Counterfactual Adversarial Learning with Representation Interpolation

Wei Wang[1], Boxin Wang[2], Ning Shi[1]
Jinfeng Li[1], Bingyu Zhu[1], Xiangyu Liu[1], Rong Zhang[1]
[1]Alibaba Group
[2]University of Illinois at Urbana-Champaign
{luyang.ww, shining.shi, jinfengli.ljf,
zhubingyu.zby, eason.lxy, stone.zhangr}@alibaba-inc.com
boxinw2@illinois.edu

Abstract

Deep learning models exhibit a preference for statistical fitting over logical reasoning. Spurious correlations might be memorized when there exists statistical bias in training data, which severely limits the model performance especially in small data scenarios. In this work, we introduce **C**ounterfactual **A**dversarial **T**raining framework (CAT) to tackle the problem from a causality perspective. Particularly, for a specific sample, CAT first generates a counterfactual representation through latent space interpolation in an adversarial manner, and then performs Counterfactual Risk Minimization (CRM) on each original-counterfactual pair to adjust sample-wise loss weight dynamically, which encourages the model to explore the true causal effect. Extensive experiments demonstrate that CAT achieves substantial performance improvement over SOTA across different downstream tasks, including sentence classification, natural language inference and question answering. [1]

1 Introduction

Large-scale pre-trained language models such as BERT (Devlin et al., 2019), as one of the recent breakthroughs, have revolutionized the model development paradigm in natural language processing (NLP) and improved traditional task-specific models by a large margin. Although the pre-training and fine-tuning framework has been shown to be effective in transferring the pre-learned knowledge to downstream tasks and boosting the model performance, it could be a double-edged sword if there exists statistical bias in the training dataset, especially in small data scenarios (Yue et al., 2020). Taking sentiment analysis as an example, the downstream classifier can easily mistake a certain person

name for a sentimental word if it is imbalanced distributed in positive and negative samples. Yue et al. (2020) further show that when the capacity of the model to certain semantics is strong, it will in turn strengthen this bias, in which case the large pre-trained model becomes an amplifier for spurious features.

Generally, such problem can be better solved from the perspective of causality (Zhang et al., 2020). In the context of causal inference, causation is not correlation but something more essential with the mechanism of data generation (Pearl et al., 2009). Statistical bias, also named spurious correlations, is a result of *confounder*, a variable that can influence dependent variables and independent variables simultaneously (Pearl, 2009). The study of causation aims to find the true causal effects between variables that help to unveil the true casual effect behind observation data and realize more robust inference.

Specifically, counterfactuals are one feasible way to discover the causation. Counterfactual examples are defined as the ones that minimally-different from original ones but lead to different labels (Teney et al., 2020). Recent work shows that counterfactual samples can significantly improve model generalization and boost model performance. Kaushik et al. (2019) and Teney et al. (2020) use additional human-labeled or augmented counterfactual examples to mitigate spurious correlations. Zeng et al. (2020) proposes a two-stage training approach for named entity recognition tasks for better generalization.

Although these methods gain significant improvements in model performance, they are limited in practice since they are either designed for specific tasks or requiring human-labeled samples. Additionally, no optimization is conducted on those counterfactual examples which could further improve the performance shown in our work.

Our work. We revisit the above problem in NLP

---

[1]Code is available at `https://github.com/ShiningLab/CAT.git`

domain from a causality perspective. Following the definition of counterfactual examples, our motivation is to explore the following question: *What would be the minimal intervention that alters the model output?* As a result, we propose `CAT`, an end-to-end and task-agnostic **C**ounterfactual **A**dversarial **T**raining framework during fine-tuning to introduce counterfactual representations in training stage through latent space interpolation.

Concretely, to define our problem, we first introduce Structural Causal Model (SCM) to view our problem through a casual graph that depicts the data generation mechanism. To cut off the confounder, we conduct *do-calculus* (Pearl, 1995) for causal effect adjustments. Specifically, we propose a counterfactual representation interpolation technique called `CMIX` which is a variant of *Mixup* (Zhang et al., 2018) to generate counterfactuals and approximately realize *do-calculus* in deep learning framework. For each example x in training set, `CMIX` samples a counterpart x' and generates a counterfactual representation by interpolating the representation of x and x', which is adaptively optimized by a novel Counterfactual Adversarial Loss (CAL) to minimize the differences from original ones but lead to drastic label change by definition. Finally, to connect each original-counterfactual pair, besides the traditional Empirical Risk Minimization (ERM) (Vapnik and Vapnik, 1998), We extend it to a new counterpart, i.e., Counterfactual Risk Minimization (CRM), to allow the model to adjust sample-wise loss weight dynamically so as to explore the causal effect behind data rather than simple correlations memorization.

We also extend `CAT` to other complicated tasks besides simple classification, which are rarely studied by other Mixup-based methods. Through extensive experiments on text classification, natural language inference and question answering on two SOTA baselines, BERT and RoBERTa (Liu et al., 2019), we observe consistent improvement for `CAT` in promoting the testing accuracy especially for small data in an extra-data-free manner.

Our contributions are summarized as follows:

- We investigate the problem of spurious correlations from a causality perspective which has not been widely studied in conventional statistical learning.

- We propose `CMIX` for counterfactual representation interpolation to approximate *do-*

calculus realization in deep learning framework, which is adaptively optimized by a novel Counterfactual Adversarial Loss. Moreover, we extend the traditional ERM to a novel Counterfactual Risk Minimization as a new learning principle connecting original data representations and counterfactual ones, which enables `CAT` to explore causal effects and debias the spurious correlation.

- We propose `CAT` as a general framework for various types of NLU tasks, including sentence classification, natural language inference and question answering tasks. We show that `CAT` outperforms SOTA by a large margin across different tasks particularly when data is limited.

2 Related Work

Large-scale Pre-trained Language Model. The most widely-used solution for alleviating spurious correlations is using large-scale dataset. Yang et al. (2019) and Liu et al. (2019) proposed to build the pre-trained language models utilizing even larger corpus to reduce the bias.

