feature T^i. The optimization goal can be expressed as:

$$\arg\max_{f,c} \ \alpha \sum_{j=1}^M I(T^j; Z) \quad (13)$$

where α is a hyper-parameter to control the trade-off, T^j is the above-mentioned useful local semantic feature, and M is the number of remaining features. In addition, due to the intractability of computing MI, we use InfoNCE (Oord et al., 2018) as the lower bound of MI to approximate $I(T^j; Z)$.

Combining Eq. (11) and Eq. (13), the overall optimization goal can be as follows:

$$\begin{aligned} f^*, c^* = \arg\min_{f,c} \ & \sum_{i=1}^n (w_i \mathcal{L}(c(f(X_i)), Y_i) - \\ & \alpha \sum_{j=1}^M \hat{I}^{(\text{InfoNCE})}(f_T(X_i^j); f(X_i))) \end{aligned} \quad (14)$$

$$w^* = \arg\min_{w \in \mathcal{W}n} \sum_{1 \le i < j \le m_Z} \left\|\tilde{\Sigma}_{Z^iZ^j;w}\right\|_F^2 \quad (15)$$

where $f_T(\cdot)$ is the function (*i.e.*, the BERT embedding layer) that obtains the local features.

4 Experiments

In this section, we conduct extensive experiments to demonstrate (1) *DePro* outperforms the state-of-the-art comparative approaches; and (2) Both feature decorrelation and feature purification can improve the model's generalization ability.

4.1 Datasets

We conduct experiments on two well-studied NLU tasks including Natural Language Inference and Fact Verification to evaluate *DePro*. **Natural Language Inference** aims to infer the relationship between the premise and hypothesis. For this task, we use MNLI (Williams et al., 2018) as our ID data, MNLI-hard (Gururangan et al., 2018) and Heuristic Analysis for NLI Systems (HANS) (McCoy et al., 2019) as our OOD test set. **Fact Verification** aims to verify a claim by the evidence document. For this task, we use FEVER (Thorne et al., 2018) for ID evaluation and FEVER Symmetric (Schuster et al., 2019) (version 1&2) as our OOD test set.

Specifically, we report the main results and ablation studies on the test set and evaluate all the sensitivity analyses on the development set. However, for the MNLI dataset, only the train set and

Method	MNLI			FEVER		
	ID	MNLI-hard	HANS	ID	Symm. v1	Symm. v2
BERT-base	84.3	75.9	61.1	85.4	55.2	63.2
prior knowledge required						
Learned-Mixin + H (Clark et al., 2019)	84.2	-	65.8	83.3	60.4	64.9
Reg-conf (Utama et al., 2020a)	84.5	77.3	69.1	**86.4**	60.5	66.2
Reweight (Clark et al., 2019)	83.5	-	69.2	84.6	61.7	66.5
PoE + CE (He et al., 2019)	83.3	77.6	67.9	85.7	57.7	61.4
prior knowledge NOT required						
MCE (Clark et al., 2020)	83.3	77.6	64.4	-	-	-
Reg-conf (Utama et al., 2020b)	84.3	-	67.1	87.6	59.8	66.0
PoE (Sanh et al., 2020)	81.4	76.5	68.8	85.4	59.7	65.3
MoCaD (Xiong et al., 2021)	81.5	-	70.0	87.4	**65.7**	69.0
DePro (Our method)	83.2	**77.8**	**70.3**	84.5	65.2	**69.2**
w/o feature decorrelation	**84.7**	76.8	63.2	85.9	57.5	65.2
w/o feature purification	82.6	77.1	68.7	83.6	64.3	67.9

Table 1: Accuracy results on MNLI and FEVER, and out-of-distribution test sets MNLI-hard, HANS and FEVER Symmetric (version 1&2). We conduct the ablation study to further validate that our feature decorrelation and feature purification indeed improve the generalization ability. We compared 8 state-of-the-art debiasing methods including 4 debiasing methods with known bias and 4 debiasing methods with unknown bias. The hyper-parameters of BERT are identical for each model in the same dataset.

dev set are publicly available, but not the published test set. So we split 10 percent of training data into a dev set dedicated to picking hyper-parameters in order to avoid overfitting. And the original dev set of MNLI is used as the test set.

