Stable Natural Language Understanding via Invariant Causal Constraint

Anonymous ACL submission

Abstract

Natural Language Understanding (NLU) task requires the model to understand the underlying semantics of input text. However, recent analyses demonstrate that NLU models tend to utilize dataset biases to achieve high dataset-specific performances, which always leads to performance degradation on out-of-distribution (OOD) samples. To increase the performance stability, previous debiasing methods *empirically* capture bias features from data to prevent model from corresponding biases. However, we argue that, the semantic information can form a *causal* relationship with the target labels of the NLU task, while the biases information is only *correlative* to the target labels. Such difference between the semantic information and dataset biases still remains not fully addressed, which limits the effectiveness of debiasing. To address this issue, we analyze the debiasing process under a *causal perspective*, and present a causal invariance based stable NLU framework (CI-sNLU). Experimental results show that CI-sNLU can consistently improve the stability of model performance on OOD datasets.

1 Introduction

State-of-the-art Natural Language Understanding (NLU) models such as BERT have demonstrated promising performance on various tasks (Devlin et al., 2019; Liu et al., 2019). These NLU models are generally first pretrained to learn universal language representations, then finetuned to adapt to specific downstream tasks. However, recent analyses demonstrate that these models tend to exploit the dataset *biases* spuriously associated with the target labels, rather than learn the underlying semantic information (McCoy et al., 2019; Clark et al., 2019; Sanh et al., 2020). This leads to performance degradation on out-of-distribution (OOD) samples.

To mitigate the impact of dataset biases and obtain NLU models that have stable performance on both in-distribution samples and OOD samples, a number of *debiasing* methods have been proposed. These methods work by first identifying the potential dataset biases within the dataset, then regularizing NLU model prevent it from capturing the bias information. To identify the potential biases, one line of debiasing works depends on the intuitions of researchers to design features characterizing the distribution of dataset biases (Schuster et al., 2019; Clark et al., 2019; He et al., 2019). However, the assumption that the types of bias should be known a-priori limits their application to many NLU tasks and datasets. Hence, automatic debiasing methods are proposed to move beyond the reliance on prior knowledge. These works usually train a biased model to to automatically capture the dataset bias and obtain a set of *bias features*. Then based on the identified biases information, model regularization methods such as Product-of-Expert (Hinton, 2002) or Confidence Regularization (Hinton et al., 2015) can be employed to prompt model to focus on learning the semantic information.

While promising, previous debiasing methods work by empirically inducing bias features from data. However, we argue that, the semantic information of text is *causal* to the target label, while the bias information has only *correlative* relationship. This drives to the essential difference between the bias information and semantic information. Present debiasing methods are still unaware of knowledge about causal invariance. Hence, the effectiveness of bias feature identification could be rather limited, influencing the efficiency of debiasing.

Figure 1 provides an example for illustrating the difference between semantic information and bias information in the causal perspective. In specific, it is the similar semantics between the Premise: *A cat caught a mouse.* and Hypothesis: *A mouse was caught by a cat.*, that *causes* the label to be "entailment". Therefore, the semantic information forms a *causal* relationship with the label. Fur-

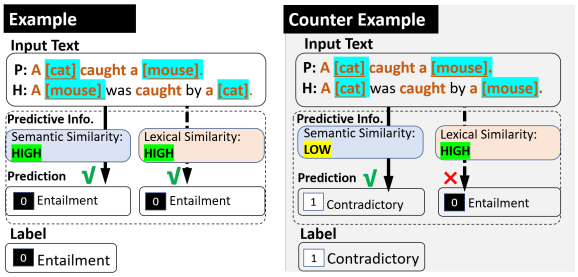

Figure 1: The semantic information is *causal* to the target label of NLU while the bias information such as lexical overlap is just *correlative*. The correlation may fail to exist on some counter examples.

thermore, as commented by Pearl et al. (2000), such causal relationship would keep *invariant* upon both in-distribution and OOD samples. On the contrary, the bias features, such as lexical overlaps, are just *correlative* to the labels. The correlation may vary upon different instances, and across different datasets, and thus without causal invariance. Hence, if the NLU model can debiase under the perspective of causal invariance, then it would have stable performance on OOD samples.

To facilitate those issues, in this paper, we propose a **C**ausal-**I**nvariance-based **s**table **NLU** framework (CI-sNLU). Based on the difference between the bias information and semantic information in *causal invariance*, CI-sNLU can find the "counter examples" on which model fails to capture the semantic information, then detect the bias information that model captures by comparing these "counter examples". Then by enforcing model to follow a causal invariance constraint, we can exclude the bias information model captured to increase the stability of performance. Furthermore, theoretical analyses demonstrate that our model regularization method can approximately minimize the mutual information between the representation of input text with the identified bias feature *close to 0*.

Experimental results show that, our approach can enhance the recognition of biased features and regularize model more efficiently, to consistently improve model stability on multiple OOD datasets, meanwhile persevere the in-distribution performance.

2 Stable Natural Language Understanding under Causal Perspective

We first analyze the stable NLU process under a causal perspective. The natural language understanding (NLU) task requires a model to understand the semantic of input text and then predict the target label. Formally, it can be characterized by a projection $X \rightarrow Y$, where X and Y denote the input text and the label, respectively. A NLU model $\mathcal{M}$ is trained to capture the predictive informations within X, and get a representation of input text $h^{\mathcal{M}} \in \mathbb{R}^d$. For brevity, in the following sections, we call $h^{\mathcal{M}} \in \mathbb{R}^d$ as *model representation*. Then the label can be predicted based on $h^{\mathcal{M}}$. Hence, concerning $h^{\mathcal{M}}$, the NLU process can be reformulated as: $\mathcal{M} : X \rightarrow h^{\mathcal{M}} \rightarrow Y$.