Data Augmentation. Data augmentation is another solution and has become a de facto technique used in state-of-the-art machine learning models. Zhang et al. (2015) performed text augmentation by replacing words or phrases with their synonyms, and recently Wei and Zou (2019) proposed more operations. Using word embedding, (Wang and Yang, 2015) tried to find a similar word for replacement. In addition, back translations (Sennrich et al., 2016) and contextual augmentation (Fadaee et al., 2017; Kobayashi, 2018) techniques have been proposed to replace target words.

Through shrinking the weight of the training data relative to L2 regularization, mixup (Zhang et al., 2018) trained a neural network on convex combinations of pairs of examples and their labels to generate new samples $(\tilde{x}, \tilde{y})$ from (x_i, y_i) and (x_j, y_j), formally as

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1 - \lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j. \end{aligned} \quad (1)$$

Followed by mixup, more works (Verma et al., 2019a,b,c; Berthelot et al., 2019; Yun et al., 2019) are proposed, mainly focusing on image-format data. Recently, such regularization techniques were brought into NLP tasks Chen et al. (2020).

Adversarial Training. Adversarial training has been proven to be an effective approach for improving the robustness of neural network models (Miyato et al., 2017; Madry et al., 2018; Tramèr et al., 2017; Shrivastava et al., 2017). Specifically, Miyato et al. (2017) applied adversarial and virtual adversarial training to text domain by applying perturbations to the word embeddings, which achieved state-of-the-art results on multiple semi-supervised and purely supervised tasks. By minimizing the resultant adversarial risk inside different regions around input samples, Zhu et al. (2019) proposed a novel adversarial training algorithm, FreeLB, that is capable of promoting higher invariance in the embedding space.

Causal Inference. Yue et al. (2020) explored estimating true causal effects on few-shot images classification, which is an intervention-based approach. Similarly, (Tang et al., 2020) focuses on long-tail data set. In addition, using counterfacutals is another way to discover the causation in training data. For example, Kaushik et al. (2019) and Teney et al. (2020) leveraged additional human-labeled counterfactual examples, while Zeng et al. (2020) proposed a two-stage training approach for named entity recognition task. These studies have been proved that counterfactual samples are more valuable in boosting model performance compared to normal samples.

3 CAT

In this section, we introduce how CAT solves the problem of spurious correlations from a causal perspective (see Figure 1). We first explicate our problem by introducing Structural Causal Model (SCM). Then we illustrate our approach by answering the following questions: (1) How can we cut off the confounder for spurious bias elimination by counterfactual representations? (2) How can we optimize our counterfactual representations? (3) How can we learn from both original and counterfactual representations to debias the spurious correlations?

3.1 Problem Definition

To explicate our problem, we first introduce SCM to depict the data generation mechanism. Every SCM can be represented as a directed acyclic graph (DAG) which can be written as:

$$\mathbf{X}_i := f_i(\mathbf{X}_{pa(i)}) + \mathbf{U}_i, i = 1, 2, ..., d, \quad (2)$$

where each $\mathbf{X}_i$ is an endogenous variable in the graph, $\mathbf{X}_{pa(i)}$ denotes the set of parent variables of $\mathbf{X}_i$, $\mathbf{U}_i$ denotes independent and identically distributed random noise, d is the number of variables, and $f_i(\mathbf{X}_{pa(i)})$ represents the direct causation from $\mathbf{X}_{pa(i)}$ to $\mathbf{X}_i$. Each $\mathbf{U}_i$ is called an exogenous variable because it is determined outside the graph.

We can use SCM to describe how our training data is generated and where spurious correlations are derived from fine-tuning process. As shown in the left part of Figure 2, $\mathbf{C}$ is the confounding variable which is the common cause of samples $\mathbf{X}_1$ and $\mathbf{X}_2$ and leads to spurious correlations between them. During fine-tuning, model $\mathbf{H}$ can easily take such training-data-specific spurious correlations as features to predict $\mathbf{Y}$, which could severely undermine model performance when inferring on test set where such correlations do not hold. In practice, $\mathbf{C}$ can be subjective bias of human annotator, the domain of data, the region where data is collected, etc.

To eliminate such spurious correlations, we conduct *do-calculus* on $\mathbf{X}_2$ (denoted as $do(\mathbf{X}_2)$), which is shown in the right side of Figure 2. This operation is realized by cutting off all edges directing to $\mathbf{X}_2$ and setting $\mathbf{X}_2$ to a certain constant $\mathbf{x}_2$ (green node in Figure 2). *Do-calculus* blocks the causal effect from $\mathbf{C}$ to $\mathbf{X}_2$ so that $\mathbf{C}$ is no longer a confounder. In general, the operation allows us to estimate the true causal effect $P(\mathbf{Y}|do(\mathbf{X}))$ instead of the correlation $P(\mathbf{Y}|\mathbf{X})$ which is the conventional machine learning objective.

3.2 CMIX

Do-calculus is a statistical tool derived from SCM for causal effect adjustments. Recent study (Zeng et al., 2020) proposed to use *Counterfactuals* as an approximate realization of *do-calculus* for neural networks. In this paper, instead of generating counterfactual examples that may require expensive human-annotation (Teney et al., 2020), we propose CMIX as a variant of mixup (Zhang et al., 2018) to generate counterfactual representations.