4.2 Implementation

Similar to the majority of current debiasing methods, we choose the uncased BERT-base model (Devlin et al., 2018) as our baseline. For all sentence-pair classification tasks, we concatenate the two sentences of one sentence pair into a single sequence and use the final-layer [CLS] embedding to represent the sentence representation. For BERT hyper-parameters, we use a batch size of 32, Adam optimizer with the learning rate $5e^{-5}$ for the MNLI dataset and $2e^{-5}$ for the FEVER dataset, respectively.

For feature decorrelation, we set the learning rate of weight to $1e^{-2}$ which decays with a rate of $1e^{-3}$ for the MNLI dataset, and the learning rate to $5e^{-2}$ which decays with a rate of $1e^{-3}$ for the FEVER dataset. For the parameter of Random Fourier Features dimension, we have verified through extensive experiments that our method can get the best performance on HANS, Symm. v1, and Symm. v2 when the RFF dimensions are four times, two times, and four times, respectively, that of the original feature space. For feature purification, α is set to $1e^{-4}$ to control the trade-off between feature decorrelation and feature purification.

Method w/o Prior Knowledge	End-to-End
MCE (Clark et al., 2019)	✖
Reg-conf (Utama et al., 2020b)	✖
PoE (Sanh et al., 2020)	✖
MoCaD (Xiong et al., 2021)	✖
DePro (Our method)	☑

Table 2: The structural details of state-of-the-art methods without the need for prior knowledge.

4.3 Experimental Results

Detection Performance

Table 1 shows the experimental results of *DePro* and comparative methods on the MNLI and FEVER datasets, respectively. Through the table, we can see that *DePro* can significantly improve the performance of the two NLU tasks and obtain state-of-the-art results on OOD datasets. Meanwhile, the

loss of *DePro* on ID datasets is not significant compared to other methods. Moreover, we also observe that the experimental results of the model under different random seeds have high variance, which has been demonstrated in previous works (Utama et al., 2020b). To mitigate this impact, we perform our experiments with five different seeds and report the average of these results.

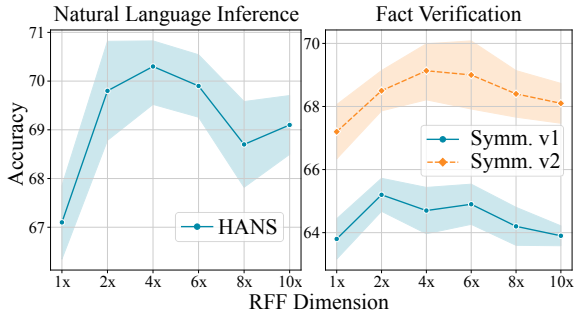

Figure 2: The results of *DePro* using different RFF dimensions. Meanwhile, the ratios of feature purification for HANS, Symm. v1, and Symm. v2 are 0.7, 0.7, and 0.6, respectively.

For the NLI task, compared to the baseline method (*i.e.*, Uncased BERT-base model), *DePro* improves by 1.9 and 9.2 percentage points on two OOD datasets MNLI-hard and HANS, respectively. The generalization ability of *DePro* on OOD datasets is also promising compared to other methods that introduce prior knowledge or unknown prior knowledge. For the Fact Verification task, *DePro* also has the best performance on the OOD dataset Symm. v2, with 6.0 percentage points higher than the accuracy of the BERT-base model. Meanwhile, the performance of our proposed method *DePro* is second only to MoCaD (Xiong et al., 2021), which is 0.5 percent lower when evaluated on Symm. v1. However, MoCaD is not an end-to-end method, but rather an improved version of the existing two-stage methods, as shown in Table 2. On the contrary, *DePro* is a complete end-to-end method, which is more flexible while preserving similar detection capabilities.

In conclusion, *DePro* outperforms the majority of state-of-the-art approaches on OOD datasets for two NLU tasks while the loss in ID datasets is acceptable.