However, the predictive information within X is actually composed of two components: the semantic information S that decides the value of label, and the dataset biases B that only correlative to the value of label (Tsipras et al., 2018; McCoy et al., 2019; Pearl, 2009). The dataset biases could range from simple lexical overlap (Gururangan et al., 2018; Poliak et al., 2018), to complex language stylistic patterns (Zellers et al., 2019; Nie et al., 2020). As Figure 2 (a) shows, since the semantic information decides the value of labels, there is a *causal relationship* between S and Y. Furthermore, such relationship would keep valid upon different instances across different datasets (Pearl et al., 2000; Pearl, 2009). Formally, such invariance can be characterized as:

$$P(Y_{\mathcal{D}_i}|S_{\mathcal{D}_i}) = P(Y_{\mathcal{D}_j}|S_{\mathcal{D}_j}), \quad (1)$$

where $\mathcal{D}_i = \{X_{\mathcal{D}_i}, Y_{\mathcal{D}_i}\}$ and $\mathcal{D}_j = \{X_{\mathcal{D}_j}, Y_{\mathcal{D}_j}\}$ are arbitrary two NLU datasets, $S_{\mathcal{D}_i}$ denotes the semantic information within $X_{\mathcal{D}_i}$.

On the contrary, the correlation relationship between the dataset biases with the label would vary across different datasets, i.e.,:

$$P(Y_{\mathcal{D}_i}|B_{\mathcal{D}_i}) \neq P(Y_{\mathcal{D}_j}|B_{\mathcal{D}_j}), \quad (2)$$

where $B_{\mathcal{D}_i}$ is the dataset biases within $X_{\mathcal{D}_i}$. Therefore, having or not causal invariance forms the essential difference between semantic information and bias information.

In order for $\mathcal{M}$ to capturing the necessary semantic information within X without involving the bias information, $h^{\mathcal{M}}$ should satisfy a *causal invariant constraint*:

$$\forall X_i \in \mathcal{T}: \; P^{do(X=X_i)}(Y|h^{\mathcal{M}}) = P^{do(X=X_i)}(Y|X)$$
$$P(Y_{\mathcal{D}_i}|h^{\mathcal{M}}_{\mathcal{D}_i}) = P(Y_{\mathcal{D}_j}|h^{\mathcal{M}}_{\mathcal{D}_j}) \quad (3)$$

where $\mathcal{T}$ denotes the training set, $P^{do(X=X_i)}$ denotes the distribution arising from assigning X to

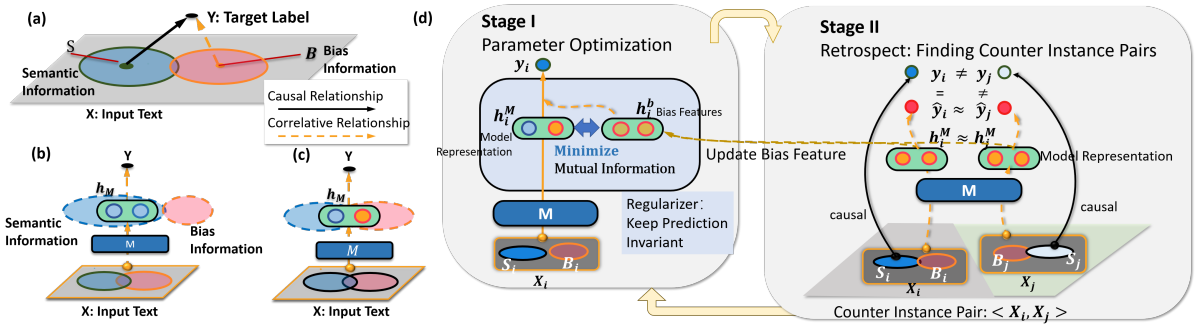

Figure 2: (a) The predictive information within the input text is composed by semantic information and bias information. (b) Ideally, a stable NLU model should avoid capturing the bias information. (c) Models empirically trained cannot distinguish causal with correlative and hence would inevitable involve bias information into model representation. (d) Debiasing process of the CI-sNLU framework.

be a certain instance X_i (Pearl, 2009). This means, (1) $h^{\mathcal{M}}$ contains the semantic information that decides the value of label Y. (2) Such causal relationship exist in **every** instance $X_i \in \mathcal{T}$, and keep valid for arbitrary OOD datasets. Hence, training a model $\mathcal{M}$ satisfying the causal invariant constraint is the goal of stable NLU, as shown in Figure 2 (b).

However, for models that only have access to the input text-target label pairs, they have no information to distinguish the causal relationship from the correlative relationship. Hence, with the existence of the bias in the dataset, such models would inevitably capture the dataset bias correlative to the target label (Figure 2 (c)). Which makes $h^{\mathcal{M}}$ contain both the semantic information S and the biases B and leads to the instability. Debiasing methods have been proposed to mitigate the influence of biases. However, due to the absence of causal mechanism in these debiasing methods, their ability for distinguishing the bias information from the semantic information would still be limited.

To address this issue, we propose a **C**ausal **I**nvariance based **s**table **NLU** (CI-sNLU) framework. CI-sNLU employs criteria deduced from the causal invariance constraint to identify the bias information that the model captured, and regularizes model to exclude the identified bias information by enforcing $h^{\mathcal{M}}$ to obey the constraint.

3 Methodology

3.1 Causal Invariance Based Biases Identification

The causal invariance bias feature identification algorithm dynamically identifies the biases captured by the model during the training process. The core assumption of the algorithm is that the semantic representation of instances will obey the causal invariance constraint, while the bias features correlative to the label does not. Hence, if we can discover the instances on which the causal invariance constraint is violated, then it could be probable to identify the bias features from the model representations of these instances.