3.2.1 Label-free Mixup

Due to the discrete nature of textual data, it is not favored to do interpolation directly on words as mixup does on images. Given that, CMIX conducts *do-calculus* and generates counterfactual representations by interpolating the hidden states. We aim to set a new value for $\mathbf{X}$ in SCM independent of $\mathbf{C}$ so that $\mathbf{X}$ is not affected by the *confounder*. Un-
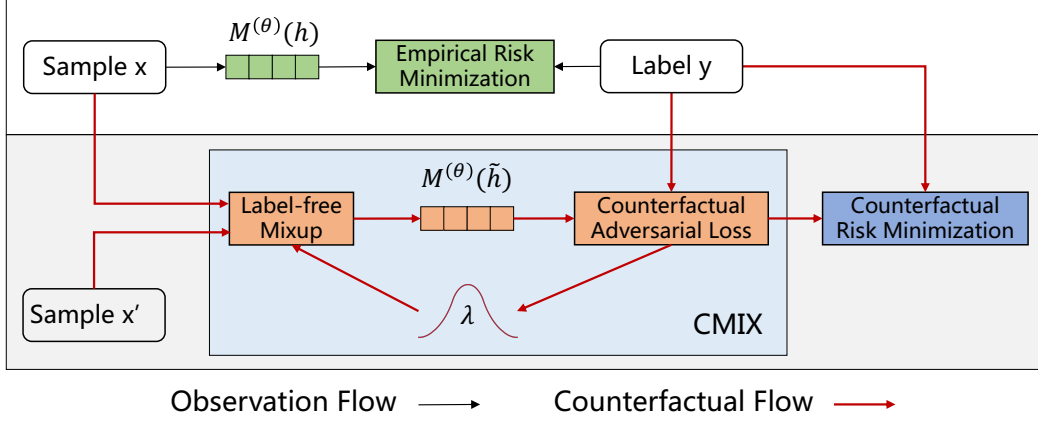
Figure 1: The framework of CAT. Besides the normal supervised ERM (Observation) flow on the top, for a certain observation x, CAT will randomly sample another x' from training data. Then a counterfactual representation $\tilde{h}$ is generated and optimized by CMIX. Finally, CRM is applied on final model output $M^{(\theta)}(\tilde{h})$.
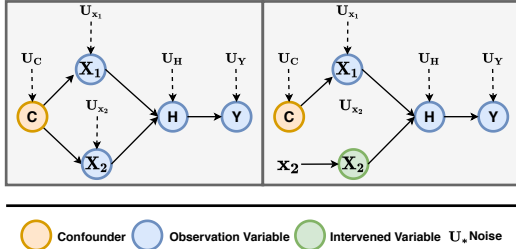

Figure 2: SCM of data generation mechanism. Left: Spurious correlations exist between $\mathbf{X}_1$ and $\mathbf{X}_2$ in observation data caused by confounder $\mathbf{C}$. Right: Confounder is eliminated by *do-calculus*.

like mixup-based methods (Chen et al., 2020) that require interpolation on labels, CMIX interpolates sample representations without label information, allowing CMIX to utilize unlabeled data.

Specifically, interpolation occurs on the hidden representations from the multi-layer transformers (Vaswani et al., 2017). Formally, we denote the multi-layer transformer model by $M^{(\theta)}$ that takes the latent representation h of x as input and output the confidence prediction $y = M^{(\theta)}(h)$. Specifically, we denote the model $M^{(\theta)}$ at the l-th layer $M_l^{(\theta)}(\cdot), l \in \{0, 1, ...L\}$, then for two samples $x^{(i)}$ and $x^{(j)}$, their corresponding latent representations $h^{(i)}, h^{(j)}$ at the l-th layer are

$$
\begin{aligned}
h_0^{(i)} &= W x^{(i)}, \quad h_0^{(j)} = W x^{(j)}, \\
h_l^{(i)} &= M_l^{(\theta)}(h_{l-1}^{(i)}), \quad l \in \{1, ..., m\}, \\
h_l^{(j)} &= M_l^{(\theta)}(h_{l-1}^{(j)}), \quad l \in \{1, ..., m\};
\end{aligned}
\tag{3}
$$

where W is the embedding look-up matrix to map the discrete sentence $x^{(i)}$ to the latent embedding h_0 as the 0-th layer representation. Then we generate the counterfactual representation $\tilde{h}^{(i)}$ by inter-

polation in m-th layer:

$$
\begin{aligned}
\tilde{h}_m^{(i)} &= \lambda^{(i)} h_m^{(j)} + (1 - \lambda^{(i)}) h_m^{(i)} \\
\tilde{h}_l^{(i)} &= M_l^{(\theta)}(\tilde{h}_{l-1}^{(i)}), \quad l \in \{m+1, ..., L\},
\end{aligned}
\tag{4}
$$

where $\lambda^{(i)}$ samples from beta distribution $Beta(\alpha, \beta)$. We discuss the choice of beta distribution in Appendices A.2.

Disscussion About Attention Mask : To address the attention mask of $\tilde{h}$ in BERT and its derivatives, which is not considered in (Chen et al., 2020), we propose several possible ways to handle the attention mask of $\tilde{h}$, including using the mask of $h^{(i)}$, $h^{(j)}$ or using the last hidden layer as the interpolation layer to avoid the use of attention mask. After experiments, we find that using the mask of $h^{(i)}$ always achieves the best performance.

3.2.2 Counterfactual Adversarial Loss

We propose Counterfactual Adversarial Loss (CAL) to further optimize the counterfactual representations $\tilde{h}$ generated by CMIX. Following the definition of Kaushik et al. (2019), CAL optimizes counterfactual representations so that they are minimally-different from the original ones $x^{(i)}$ but lead to different labels $y^{(i)}$. Specifically, we optimize the mixup parameter $\lambda^{(i)}$ by the following objective:

$$
\begin{aligned}
\underset{\lambda^{(i)}}{\arg\max} - \left\| \lambda^{(i)} \right\|_p + \gamma L(M^{(\theta)}(\tilde{h}^{(i)}), y^{(i)}) \\
+ \eta \Phi(M^{(\theta)}(\tilde{h}^{(i)})),
\end{aligned}
\tag{5}
$$

where $\|\cdot\|_p$ is the L^p norm, $L(\cdot, \cdot)$ is the loss function and γ and η are the hyperparameters. $\Phi(\cdot)$ indicates extracting the maximum probability of a discrete distribution.