Ablation Study

We also perform two ablation experiments to check whether feature decorrelation and purification can contribute to *DePro* or not. Through the results in Table 1, we find that feature decorrelation and purification can both boost the generalization ability of *DePro*. As aforementioned, the essence of spurious correlation is the subtle dependencies between relevant and irrelevant features. Therefore, after removing dependencies between features, we can mitigate the impact caused by spurious correlations, thus improving the model's generalization ability on OOD datasets. The results in Table 1 are consistent with this situation. On the other hand, if we directly perform feature purification on the original features, the model's performance on ID datasets can be enhanced. It is reasonable because feature purification can align the useful local features and the sentence representation, so that the representation generated by the model is more independent of useless local features, allowing the model to focus on the useful parts of the training data. After combining feature decorrelation with feature purification, *DePro* can achieve state-of-the-art performance on distinguishing samples in OOD datasets. Such results indicate that compared to aligning uncorrelated sentence representation, using feature purification on decorrelated representation enables sentence representation to better align the useful local features while staying away from the useless local features.

In conclusion, both feature decorrelation and feature purification can improve the detection ability, but if we can first remove the dependencies between features and then purify these decorrelated features, the generalization ability of the model can be improved to the level of state-of-the-art.

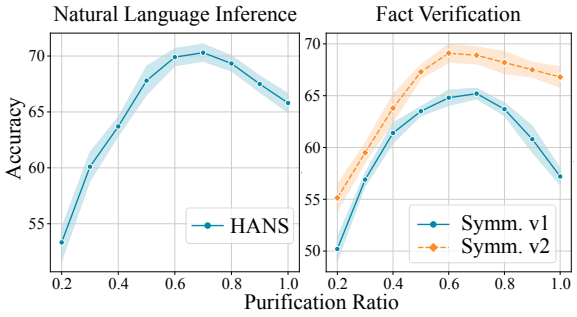

Figure 3: The results of *DePro* using different purification ratios. Meanwhile, the RFF dimensions for HANS, Symm. v1, and Symm. v2 are 4x, 2x, and 4x, respectively.

Sensitivity Analysis

In this part, we further explore the effect of the mapping dimension size of RFF and the degree of feature purification on the generalization ability of the

model. Specifically, we choose six different RFF dimensions and nine different purification ratios to commence our study. Due to the limited pages, we only show the corresponding experimental results of the best parameters in Figure 2 and Figure 3. For the NLI task, *DePro* performs the best when the RFF dimension is four times that of the original features and the top 70% of the features are used for purification. In addition, for the FEVER dataset, *DePro* can maintain the best results on Symm. v1 and Symm. v2 when the RFF dimensions are two times and four times, respectively, that of the original features and the top 70% and 60%, respectively, of the features are used for purification. Through these two figures, we see that the detection effect of *DePro* is different when choosing different RFF dimensions and different purification ratios. When the dimension is expanded to a certain number, the dependencies between features can be easily removed. At this point, when continuing to increase the dimension, it may bring additional overhead and impact, making the detection effect decrease instead.

For feature purification, if too many local features are removed, it can make the aligned sentence representation contain too little information. Moreover, if too many local features are purified, it may make the sentence representation contain too much useless information, so that the subsequent classifier cannot make predictions well based on the sentence representation.

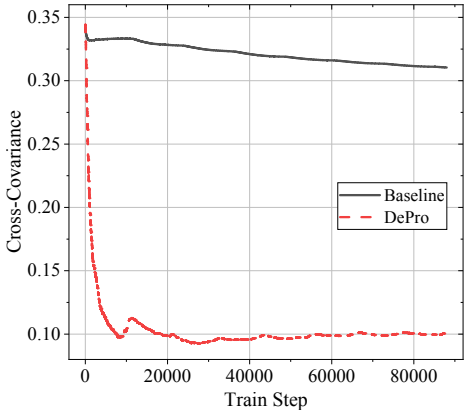

Figure 4: The mean of the correlations (*i.e.*, cross-covariance) between features at different iterations.

Decorrelation Study

Finally, we check whether feature decorrelation can remove the dependencies between features or not. Specifically, during the training phase, we record the mean of the correlations between features at different iterations. For the baseline experiment, we use the same RFF mapping functions to map the features to high-dimensional space. However, the reconstructed features are only used to calculate the cross-covariance, not to calculate the loss and optimize the parameters. Through the comparative results in Figure 4, we observe that the cross-covariance between features can be reduced as the number of iterations increases in *DePro*. However, in the baseline experiment, it barely decreases.