In specific, as described in the causal invariance constraint (Eq. 3), if the model can capture the semantic information causal to the label and obtain representation $h^{\mathcal{M}}$, then for two instances X_i and X_j, if their representation $h_i^{\mathcal{M}}$ and $h_j^{\mathcal{M}}$ are rather similar, then $P(Y_i|\mathcal{M}_i)$ should be rather close to $P(Y_j|\mathcal{M}_j)$. Therefore, if we can find instance pairs $\langle X_i, X_j \rangle$, on which $h_i^{\mathcal{M}}$ is rather close to $h_j^{\mathcal{M}}$, whereas $Y_i \neq Y_j$, then these examples can be regarded as counter instances that violates the causal invariance constraint, and can be utilized for detecting the bias information $\mathcal{M}$ utilizes. For clarity, we define such instance pairs $\langle X_i, X_j \rangle$ as a *counter instance pair*:

Definition 1 (Counter Instance Pair): $\forall \{X_i, Y_i\}, \{X_j, Y_j\} \in \mathcal{T}, i \neq j$, if:

$$S(h_i^{\mathcal{M}}, h_j^{\mathcal{M}}) > \tau, \; s.t. \; Y_i \neq Y_j \wedge (\hat{Y}_i = Y_i \vee \hat{Y}_j = Y_j) \quad (4)$$

where $S(\cdot)$ is a score function measuring the similarity between $h_i^{\mathcal{M}}$ and $h_j^{\mathcal{M}}$, τ is a threshold controlling the confidence that $h_i^{\mathcal{M}}$ and $h_j^{\mathcal{M}}$ can been deemed as the representation of the same semantic information. $\hat{Y}_i, \hat{Y}_j$ are model prediction of Y_i and Y_j, respectively. Note that, the additional condition $\hat{Y}_i = Y_i \vee \hat{Y}_j = Y_j$ requires that model should make a correct prediction for X_i or X_j. This is because, if $\mathcal{M}$ fails to correctly predict the label on both X_i and X_j, then it is more likely that $\mathcal{M}$ has not captured the predictive information in X_i and X_j, rather than using the bias information.

To dynamically detect counter instance pairs within $\mathcal{T}$ during the training process, as Figure 2 (d)

shows, we divide the training process into two alternate stages: a parameter optimization stage and a retrospect stage. In the parameter optimization stage, we train $\mathcal{M}$ to find new predictive features from data, meanwhile regularize $\mathcal{M}$ to exclude the known biases using a causal invariance based regularizer (which is described in the following section). In the retrospect stage, CI-sNLU finds the counter instance pairs using the model representations, to discover the bias features that model temporarily captures and update the bias features.

Parameter Optimization Given training dataset $\mathcal{T} = \{X_i, Y_i\}$, we update the parameters of the NLU model $\mathcal{M}$ through maximizing the likelihood. Meanwhile $\mathcal{M}$ is regularized using a regularizer $\mathcal{R}(\mathcal{M}, h_t^b)$ to exclude the bias components from fitting the bias information, where $h_t^b = \{h_{i,t}^b\}$ is a d-dimension random variable, characterizing the distribution of the dataset biases on the training set $\mathcal{T}$, $h_{i,t}^b$ is the temporal bias feature of sample i at tth step. h_0^b is initialized using the previous automatic debiasing methods (e.g., the method of Utama et al. (2020b)), and updated in the retrospect stage of the training process.

Retrospect After t steps of parameter optimization, we examine whether all model representation of each instance $\{X_i, Y_i\} \in \mathcal{T}$ satisfy the causal invariance constraint, to find the representation of instances that may contains bias information, and update h^b. In specific, for each instance $\{X_i, Y_i\} \in \mathcal{T}$, we utilize $\mathcal{M}$ to obtain corresponding temporal model representations $h_{i,t}^{\mathcal{M}}$, and predictions of the label $\hat{Y}_{i,t}$, together with a probability $\hat{p}_{i,t} = P(\hat{Y}_{i,t}|h_{i,t}^{\mathcal{M}})$.

Then We traverse $\mathcal{T}$ to find all the counter instance pairs according to the definition described in Eq.4. In practice, we implement $S(\cdot)$ using the cosine similarity function, and introduce two more conditions to enhance the confidence of counter instance pairs detection, i.e., for $X_i, X_j \in \mathcal{T}$, to be the counter instance pairs, we further require that:

$$\hat{p}_{i,t} > \tau_p, \;\; \hat{p}_{j,t} > \tau_p \tag{5}$$

where τ_p is a threshold for filtering out the predictions that have low confidence. With this additional threshold, we can further control the confidence on the discovered bias feature. Note that, the counter instance pairs are found at the tth parameter optimization step. Hence, we can dynamically detect the bias feature temporarily captured by the model for updating the bias feature distribution, rather than only relying on fixed bias features.

Hence, for each sample $X_i \in \mathcal{T}$, at the tth step, its bias features $h_{i,t}^b$ can be updated using the representation of corresponding counter instances as:

$$h_{i,t}^b = h_{i,t_0}^b + \alpha \frac{1}{|\mathcal{N}_{\mathcal{C}_i}|} \sum_{j \in \mathcal{N}_{\mathcal{C}_i}} h_{j,t}^{\mathcal{M}} \tag{6}$$

where h_{i,t_0}^b is the bias feature of the i th sample previously obtained after the t_0th parameter optimization step, $\mathcal{N}_{\mathcal{C}_i}$ is a set composed by all counter instances of X_i, $|\mathcal{N}_{\mathcal{C}_i}|$ is the size of $\mathcal{N}_{\mathcal{C}_i}$, α is a coefficient controlling the proportion of bias feature update.

After the retrospect stage, following parameter stage is conducted with $\mathcal{M}$ regularized using the updated bias features $h_{i,t}^b$.