Algorithm 1: Counterfactual Adversarial Training Approach (`CAT`)

Input: Dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, model $M^{(\theta)}$, mixup layer candidate set $\mathcal{Q}$, Beta distribution parameters α and β, denote as $Beta(\alpha, \beta)$, couterfactual adversarial loss iteration step $\mathcal{L}$, warm up step $\mathcal{K}$, max step $\mathcal{T}$

for *step* $k \in \{0, 1, ..., \mathcal{K}\}$ **do**

   Sample one batch $X^{(k)} \in D$. Denote corresponding representations as $h^{(k)}$; Do ERM on $M^{(\theta)}(h^{(k)})$;

for *step* $t \in \{0, 1, ..., \mathcal{T}\}$ **do**

   Sample one batch $X^{(t)}$. Denote corresponding representations as $h^{(t)}$;

   For each $x^{(i)}$ in $X^{(t)}$, random sample $q \in \mathcal{Q}$ and $\lambda^{(i)} \sim Beta(\alpha, \beta)$ and generate mixed representations in latent space using (Eq.4) to get one batch of counterfactual representations $\tilde{h}^{(t)}$;

   for $l \in \{0, 1, ...\mathcal{L}\}$ **do**

      Optimize counterfactual representations using CAL (Eq.5);

   Do CRM on $M^{(\theta)}(\tilde{h}^{(t)})$ and $M^{(\theta)}(h^{(t)})$;

   Do ERM on $M^{(\theta)}(h^{(t)})$;

CAL is a trade-off game for minimizing difference and maximizing label change. Maximizing $-\left\|\lambda^{(i)}\right\|_p$ encourages smaller shifts of counterfactual representations, while maximizing loss function adversarially changes model prediction to anyone but not original label $y^{(i)}$. The last term is to make the model more confident about counterfactual representation predictions.

3.3 Counterfactual Risk Minimization

CRM is designed to enable the model to learn from both original representations and counterfactual ones. Recall that in supervised learning, given (h, y) and their joint distribution P, the model M^{θ} : $H \rightarrow Y$ is learnt by minimizing the average of loss function over data distribution P, also known as the *expected risk*:

$$R(M^{(\theta)}) = \int L(M^{(\theta)}(h), y) dP(h, y). \quad (6)$$

Unfortunately, distribution P in unknown in most practical situations so we approximate Eq.6 by an empirical form:

$$\hat{R}(M^{(\theta)}) = \frac{1}{n} \sum_{i=1}^{n} L(M^{(\theta)}(h^{(i)}), y^{(i)}). \quad (7)$$

Minimizing Eq.(7) is known as the Empirical Risk Minimization (ERM) principle (Vapnik and Vapnik,

1998), which is widely adopted in most of machine learning models today.

Counterfactual Risk Minimization (CRM) is derived from ERM. Similar to Swaminathan and Joachims (2015), Charles et al. (2013) and Jung et al. (2020), we rewrite Eq.(6):

$$
\begin{aligned}
R(M^{(\theta)}) &= \mathbb{E}_{h \sim P(H)} \mathbb{E}_{y \sim P(Y|h)} L(M^{(\theta)}(h), y) \\
&= \mathbb{E}_{h \sim P(H)} \mathbb{E}_{y \sim P(Y|\tilde{h})} \left[\frac{P(y|h)}{P(y|\tilde{h})} L(M^{(\theta)}(h), y) \right] \\
&:= R_c(M^{(\theta)}),
\end{aligned}
$$
$$(8)$$

where the subscript c denotes that $R_c(\cdot)$ is from the counterfactual distribution. Since we can estimate $P(y|h)$ and $P(y|\tilde{h})$ by the model $M^{(\theta)}$, we can derive a tractable estimation for Eq.(8) via Monte Carlo approximation:

$$
\begin{aligned}
\hat{R}_c(M^{(\theta)}) &= \frac{1}{n} \sum_{i=1}^{n} \frac{\Phi(M^{(\theta)}(h^{(i)}))}{\Phi(M^{(\theta)}(\tilde{h}^{(i)}))} L(M^{(\theta)}(h^{(i)}), y^{(i)}) \\
&= \frac{1}{n} \sum_{i=1}^{n} \hat{\omega}(h^{(i)}) L(M^{(\theta)}(h^{(i)}), y^{(i)}).
\end{aligned}
$$
$$(9)$$

We call Eq.(9) *counterfactual risk*, which can be also viewed as an importance sampling estimator that connects original-counterfactual distributions. Intuitively, CRM adjusts sample-wise loss weight dynamically according to $\hat{\omega}(h^{(i)})$. The counterfactuals that have low confidence are more penalized and conversely the over-confident original data are discouraged. This makes the decision boundary more discriminative and smooth. In practice, we bound $\hat{\omega}(h^{(i)})$ for numerical stability:

$$\hat{R}_c(M^{(\theta)}) = \frac{1}{n} \sum_{i=1}^{n} B(\hat{\omega}(h^{(i)})) L(M^{(\theta)}(h^{(i)}), y^{(i)}), \quad (10)$$

where

$$B(x) = \begin{cases} x & x \in [A_1, A_2] \\ A_1 & x < A_1 \\ A_2 & x > A_2, \end{cases} \quad (11)$$

for some $0 < A_1 < A_2$.

During training, we first train a warm-up phase using only ERM like traditional fine-tuning. Then we start counterfactual adversarial training with ERM and CRM alternatively for succeeding steps. The whole framework is in algorithm 1. One of the special cases are discussed in the A.1.