Overall, *DePro* can effectively remove dependencies between features. In this way, the spurious correlations can be mitigated at the feature level.

DePro	MNLI		FEVER	
	ID	HANS	ID	Symm. v2
With β-VAE	82.7	67.3	83.6	65.9
With RFF	**83.2**	**70.3**	**84.5**	**69.2**

Table 3: Evaluation results of the feature decorrelation phase leveraging Random Fourier Features (Rahimi and Recht, 2007) and β-VAE (Higgins et al., 2017) on two tasks, respectively.

4.4 Discussion

In this subsection, we primarily discuss two aspects: (1) Why we choose Random Fourier Features to decorrelate features in the feature decorrelation component; and (2) What distinguishes this work from prior works that use RFF to decorrelate features.

Many works (Rahimi and Recht, 2007; Zhang et al., 2021; Kingma and Welling, 2014) have been proposed to improve the generalization of the model by performing latent representation decorrelation learning. We compare the performance of two decorrelation methods RFF (Rahimi and Recht, 2007) and β-VAE (Higgins et al., 2017) in our model structure. The performance results are illustrated in Table 3, which shows that RFF outperforms β-VAE in our model both in ID and OOD datasets. In contrast to RFF, VAEs decorrelate the representation while compressing it, thus damaging the generalization ability. So we choose RFF to decorrelate the feature representation to obtain the uncompressed decorrelated representation, which benefits succeeding feature purification to distinguish useful from useless local features.

The distinction between *DePro* and other RFF-

based methods (Rahimi and Recht, 2007; Giffon et al., 2019; Zhang et al., 2021) is that our proposed method not only uses RFF for feature decorrelation but also combines two complementary approaches (*i.e.*, feature decorrelation and feature purification). These two methods are not mutually exclusive. In Section 4.3, we analyze the relationship between these two in detail, that is, the decorrelated features can be better purified, allowing the model to ignore more impurities when purifying useful features. Moreover, after feature decorrelation, feature purification can constrain the model to concentrate more on useful features rather than useless features.

5 Conclusion

In this paper, to improve the generalization ability of deep models on OOD datasets, we design an end-to-end framework called *DePro* which can eliminate spurious correlations and purify the decorrelated features. Extensive experiments on two well-studied NLU tasks demonstrate the synergistic effect between decorrelation and purification. After combining them, our method outperforms state-of-the-art methods in terms of effectiveness.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. We would also like to thank Feng Cheng, Haoxiang Jia, Wenxuan Li, and Yuhao Zhou for their help during the revision phase of the paper. This work was partially National Natural Science Foundation of China (No. 62076069, 61976056), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

References

Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. 2012. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. 2020. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198.

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.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luc Giffon, Stéphane Ayache, Thierry Artières, and Hachem Kadri. 2019. Deep networks with adaptive nyström approximation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. 2007. A kernel statistical test of independence. *Advances in neural information processing systems*, 20.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563.

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.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Gary Marcus. 2018. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

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.

Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. 2017. Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics*, pages 489–498. PMLR.

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2020. Learning from others' mistakes: Avoiding dataset biases without modeling them. *arXiv preprint arXiv:2012.01300*.

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.

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kunag. 2020. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5692–5699.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. *EMNLP 2018*, 80(29,775):1.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. 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.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Zhiquan Wen, Guanghui Xu, Mingkui Tan, Qingyao Wu, and Qi Wu. 2021. Debiased visual question answering from feature and sample perspectives. *Advances in Neural Information Processing Systems*, 34.

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.

Ruibin Xiong, Yimeng Chen, Liang Pang, Xueqi Cheng, Zhi-Ming Ma, and Yanyan Lan. 2021. Uncertainty calibration for ensemble-based debiasing methods. *Advances in Neural Information Processing Systems*, 34:13657–13669.

Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5372–5382.

Zhihong Zhang, Yiyang Tian, Lu Bai, Jianbing Xiahou, and Edwin Hancock. 2017. High-order covariate interacted lasso for feature selection. *Pattern Recognition Letters*, 87:139–146.