3.2 Causal Invariance Based Model Regularization

In the parameter optimization stage, we employ an causal invariance based regularizer to exclude the bias information model captured by forcing the model $\mathcal{M}$ to fulfilling the causal invariance constraint. As described in Eq. 3, the prediction should keep invariant under different kinds of bias features. A sufficient condition is that the information model representation $h^{\mathcal{M}}$ contains keeps invariant under different biases. In other words, manipulating the value of bias feature does not influence information contained within $h^{\mathcal{M}}$. The information within $h^{\mathcal{M}}$ can be characterized using its information entropy $H(h^{\mathcal{M}})$. Hence, formally,

$$\begin{aligned} &\forall h_i^b, h_j^b \in h^b; \forall X_i \in X, \\ &H^{do(h^b = h_i^b)}(h_i^{\mathcal{M}}|X_i) = p^{do(h^b = h_j^b)}(h_i^{\mathcal{M}}|X_i) \end{aligned} \tag{7}$$

One way to ensure such invariance is making the model representation $h^{\mathcal{M}}$ independent to the bias feature h^b, i.e, $h^{\mathcal{M}} \perp\!\!\!\perp h^b$. The independence makes the mutual information between h^b and h^b becomes 0, i.e., $I(h^{\mathcal{M}}, h^b) = 0$. Hence, $H(h^{\mathcal{M}}|h^b) = H(h^{\mathcal{M}}) - I(h^{\mathcal{M}}, h^b) = H(h^{\mathcal{M}})$. In other words, the value of bias will not influence the information contained in $h^{\mathcal{M}}$.

Therefore, at an arbitrary parameter optimization step t, if we can calculate the temporary mutual information $I(h_t^{\mathcal{M}}, h_t^b)$ and encourage it to be zero, then we can reduce the proportion of bias information contained in model representation $h^{\mathcal{M}}$ to increase the stability of model. However, in general, the precise value of mutual information $I(h_t^{\mathcal{M}}, h_t^b)$ for two random variables with complex

distribution are hard to calculate. To address this issue, we resort to approximations.

With information theory and the Law of Large Numbers (Cramér, 2016; Rao, 1992), we find that, in the training process, by predicting the label as:

$$\hat{Y}_i = \sigma(W h^z_{i,t}); \quad (8)$$

$$h^z_{i,t} = (h^{\mathcal{M}}_{i,t} + h^b_{i,t}) \quad (9)$$

where $\hat{Y}_i$ is the predicted label of sample i, $\sigma(\cdot)$ is a sigmoid function, $W \in \mathbb{R}^{d \times d}$ is a weight matrix, and then training $\hat{Y}_i$ to close to the ground truth label Y_i, **theoretically**, we can approximately make $I(h^{\mathcal{M}}_t, h^b_t)$ **achieving its lower bound 0**. We show the specific proving process in the Appendix. For clarity, we denote this regularizer as ADD.

4 Experiments

4.1 Evaluation Tasks

We evaluate our approach on three NLU tasks: natural language inference (NLI), fact verification, and paraphrase identification. We compare the in-distribution performance on the test set of each task. Then examine the stability of model on OOD samples by comparing the **zero-shot** performance on the corresponding challenge dataset. On the NLI and fact verification task, model performance is evaluated using prediction accuracy. Following Devlin et al. (2019) and Radford et al. (2018), on the Paraphrase Identification task, we evaluate model performance using the F1 score.

Natural Language Inference This task requires the model to predict the semantic entailment relationship between a premise and a hypothesis. We use the MNLI dataset (Williams et al., 2018) as the benchmark, and use corresponding challenge HANS McCoy et al. (2019) to test the stability on OOD samples. Since HANS is built by removing the lexical overlap bias that extensively exists in the MNLI dataset, models trained on MNLI often perform close to a random baseline on HANS.

Fact Verification This task requires a model to predict whether a claim can be supported or refuted by corresponding evidences. We train model on the Fever dataset (Thorne et al., 2018), and evaluate the stability of models on the FeverSymmetric V 0.1 (Schuster et al., 2019) dataset, which is collected to remove the claim-only biases (i.e., the biases within the claims which make models able to make predictions without evidence).

Paraphrase Identification We conduct experiments on the QQP dataset[1], which consists of 362K questions pairs annotated as either duplicate or non-duplicate, together with the corresponding challenge dataset PAWS (Zhang et al., 2019b), which is constructed by removing the lexical overlap biases within the QQP dataset.

4.2 Experimental Details

On all three tasks, we implement the main model $\mathcal{M}$ using the BERT-base model (Devlin et al., 2019), and regularize $\mathcal{M}$ with the ADD regularizer. The biased feature of each example h^b_i is initialized using the automatic debiasing method of (Utama et al., 2020b), which employs a BERT-base model trained upon a tiny subset of the original training set to capture the biased information.

During the training process, the biased feature detecting algorithm detects and updates the biased features at the start of the 2nd to last epoch. Before fed into the model, each example is preprocess into a [CLS] premise [SEP] hypothesis / [CLS] claim [SEP] evidence form, where [CLS] and [SEP] are two special tokens (Devlin et al., 2019). Then we employ the embedding vector of the [CLS] token at the top transformer layer as the model representation $h^{\mathcal{M}}$ of each instance for finding the counter instances pairs. To increase the confidence of detected biased feature, on all three datasets, we set $\tau_p = 0.95$, and $\tau = 0.9$. The information update coefficient is set as $\alpha = 0.5$. We report the average result across 5 runs. More details about the hyperparameter selection and time-costing are provided in the Appendix.

4.3 Baseline Methods

We make comparisons with:

(i) BERT (Devlin et al., 2019) refers to the BERT-base model trained upon each NLU dataset without debiasing process.

Prior-knowledge-depended Debiasing Methods

These methods rely on prior knowledge about the distribution of dataset biases to detect the biased instances, then regularize model by down-weighting the biased instances, so that the main model can focus on learning from harder examples. The major difference resides in how the main model is regularized.