How CAL helps CRM: (a) <u>Stability</u>: In Eq.(9), when the denominator $\Phi(M^{(\theta)}(\tilde{h}^{(i)}))$ gets close to zero, $\hat{R}_c(M^{(\theta)})$ can be arbitrarily far away from the true risk. The last term in CAL tend to maximize $M^{(\theta)}(\tilde{h}^{(i)})$ so that CRM is more stable. (b) <u>Smaller variance</u>: The optimal choice is $P^*(y|\tilde{h}^{(i)}) = P(y|h^{(i)})L(M^{(\theta)}(h)^{(i)}, y^{(i)})/\mu$ (Glasserman, 2004), where μ is the true expected risk. In practice, although this is untraceable since μ is unknown, a $P(y|\tilde{h}^{(i)})$ approximately proportional to $P(y|h^{(i)})L(M^{(\theta)}(h^{(i)}), y^{(i)})$ is preferred for variance reduction. Note that we can view traditional ERM as an importance sampling with $P(y|\tilde{h}^{(i)}) = P(y|h^{(i)})$ and are estimated by $\Phi(M^{(\theta)}(h^{(i)}))$. Here we consider two situations:

(1) When $L(M^{(\theta)}(h^{(i)}), y^{(i)})$ is large, then $\Phi(M^{(\theta)}(h^{(i)}))$ is likely small. This is an indication of the model sensitivity, which means the gradient regarding $\gamma L(M^{(\theta)}(h^{(i)}), y^{(i)})$ in CAL is large to push the counterfactual in $\nabla \max ||\lambda^{(i)}||_p$ direction (towards $h^{(j)}$), hence to generate $\Phi(M^{(\theta)}(\tilde{h}^{(i)})) > \Phi(M^{(\theta)}(h^{(i)}))$ during optimization.

(2) When $L(M^{(\theta)}(h^{(i)}), y^{(i)})$ is small, which indicate the model is confident, gradient regarding $\lambda^{(i)}$ in $-||\lambda^{(i)}||_p$ is more possible to dominate the total gradient of CAL and pull the counterfactuals close to $h^{(i)}$, then $\Phi(M^{(\theta)}(\tilde{h}^{(i)})) \approx \Phi(M^{(\theta)}(h^{(i)}))$. To conclude, in both situations, $\Phi(M^{(\theta)}(\tilde{h}^{(i)}))$ is a better choice of $P(y|\tilde{h}^{(i)})$ (at least not worse) in the context of variance reduction.

3.4 CAT in Question Answering

In addition to classification tasks that most mixup-based methods are examined on, we further extend CAT to question answering where logic reasoning is preferred over correlation memorization. We propose four mixup strategies to handle question answering: i) directly mix, which is the same as in sentence classification task; ii) mix only on context; iii) mix only on queries; iv) mix only on non-answer contexts. Empirical results suggest that mixing only on non-answer contexts leads to the best and most consistent outcome. Regarding CAL and CRM, we sum up the start position loss and end position loss as the final loss.

4 Experiments

We evaluate CAT on five widely used open-source benchmark datasets, including text classification, natural language inference and question answering.

4.1 Dataset

Yahoo! Answers (Chang et al., 2008) consists of questions and their corresponding answers along with the categories that are assigned to questions. We carry out the same pre-processing as in (Chen et al., 2020).

IMDB (Lin et al., 2011) is a typical dataset for binary sentiment analysis including 50k samples.

SNLI (Bowman et al., 2015) is a popular text entailment dataset that contains 570k human annotated sentence pairs.

SQuAD 1.1 (Rajpurkar et al., 2016) consists of 100k question/answer pairs. Given a question and a Wikipedia passage containing the answer, the task is to predict the answer span in the passage.

SQuAD 2.0 ((Rajpurkar et al., 2018)) combines the existing SQuAD 1.1 data with over 50k unanswerable questions written **adversarially** by crowdworkers.

For each dataset, experiments are conducted on multiple data sizes. Experiments are controlled in an incremental manner where we gradually increase the training set size. For sentence classification and natural language inference tasks, we randomly select a fixed test set of size 2000. For question answering, the full dev set is used for evaluation.

4.2 Implementation

For a fair comparsion, We employ BERT$_{\text{BASE}}$, BERT$_{\text{LARGE}}$, RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$, and BERT$_{\text{BASE}}$ with TMix (Chen et al., 2020) [2] as strong baselines. CAT is applied in two forms: CAT without CAL optimization (denoted as CAT *) and standard CAT. All models are concatenated with a two-layer perceptron with $Tanh$ as activation on the top. We adopt the fourth mix strategy in CMIX for question answering in 3.4. We report accuracy for sentence classification and natural language inference and EM/F1 for question answering.

We summarize test hyperparameters in Table 4, and introduce the model-specific mixup candidate layers as follows: $\{8, 9, 10\}$ (see Section 4.5 for detailed illustration) is used for BERT$_{\text{BASE}}$, $\{20, 21, 22\}$ for BERT$_{\text{LARGE}}$, $\{6, 7, 8\}$ for RoBERTa$_{\text{BASE}}$ and $\{17, 18, 19\}$ for RoBERTa$_{\text{LARGE}}$. During experiments, every trial is repeated 3-5 times. While for large versions of the

---

[2]Since the original TMix is trained without attention mask, in our experiments, we add attention mask aligning with CAT.

Model	Yahoo! Answers				IMDB				SNLI			
	10	50	250	1000	10	50	250	1000	10	50	250	1000
BERT$_{BASE}$	61.02	66.39	70.07	72.33	73.28	78.03	82.38	85.88	42.68	57.62	70.17	77.16
TMix	62.19	67.01	70.15	72.30	74.32	78.64	82.58	85.90	43.90	58.55	70.57	77.40
CAT *	62.34	67.20	70.11	72.29	73.77	78.98	82.45	85.96	44.37	59.42	71.23	77.89
CAT	**63.53**	**68.11**	**71.40**	**72.52**	**75.55**	**80.13**	**83.15**	**86.11**	**46.23**	**60.27**	**72.13**	**78.20**
RoBERTa$_{BASE}$	61.95	66.96	69.61	71.21	81.57	84.30	87.00	88.36	40.72	59.92	77.96	83.09
CAT *	63.09	**67.84**	70.08	71.95	82.80	85.11	87.40	88.45	**41.95**	63.33	79.15	83.25
CAT	**63.55**	67.78	**70.45**	**72.02**	**83.25**	**85.12**	**87.50**	**88.93**	41.30	**64.47**	**79.69**	**83.75**
BERT$_{LARGE}$	63.54	67.96	70.75	72.93	76.51	81.22	85.42	**87.32**	**44.33**	60.10	74.02	81.04
CAT *	64.33	68.07	70.72	72.95	**76.97**	81.05	85.38	86.93	43.07	62.80	75.97	81.18
CAT	**64.73**	**68.15**	**70.95**	**73.06**	75.10	**82.52**	**86.02**	87.00	43.83	**64.77**	**76.77**	**81.67**
RoBERTa$_{LARGE}$	64.38	67.80	70.60	72.28	81.50	87.63	89.03	90.06	38.22	62.73	82.27	85.99
CAT *	66.20	68.92	71.10	72.90	79.95	87.55	89.48	90.10	39.15	61.85	82.90	85.63
CAT	**66.30**	**69.28**	**71.25**	**73.30**	**84.80**	**88.55**	**89.85**	**90.10**	**40.33**	**65.07**	**83.15**	**86.05**