(ii) Known-bias$_{\text{Reweighting}}$ (Clark et al., 2019; Schuster et al., 2019) weights the importance of a instance using the probability that the instance exhibits a bias. **(iii) Known-bias$_{\text{PoE}}$** (Clark et al.,

[1] https://data.quora.com

Method	MNLI	HANS	Δ	Fever	symm.	Δ	QQP	PAWS	Δ
Bert-base	**84.5**	61.5	-	85.6	55.7	-	**87.9**	48.7	-
Known-bias $_{\text{Reweighting}}$	83.5	69.2	+7.7	84.6	61.7	+6.0	85.5	49.7	+1.0
Known-bias $_{\text{POE}}$	82.9	67.9	+6.4	86.5	60.6	+4.9	84.3	50.3	+1.6
Known-bias $_{\text{Conf-reg}}$	84.5	69.1	+7.6	86.4	60.5	+4.8	85.0	49.0	+0.3
Shallow Model Debiasing $_{\text{Reweighting}}$	82.3	69.1	+7.6	87.2	60.8	+5.1	79.4	46.4	-2.3
Shallow Model Debiasing $_{\text{POE}}$	82.7	69.8	+8.3	85.4	60.9	+5.2	80.7	47.4	-1.3
Shallow Model Debiasing $_{\text{Conf-reg}}$	83.9	67.7	+6.2	**87.9**	60.4	+4.7	83.9	49.2	+0.5
Weak Learner Debiasing	83.3	67.9	+6.4	85.3	58.5	+2.8	-	-	-
LGTR	84.4	58.0	-3.5	85.5	57.9	+2.2	-	-	-
CI-sNLU	83.2	**73.1**	**+11.6**	85.0	**63.9**	**+8.2**	85.5	**51.3**	**+2.6**
CI-sNLU+bias	**84.5**	-	-	84.8			86.2		

Table 1: Model performance (MNLI / Fever: accu. (%); QQP: F1) on in-distribution and corresponding challenge instances.

2019) forces the main model to focus on learning from examples that are not predicted well by the biased model. **(iv) Known-bias$_{\text{Conf-reg}}$** (Utama et al., 2020a) regularizes the main model by decreasing model confidence on biased examples.

Auto-Debiasing Methods

(v) Shallow Model Debiasing (Utama et al., 2020b) trains a BERT-base model on a small subset of the training set to obtain biased features characterizing dataset biases distribution. **(vi) Weak Learner Debiasing** (Sanh et al., 2020) employs a Tiny-BERT model (Turc et al., 2019) as a weak learner to obtain the biased features. **(vii) LTGR** (Du et al., 2021) employs a teacher model to regularize the main model from capturing bias and encourage it to learn long-tailed features.

In this paper, all the baseline debiasing methods take the BERT-base model as the main model.

4.4 Main Results

From Table 1 we observe that:

(1) Comparing the automatic debiasing methods with the prior knowledge based debiasing methods show that, in general, there is still a performance gap between automatic and prior-knowledge based debiasing methods. This is because the distribution of dataset bias can be rather complex, which leads to challenges in precisely and comprehensively detecting the potential biases, and makes automatic debiasing still remaining a challenging problem. However, comparing with the prior-knowledge based debiasing methods, our approach can have better or comparable performance on all three challenge datasets and can have comparable in-distribution performance. This indicates the effectiveness of our approach.

(2) Comparing to the Shallow Model Debiasing and the Weak Learner Debiasing which rely on pre-detected fixed bias features, CI-sNLU can consistently improve model performance on all three challenge datasets. This indicates that, by dynamically detecting the bias information and regularizing model to follow the causal invariance constraint, our approach can effectively increase the stability of model performance.

Model	MNLI	HANS
CI-sNLU	**83.2**	**73.1**
-w/o den	81.2	72.9
-w/ POE	81.7	71.3
- -iden + POE	79.4	67.3

Table 2: Results of ablation studies.

(3) In general, improvements on the challenge datasets come with the expense of the in-distribution performance. This is because, the bias provides additional clues that leak the label information (Zhang et al., 2019a; Tsipras et al., 2018; Sanh et al., 2020). Hence, omit of the bias would naturally lead to a performance decrease on in-distribution samples. Different from previous methods, as described in Eq.(8-9), CI-sNLU can also combine the bias features to make predictions, to accommodate the situation where only in-distribution performance is desired.

4.5 Ablation Study

We conduct ablation study by training CI-sNLU without the causal invariance based feature identification (denoted as CI-sNLU -w/o iden) and substituting the `ADD` regularizer by a POE regularizer (denoted as CI-sNLU -w POE), as POE regularizer show strong performance across multiple datasets and hence can be a strong baseline. The results are shown in Table 2, where -iden + POE represents the combination of -w/o iden and -w POE.

We have the following observations: (1) Compared to CI-sNLU -w/o iden, the vanilla CI-sNLU has better performance on the challenge set. This indicates that based on the causal invariance constraint, our approach can effectively detect the bias

Model	ANLI-R1	R2	R3
BERT-base	0	28.9	28.8
Shallow Model Debiasing	26.4	29.5	29.3
CI-sNLU	**27.2**	**29.5**	**31.5**

Table 3: Zero-shot performance on target datasets.

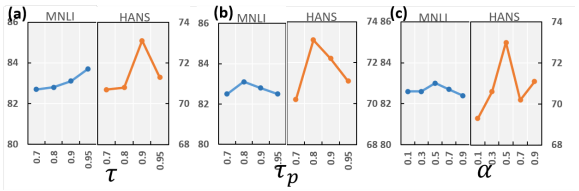

Figure 3: Model performance (Accu.(%)) under different choice of hyperparameters.

information model used during the training process, to enhance the effectiveness of debiasing. (2) Compared to CI-sNLU -w POE, CI-sNLU has better performance. This is because, the causal invariant based regularizer can minimize the mutual information between the bias features and model representations to zero, while POE cannot precisely control the amount of mutual information between $h^{\mathcal{M}}$ and h^b, so that the model representation may still contain bias information. (3) The large performance gap between CI-sNLU and CI-sNLU -iden + POE indicates the synergetic relationship between bias feature identification and model regularizing.