Table 1: The average accuracy after multiple runs on Yahoo! Answers, IMDB and SNLI datasets. Bellowing the individual dataset is the number of training samples per class.

Model	SQuAD 1.1			SQuAD 2.0		
	1/20	1/10	1/5	1/20	1/10	1/5
BERT$_{BASE}$	51.83/62.50	66.06/76.56	72.25/81.75	51.10/54.12	55.60/58.84	61.84/65.42
CAT *	**63.90/74.93**	69.36/79.44	74.10/83.34	55.44/57.55	**59.84/62.44**	61.77/64.97
CAT	62.71/74.14	**69.49/79.44**	**74.33/83.43**	**56.22/58.47**	59.71/62.44	**63.26/66.72**
BERT$_{LARGE}$	70.66/81.29	75.85/85.16	79.14/87.24	59.41/63.03	66.28/70.30	**71.30/74.88**
CAT *	72.18/82.15	75.69/84.83	79.06/87.08	61.84/65.27	66.55/70.08	69.40/72.87
CAT	**72.30/82.17**	**76.37/85.09**	**79.18/87.28**	**61.82/65.32**	**67.38/70.79**	69.31/72.37

Table 2: The model performance of EM/F1 on SQuAD 1.1 and SQuAD 2.0. Bellowing the individual dataset is the proportion of full training data used.

Model	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
BERT$_{BASE}$	80.80	88.50	72.57	75.99
CAT	**81.77**	**88.98**	**74.13**	**77.36**

Table 3: The EM/F1 on full QA data.

pre-trained models, we also observe some unstable results as mentioned by Devlin et al. (2019), so the outliers are removed and the average accuracy is reported to reduce randomness. We also present other implementation details and hyperparameters in Appendices A.3, A.4 and A.5.

4.3 Results

Performance of CAT on classification and NLI. As shown in Table 1, it can be seen that CAT achieves the best performance at the most settings across different pre-trained models. Improvements become more significant when the data size decreases, which is aligned with the analysis by Yue et al. (2020) that when the size of training data gets smaller, the impact of spurious bias increases.

Besides, a more remarkable performance gain is observed on SNLI than that on IMDB which is relatively a simpler task. For instance, RoBERTa$_{BASE}$ achieves 81.57 with only 10 samples per class. A simple task may be less influenced by spurious bias thus limit the improvement of CAT. To verify our intuition, we remove the attention mask to increase the difficulty on IMDB, and the improvement on BERT$_{BASE}$ for CAT increases to 4.5, 2.9, 1.7, 1.0 percent for 10, 50, 250, 1000 number of class samples setting. As a result, we conclude that benefits given by CAT are more obvious for a challenging task where spurious bias is serious.

Performance of CAT on question answering. CAT achieves the best performance nearly across all settings (Table 2). Similar to the trend in sentence classification and NLI, the improvement increases as the data size decreases. Particularly, CAT improves BERT$_{BASE}$ by 10.88%, 3.43%, 2.08% in EM respectively when data is 1/20, 1/10, 1/5 of the full SQuAD 1.1. Another noteworthy point is on average there are large improvements in EM than F1 (For example, 5.22%, 4.11% and 1.42% in EM

Task	Batch Size	α	β	γ	η	A_1	A_2	Adv. Step	Adv. LR
Classfication & NLI	8	0.3	0.3	10	20	0	10	3	$2e^{-2}$
QA	12	5 (2)	5 (2)	10	20	0.7	10 (2)	1	$5e^{-2}$

Table 4: Hyperparameters for CAT, number in brackets are for large version pre-trained models

on SQuAD 2.0 and 4.35%, 3.60% and 1.30% in F1 for BERT$_{\text{BASE}}$ as data size grows), which shows that our mixup strategy can help model find answer boundary more precisely through the counterfactuals on non-answer context. Evaluation results in Table 3 shows CAT still effective with full data.

Improvement of CAT becomes more significant under adversarial QA. We also observe a larger boosting on SQuAD 2.0 than SQuAD 1.1 averagely (Table 3), which demonstrates that CAT can achieve more considerable improvement on more adversarial data than on benign data. Full data result also align with the trend (Table 2), which demonstrates the effectiveness of CAT on full adversarial data.

4.4 Case Study

We further explore a concrete spurious bias, which is a statistical association between labels and some certain phrase in inputs, that has no real causation with labels. Take the phrase *on a bench* on SNLI for instance, it logically has no causal effect on sentence pairs relations but may have statistical correlations with labels. To achieve that, we manually build a training set with 100 samples of which 10% are with label entailment, 80% with label contradiction and 10% with label neutral. Such skewed distribution indicates a spurious bias between *on a bench* and *contradiction*. Then we build an equal-size test set, also equipped *on a bench*, with the 40%, 20% and 40% proportion respectively. Hence test set is a out of distribution (OOD) set w.r.t. phrase *on a bench*. The result shows that BERT$_{\text{BASE}}$-CAT improves the baseline significantly from 0.35 to 0.48 in Acc by successfully overcome the bias. For instance, samples with phrase *"a man sits on a bench"* are all contradictions during training but 83% of them are neutral or entailment in test data. This phenomenon indicates that CAT successfully alleviates this particular bias to better estimate the true causal effect between inputs and outputs, especially under OOD setting.