4.6 Transferability Analysis

We further examine the stability of our approach through a transferability analysis. In specific, we train CI-sNLU on the MNLI dataset, then evaluate its zero-shot performance on three challenge sets ANLI R1-R3 (Nie et al., 2020). ANLI R1-R3 contains instances designed **to fool the model to make wrong predictions by human edition on input text**. Hence, to make correct predictions, models have to understand the semantic of input. Models utilizing bias information always have a zero-shot performance close to 0. The reason for not adopting other NLI datasets is that, different NLI datasets could probably share similar dataset bias patterns (McCoy et al., 2019; Geva et al., 2019; Du et al., 2021). Hence, it would be hard to distinguish the performance improvement brought by utilizing the same bias pattern, or by promotion in the understanding of the semantic information. Two baselines are involved for comparison: BERT-base, and Shallow Model Debiasing.

The results are shown in Table 3. We observe that: (1) The BERT-base model has poor performance on all three target tasks, especially on the ANLI R1 dataset, as it is specifically designed to fool the BERT model to make its performance close to 0. This suggests that BERT may utilize a large amount of bias features for making predictions. (2) Shallow Model Debiasing and CI-sNLU can enhance model performance on all three target datasets, indicating the effectiveness of automatic debiasing methods in mitigating the influence of dataset bias to improve model stability. (3) Compared to Shallow Model Debiasing, our approach can further increase the model performance on all three target datasets, and has more consistent performance. This suggests that by introducing the causal invariance constraint, CI-SNLU can better detect the bias information model used and regularize model to further increase the stability.

4.7 Sensitivity Analysis

In the bias feature identification process, we employ two hyperparameters τ_p and τ to control the confidence of the identified bias feature, and a hyperparameter α to control the proportion of bias feature update. We investigate the sensitivity of model performance upon these hyperparameters by changing the value of one hyperparameter and fixing the value of the other three hyperparameters, then observe the model performance. Experiments are conducted on MNLI and HANS. The results are shown in Figure 3.

From which we observe that: (1) Empirically, the performance of CI-sNLU keep relative stable with a wide range of hyperparameter values, for example, when $\tau_p > 0.8$, $\tau > 0.9$ and $\alpha \in [0.3, 0.7]$. This indicates the robustness of our approach on hyperparameter settings. (2) The change of hyperparameters can lead to a trade-off between the performance on in-distribution samples and OOD samples. This is because, with lower τ_p, more examples can find corresponding counter instances. With lower τ, a given example can match up with more counter instances. Hence, more abundant potential bias features can be identified. However, with a lower confidence on identified bias feature, some of the semantic information would be mistaken as bias information. This would lead to a performance decrease on the in-distribution samples. On the contrary, with τ and τ_p close to 1, the identification of new bias features would be strictly controlled, which impact the effectiveness of debiasing and influence the performance on the challenge set. On the other hand, too much or less bias feature updates would both harm the perfor-

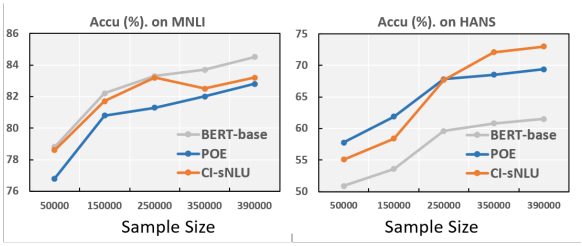

Figure 4: Influence of sample size on model performance.

mance. Hence, a moderate value of τ, τ_p, and α is still necessary to achieve balanced performance in both in-distribution and OOD samples.

4.8 Influence of Sample Size

In this paper, we detect the bias features by finding counter instance pairs. However, the probability that proper counter instances can be found would also be affected by the size of the training set, as if the training set is not large enough, the probable counter instances would not be covered by the dataset. Hence, we examine the influence of sample size on effectiveness of debiasing. In specific, we randomly sampled several new training sets containing $\{50K, 150K, 250K, 350K\}$ samples, respectively, from the original MNLI dataset. Then examine the performance of our approach together with BERT and Shallow-Model Debiasing on the dev set of MNLI and HANS. The results are shown in Figure 4.

From which we observe that, as the size of the training set increases, the performance on in-distribution dev set and challenge set consistently increases. This is because, on the one hand, a larger training set can provide more semantic information. On the other hand, as the size of the training set increases, the probability that the model can find corresponding counter instance also increases. Furthermore, the relationship between training set size and performance on the challenge set shows that, different from Shallow Model Debiasing, CI-sNLU can further increase the performance of debiasing once the training set of MNLI is further enlarged.

5 Related Work

The NLU task requires model to understand the underlying semantic information. However, the existence of dataset biases allows model to complete the task without learning the intended reasoning skill (Gururangan et al., 2018; McCoy et al., 2019; Belinkov et al., 2019). This phenomenon exist in various different tasks, such as reading comprehension (Kaushik et al., 2019), question answering (Mudrakarta et al., 2018), and fact verification (Schuster et al., 2019). To better evaluate the reasoning ability of models, researchers constructed challenge datasets composed of "counterexamples" to the biases that models may adopt (McCoy et al., 2019; Schuster et al., 2019; Naik et al., 2018). Model performances always have a significant decline on these challenge sets.

One line of debiasing methods mitigates the dataset biases based on human prior knowledge. Based on the human intuitions on task-specific biases, Schuster et al. (2019); Clark et al. (2019); He et al. (2019) detect the biased examples and down-weight these samples, while Min et al. (2020); Belinkov et al. (2018) explicitly modify the dataset distribution by data augmentation. However, these methods are limited by their assumption that the task-specific biases should be available a priori. To address this issue, automatic debiasing methods are proposed to detect the dataset biases without dependency on prior knowledge. For example, Utama et al. (2020b) automatically capture the dataset bias by training a shallow model on a tiny training set, while Sanh et al. (2020) capture the dataset bias using a learner with limited capacity. However, researches indicate the difference between the biases shallow model or weaker learner captured, and that the main NLU model captured (Kaushik et al., 2019). More crucially, without the incorporation of causal assumptions, the debiasing methods still cannot distinguish the causation with correlation, which limits the effectiveness of debiasing.

In this paper, we propose a Causal Invariance based stable NLU framework. With the causal invariance constraint, CI-sNLU can effectively detect the bias feature model utilized and regularize model to exclude the correlative component to increase the stability of model performance.

6 Conclusion

In this paper, we propose a Causal Invariance based stable NLU framework. By introducing a causal invariance constraint into the debiasing process, we can dynamically detect the bias information model captured, then regularize the model to exclude the bias components temporarily captured by the model to enhance the stability. Experimental results show that our approach can significantly increase stability on out-of-distribution samples compared to previous methods.

References

Yonatan Belinkov, Yonatan Bisk, and B A. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. 2019. Dont 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.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Dont 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.

Harald Cramér. 2016. *Mathematical Methods of Statistics (PMS-9), Volume 9*. Princeton university press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of nlu models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929.

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.

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.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906.

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.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

C Radhakrishna Rao. 1992. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in statistics*, pages 235–247. Springer.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2020. 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.

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

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

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. *arXiv preprint arXiv:2009.12303*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019a. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308.

7 Appendix

7.1 Theoretical analysis of the regularizer

During the parameter optimization stage, we devise a causal invariance based ADD regularizer to exclude the bias information model captured. We argue that, theoretically, this regularizer can approximately minimize the mutual information between the model representation $h^{\mathcal{M}}$ and the bias feature distribution h^b to lower bound 0 at each training step. Here is the specific proof process.

Let $I(h_t^{\mathcal{M}}, h_t^b)$ denote the mutual information between $h^{\mathcal{M}}$ and h^b at the tth parameter optimization step. We notice that, to make $I(h_t^{\mathcal{M}}, h_t^b) = 0$, a sufficient condition is that, for each instance $i \in \mathcal{T}$, $I(h_{i,t}^{\mathcal{M}}, h_{i,t}^b) = 0$, where $h_{i,t}^{\mathcal{M}}$ and $h_{i,t}^b$ is the model representation and bias feature for *sample i*. Note that, here we regard $h_{i,t}^{\mathcal{M}}$ and $h_{i,t}^b$ as two random variables, as both $h_{i,t}^{\mathcal{M}}$ and $h_{i,t}^b$ are estimated from data, hence are not certain values and with uncertainty.

However, in general, the precise value of mutual information $I(h_{i,t}^{\mathcal{M}}, h_{i,t}^b) = 0$ for two random variable with complex distribution are hard to calculate. To address this issue, we resort to approximations.

We start from modeling the distribution of $h_{i,t}^{\mathcal{M}}$ and $h_{i,t}^b$. Cramér (2016) and (Rao, 1992) have proved that, under specific regularity conditions, if a random variable is estimated through maximizing the likelihood function, then it should asymptotically converge to a normal distribution. In this paper, since we initialize $h_{i,t}^b$ from the methods of Utama et al. (2020b) which captures bias features by maximizing the likelihood, and the model representation $h^{\mathcal{M}}$ is optimized using maximizing the likelihood function, hence both the bias feature distribution $h_{i,t}^b$ and the model representation $h_{i,t}^{\mathcal{M}}$ would asymptotically converge to normal distribution:

$$h_{i,t}^{\mathcal{M}} \to N(\mu_{i,t}^{\mathcal{M}}, \Sigma^{\mathcal{M}})_{i,t};\ h_{i,t}^b \to N(\mu_{i,t}^b, \Sigma_{i,t}^b) \tag{10}$$

where $\mu_{i,t}^b, \mu_{i,t}^{\mathcal{M}} \in \mathbb{R}^d$ is the mean of the corresponding distributions, $\Sigma^{\mathcal{M}})_{i,t}, \Sigma_{i,t}^b \in \mathbb{R}^{d\times d}$ is the covariance matrix of corresponding distributions.

Furthermore, for arbitrary two normal distributed random variables a_1 and a_2, their mutual information has a close form:

$$I(a_1, a_2) = -\frac{1}{2}\ln\left(\frac{\mathrm{Det}(\Sigma_z)}{\mathrm{Det}(\Sigma_{a_1})\mathrm{Det}(\Sigma_{a_2})}\right) \quad (11)$$

where $z = p(a_1, a_2)$ is the joint distribution of a_1 and a_2; Det is the determinant operator.

Hence, if we can: (1) derive the joint distribution of $h^b_{i,t}$ and $h^{\mathcal{M}}_{i,t}$, which we denote as $h^z_{i,t} = p(h^b_{i,t}, h^{\mathcal{M}}_{i,t})$; (2) obtain $\Sigma^z_{i,t}$, $\Sigma^{\mathcal{M}})_{i,t}$, and $\Sigma^b_{i,t}$, then we can calculate the mutual information $I(h^{\mathcal{M}}_t, h^b_t)$.