4.5 Analysis

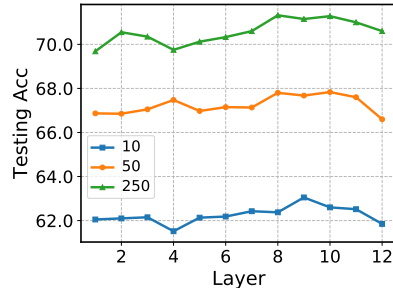**Impact of Interpolation Layers**. We study the impact of using different layers for interpolation

Figure 3: The impact of different interpolation layer on testing accuracy on Yahoo! Answer dataset.

on BERT$_{\text{BASE}}$ and the result is shown in Figure 3. It is clear that the 8, 9, 10-th layers obviously outperform other layers consistently. In addition, we take $\{8, 9, 10\}$ layers as a candidate set and then sample one of them in each batch, which can further improve the model performance. For the other 3 language models, due to the limited computational resources, we experiment on several layers combinations and report the best one.

Performance and Stability of CAT *. It is observed that CAT * can also improve pre-trained models but is less impressive than CAT. More importantly, CAT * is more unstable and sometimes even worse than baselines, e.g., CAT * for RoBERTa$_{\text{LARGE}}$ on IMDB and RoBERTa$_{\text{LARGE}}$ on SQuAD 2.0. The reason can be seen in section 3.3, where we discuss how CAL helps CAT in both performance and stability.

Training Process of CAT and CAT *. We further explore how normal training data and counterfacutals evolve in latent space during training, particularly by visualization. Taking CAT on SNLI as an example, the representations during training are shown in Figure 4 (a), in which blue nodes denote original data representations and orange ones are counterfactuals. Obviously, We can conclude the process as three main stages illustrated as follows:

First stage (epoch 0-4): The original data representations and counterfactuals are entangled together and the decision boundary for original data is not clear.

Second stage (epoch 6-10): Model begins to converge and the decision boundary for original data becomes more clear. Counterfactuals diverge
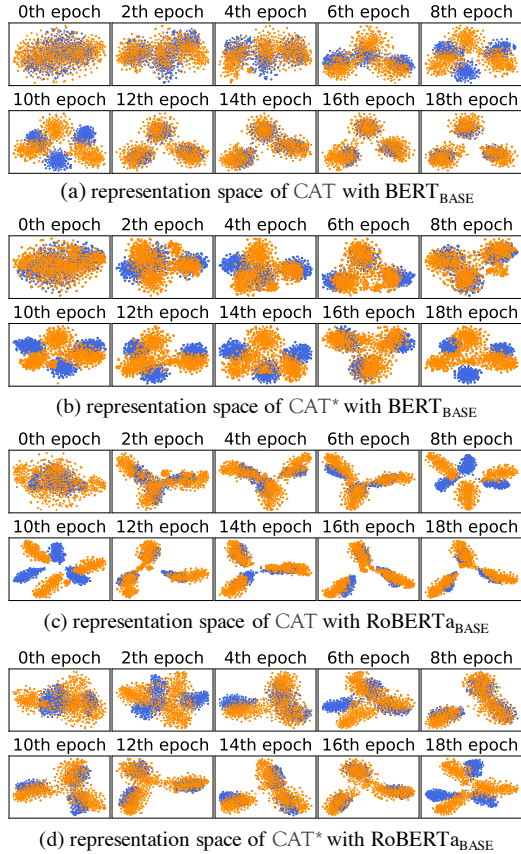
Figure 4: Representation space visualization through tSNE for CAT and CAT *. during the training process on SNLI data with 250 samples per class. (a) and (b) represent CAT and CAT * on BERT$_{BASE}$ and (c) and (d) for RoBERTa$_{BASE}$

.

gradually from the original data and clusters at the decision boundary. Such observation indicates that the classification loss manifold is not smooth enough and has many cliffs. Therefore, the gradient of CAL tends to pull the counterfactuals far away from the original data. In other words, $\gamma L(M^{(\theta)}(\tilde{h}^{(i)}), y^{(i)}))$ dominates the trade-off game over $\left\| \lambda^{(i)} \right\|_p$.

Third stage (epoch 12-18): Counterfactuals move closer to the original data and few nodes are at the decision boundary. This observation shows the classification loss manifold becomes smoother and most of the cliffs disappear, thus main part of generated gradient of CAL during optimization starts to maximize $\left\| \lambda^{(i)} \right\|_p$ instead of $\gamma L(M^{(\theta)}(\tilde{h}^{(i)}), y^{(i)}))$, which pulls counterfactuals close to the original data.

Through counterfactuals, CAT alleviates the impact of spurious bias and encourages the model to discover the causal effect between representations and labels, thus helping pre-trained models con-

struct a smoother and clearer loss manifold. The phenomenon that counterfactuals finally locates nearby the original data in representation place at stage 3 is well consistent with our counterfactual definition: minimally-different from original data but leads to different labels.

While for CAT *, as shown in Figure 4 (b), we also observe the similar stage 1 and stage 2, but the stage 3 is significantly different. Such difference indicates CAT * could only conduct random interpolation from a Beta distribution without CAL to further optimize counterfactuals. As a result, lots of interpolation representations will locate at the vicinity of the decision boundary when the model converges. Such representation has two obvious weaknesses compared with that of CAT: i) It may lead to the unstable performance of CRM since $Phi(M^{(\theta)}(\tilde{h}^{(i)}))$ could be extremely small. ii) The interpolated representation is not as efficient as minimal-different counterfactual ones in boosting performance as shown in Kaushik et al. (2019).