For the first problem, notice that the joint distribution $h^z_{i,t} = p(h^b_{i,t}, h^{\mathcal{M}}_{i,t})$ is a combination of the information of $h^b_{i,t}$ and $h^{\mathcal{M}}_{i,t}$.Hence, we can get an approximation of $h^z_{i,t}$ by integrating $h^b_{i,t}$ and $h^{\mathcal{M}}_{i,t}$ in various ways. For example, integrating $h^b_{i,t}$ and $h^{\mathcal{M}}_{i,t}$ using a neural network as $h^z_{i,t} = NN(h^b_{i,t}, h^{\mathcal{M}}_{i,t})$, or by directly adding $h^b_{i,t}$ and $h^{\mathcal{M}}_{i,t}$ as $h^z_{i,t} = (h^b_{i,t} + h^{\mathcal{M}}_{i,t})$.

To calculate the covariance matrices, interestingly, Cramér (2016) and Rao (1992) have also proven that, during the training process, if a group of parameters h is estimated by maximizing the likelihood, as the value of h converge to its expectation $\mathbb{E}(h)$, the covariance matrix of h asymptotically converge to the expectation of the square of the partial derivative, i.e.:

$$\Sigma_h \to \mathbb{E}\frac{\partial Y}{\partial h}^2 \quad (12)$$

where Y is the model output.

Hence, we can get the expectation of $\Sigma^z_{i,t}$, $\Sigma^{\mathcal{M}})_{i,t}$, and $\Sigma^b_{i,t}$ based on the partial gradients on $h^z_{i,t}$, $h^{\mathcal{M}}_{i,t}$, and $h^b_{i,t}$ the at the tth parameter optimization step, i.e.:

$$\Sigma^z_{i,t} = \mathbb{E}(\frac{\partial Y_i}{\partial h^z_{i,t}}^2) \quad (13)$$

$$\Sigma^{\mathcal{M}}_{i,t} = \mathbb{E}(\frac{\partial Y_i}{\partial h^{\mathcal{M}}_{i,t}}^2) \quad (14)$$

$$\Sigma^b_{i,t} = \mathbb{E}(\frac{\partial Y_i}{\partial h^b_{i,t}}^2) \quad (15)$$

In practical, we use the sample partial gradient to estimate the expectation of covariance matrix as:

$$\hat{\Sigma}^z_{i,t} = (\frac{\partial Y_i}{\partial h^z_{i,t}}^2) \quad (16)$$

$$\hat{\Sigma}^{\mathcal{M}}_{i,t} = (\frac{\partial Y_i}{\partial h^{\mathcal{M}}_{i,t}}^2) \quad (17)$$

$$\hat{\Sigma}^b_{i,t} = (\frac{\partial Y_i}{\partial h^b_{i,t}}^2) \quad (18)$$

where $\hat{\Sigma}^z_{i,t}$, $\hat{\Sigma}^{\mathcal{M}}_{i,t}$, $\hat{\Sigma}^b_{i,t}$ is the expectation of $\Sigma^z_{i,t}$, $\Sigma^{\mathcal{M}})_{i,t}$, and $\Sigma^b_{i,t}$, respectively.

By substituting Eq. 16 17 18 into Eq. 11, we can get the estimation of $I(h^{\mathcal{M}}_{i,t}, h^b_{i,t})$.

Furthermore, we prove that, by training model using the ADD regularizer which is formulated as:

$$\hat{Y}_i = \sigma(W h^z_{i,t});$$
$$h^z_{i,t} = (h^{\mathcal{M}}_{i,t} + h^b_{i,t}) \quad (19)$$

where $\sigma(\cdot)$ is a sigmoid function, $W \in \mathbb{R}^{d\times d}$ is a weight matrix. Then we can make $I(h^{\mathcal{M}}_{i,t}, h^b_{i,t})$ achieving its lower bound 0.

In specific, usingnder the formalization in Eq. 19, the partial derivatives upon $h^z_{i,t}$, $h^{\mathcal{M}}_{i,t}$ and $h^b_{i,t}$ equal:

$$\hat{\Sigma}^z_{i,t} = (\frac{\partial Y_i}{\partial h^z_{i,t}}^2) = (\hat{Y}_{i,t} - Y)^T W \quad (20)$$

$$\hat{\Sigma}^{\mathcal{M}}_{i,t} = (\frac{\partial Y_i}{\partial h^{\mathcal{M}}_{i,t}}^2) = (\hat{Y}_{i,t} - Y)^T W \quad (21)$$

$$\hat{\Sigma}^b_{i,t} = (\frac{\partial Y_i}{\partial h^b_{i,t}}^2) = (\hat{Y}_{i,t} - Y)^T W \quad (22)$$

By substituting Eq. 20 21 22 into Eq. 11, we can have $I(h^{\mathcal{M}}, \hat{h}_b) = 0$.

By employing the ADD regularizer, we can gradually exclude the bias components within $h^{\mathcal{M}}$ to increase the model stability. Furthermore, $I(h^{\mathcal{M}}_t, h^b_t)$ is calculated based on the partial gradients, and the partial gradients can reflect the sensitivity of model prediction $\hat{Y}$ upon h_z, $h^{\mathcal{M}}$, and h^b. Hence, our regularizer is built upon the actual contribution of model representation and bias feature on model prediction at each parameter optimization step, rather than using fixed weights to regularize $\mathcal{M}$ as previous researches adopts.

7.2 Experimental Details

MNLI

- batch size: 32

- number of epochs: 3
- learning rate: 5e-5
- Optimizer: Adam

Time Costing: 1 Geforce_rtx_2080_ti with 4 CPUs: ∼8h

Fever

- batch size: 32
- number of epochs: 3
- learning rate: 2e-5
- Optimizer: Adam

Time Costing: 1 Geforce_rtx_2080_ti with 4 CPUs: ∼4h

QQP

- batch size: 32
- number of epochs: 3
- learning rate: 2e-5
- Optimizer: Adam

Time Costing: 1 Geforce_rtx_2080_ti with 4 CPUs: ∼5.5h