Additonally, we observe similar training process visualization when applying CAT and CAT * on SNLI for RoBERTa$_{BASE}$. The results are shown in Figure 4 (c) and 4 (d) respectively, which demonstrates an consistent convergence process has been achieved cross different pre-trained models.

5 Conclusion

To alleviate the spurious correlation bias in training corpus and encourage causal discovery instead of simple correlations, we propose CAT from the causality perspective for introducing counterfactual representations in the training stage through latent space interpolation. Through extensive experiments on three benchmarks on the text classification, natural language inference and question answering tasks, we demonstrate that CAT is effective in promoting testing accuracy especially in the small data scenario, which outperforms SOTA baselines across different pre-trained models.

Acknowledgements

References

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059.

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

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835.

Denis Charles, Max Chickering, and Patrice Simard. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157.

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)*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573.

Paul Glasserman. 2004. *Monte Carlo methods in financial engineering*. Springer, New York.

Yonghan Jung, Jin Tian, and Elias Bareinboim. 2020. Learning causal effects via weighted empirical risk minimization. *Advances in Neural Information Processing Systems*, 33.

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*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. 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 452–457.

Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. 2011. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

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*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

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

Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116.

Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823. PMLR.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Vladimir Vapnik and Vlamimir Vapnik. 1998. Statistical learning theory wiley. *New York*, 1:624.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019a. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. 2019b. Interpolation consistency training for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3635–3641. AAAI Press.

Vikas Verma, Meng Qu, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. 2019c. GraphMix: Regularized training of graph neural networks for semi-supervised learning. arxiv e-prints, art. *arXiv preprint arXiv:1909.11715*.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. 2020. Interventional few-shot learning. *Advances in Neural Information Processing Systems*, 33.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280.

Cheng Zhang, Kun Zhang, and Yingzhen Li. 2020. A causal view on robustness of neural networks. *Advances in Neural Information Processing Systems*, 33.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. FreeLB: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

A Appendices

A.1 Same Label Data Mixup

During `CMIX`, it could happen that $\tilde{y}^{(i)} = y^{(i)}$ especially if the two original samples used for `CMIX` are with the same label. In this case $\omega(h^{(i)}) \approx 1$ and CRM degenerate into ERM thus do no harm to our discriminative model. Furthermore, since $\lambda^{(i)}$ is an exogenous random variable, as the training steps grow, we can expect a different $\tilde{h}^{(i)}$ in the next epoch such that $\tilde{y}^{(i)} \neq y^{(i)}$.

A.2 Beta Distribution Hyperparameters

An interesting question here is the choice of prior $Beta(\alpha, \beta)$. Intuitively, the expectation of $Beta(\alpha, \beta)$ is $\alpha/(\alpha + \beta)$, thus one may want this value be close to 1 to make counterfactuals start searching from vicinity of observations. In experiments we find this strategy works better when the dataset is small. We think the reason is that when observations become more, the average distance between each sample goes down and the observation distribution become denser, thus the gains of searching from the vicinity is limited. Another noteworthy point is whether to choose a unimodal or bimodal distribution. Unimodal distribution generates $\lambda^{(i)}$ in the middle while bimodal distribution pushes $\lambda^{(i)}$ close to either 0 or 1. During the experiments we find the latter is usually better since it can explore a large range of interpolations.

A.3 Other Implementation Details

The datasets statistic are in Table 5

Dataset	Classes	Test	Task
Yahoo! Answers	10	2000	classification
IMDB	2	2000	classification
SNLI	3	2000	NLI
SQuAD 1.1	NA	10570	QA
SQuAD 2.0	NA	11873	QA

Table 5: The details of test set used in our experiments.

We train sentence classification and natural language inference tasks on a single Tesla V100, question answering tasks on Tesla P100 and NVIDIA A100. We observe some performance discrepancy on question answering especially when data is small so we train all base models on Tesla P100 and all large models on NVIDIA A100 for consistency.

A.4 Other Hyperparameters

For sentence classification and NLI tasks, epoch is set as 30 when the number of samples in each category is less than 50, otherwise is set as 20. For question answering, epoch is set as 2 regardless of training data size.

Regarding warm-up steps, for sentence classification and natural language inference tasks, we take the first epoch for warm-up training. For question and answering tasks, we choose from $\{300, 600, 1000\}$ steps to find the best. For learning rate, we choose from $\{1e - 5, 2e - 5\}$ for small data (number of samples in each class is less than 50) and fix the learning rate to $1e - 5$ otherwise. Max sequence length is set as 128 for Yahoo! Answers and SNLI, 256 for IMDB and 384 for SQuAD 1.1 and SQuAD 2.0.

During experiments we find that $\text{BERT}_{\text{LARGE}}$ and $\text{RoBERTa}_{\text{LARGE}}$ are harder to train than corresponding base version and sometimes crash for long time training. Therefore we make some adjustments to mitigate it such as a smaller learning rate, tighter CRM bound A_1 and A_2, longer warm-up steps, etc.

A.5 Hyperparameters Searching

We list the hyperparameters searching space here. For α and β in beta distribution, we iterate through $[0.1, 0.3, 0.5, 0.7, 0.9, 2, 5, 10]$ to find the best setting. For γ and η, we keep $\eta = 2\gamma$ to reduce complexity and search γ from 0 to 20. For CRM lower and upper bound, we gradually narrow our interval from $[0, 10]$ to $[0.9, 2]$ to find the optimal setting. We find that `CAT` is difficult to converge with a lower bound near to 0 under QA task. Finally, as regard to adversarial settings, we try learning step in $[1, 3, 5, 10]$ and learning rate in $[2e - 3, 5e - 3, 2e - 2, 5e - 2]$. For the CRM learning rate, we generally set $2e - 5$. except that we find for $\text{BERT}_{\text{LARGE}}$ and $\text{RoBERTa}_{\text{LARGE}}$ on SNLI, when sample size per class is less than 50, a smaller learning rate $1e - 5$ will lead to better and more stable performance.