Counterfactual Generation Under Confounding

Abbavaram Gowtham Reddy
IIT Hyderabad, India
cs19resch11002@iith.ac.in

Saloni Dash
Microsoft Research Bengaluru, India
salonidash77@gmail.com

Amit Sharma
Microsoft Research Bengaluru, India
amshar@microsoft.com

Vineeth N Balasubramanian
IIT Hyderabad, India
vineethnb@iith.ac.in

Abstract

A machine learning model, under the influence of observed or unobserved confounders in the training data, can learn spurious correlations and fail to generalize when deployed. For image classifiers, augmenting a training dataset using counterfactual examples has been empirically shown to break spurious correlations. However, the counterfactual generation task itself becomes more difficult as the level of confounding increases. Existing methods for counterfactual generation under confounding consider a fixed set of interventions (e.g., texture, rotation) and are not flexible enough to capture diverse data-generating processes. We formally characterize the adverse effects of confounding on any downstream tasks and show that the correlation between generative factors can be used to quantitatively measure confounding. To minimize such correlation, we propose a counterfactual generation method that learns to modify the value of any attribute in an image and generate new images. Our method is computationally efficient, simple to implement, and works well for any number of generative factors and confounding variables. Our experimental results on both synthetic (MNIST variants) and real-world (CelebA) datasets show the usefulness of our approach.

1 Introduction

A confounder is a variable that causally influences two or more variables that are not necessarily directly causally dependent [26]. Often, the presence of confounders in a data-generating process is the reason for spurious correlations among variables in the observational data. The bias caused by such confounders is inevitable in observational data, making it challenging to identify invariant features representative of a target variable [33,23,39]. Removing the effects of confounding in trained machine learning models has shown to be helpful in various applications such as disentanglement, domain generalization, counterfactual generation, algorithmic fairness, etc. [35,19,45,42,34,11,6,32,9]. Recent years have seen a few efforts to handle the spurious correlations caused by confounding effects in observational data [36,34,11,32]. However, these methods either make strong assumptions on the underlying causal generative process or require strong supervision. In this paper, we study the adversarial effect of confounding in observational data on a classifier’s performance and propose a mechanism to marginalize such effects by counterfactual data augmentation.

The causal generative processes considered throughout this paper are shown in Figure 1(a). We assume that a set of generative factors (attributes) $Z_1, Z_2, \ldots, Z_n$ (e.g., background, shape, texture) and a label $Y$ (e.g., cow) cause a real-world observation $X$ (e.g., an image of a cow in a particular background) through an unknown causal mechanism $g$ [28]. To study the effects of confounding, we consider $Y, Z_1, Z_2, \ldots, Z_n$ to be confounded by a set of confounding variables $C_1, \ldots, C_m$.

*Equal contribution

(e.g., certain breeds of cows appear only in certain shapes or colors and appear only in certain countries). Such causal generative processes have been considered earlier for other kinds of tasks such as disentanglement \[35, 37, 32\].

A related recent effort by [34] proposes Counterfactual Generative Networks (CGN) to address this problem using a data augmentation approach. This work assumes each image to be composed of three Independent Causal Mechanisms (ICMs) \[29\] responsible for three fixed factors of variations: shape, texture, and background (as represented by \(Z_1, Z_2,\) and \(Z_3\) in Figure 1(b)). This work then trains a generative model that learns three ICMs for shape, texture, and background separately, and combines them in a deterministic fashion to generate observations. However, fixing the architecture to specific number and types of mechanisms (shape, texture, background) is not generalizable, and may not directly be applicable to settings where the number of underlying generative factors is unknown. It is also computationally expensive to train different generative models for each aspect of an image such as texture, shape or background.

In this work, we begin with quantifying confounding in observational data that is generated by an underlying causal graph of the form shown in Figure 1(a). We then provide a counterfactual data augmentation methodology called CONIC (Counterfactual geNeratI on under C onfounding). We hypothesize that the counterfactual images generated using the proposed CONIC method provide a mechanism to marginalize the causal mechanisms responsible for spurious correlations (i.e., causal arrows from \(C_i\) to \(Z_j\) for some \(i, j\)). We take a generative modeling approach and propose a neural network architecture based on conditional CycleGAN \[46\] to generate counterfactual images. Our contributions include:

- We formally quantify confounding in causal generative processes of the form in Fig 1(a), and study the relationship between correlation and confounding between any pair of generative factors.
- We present a counterfactual data augmentation methodology to generate counterfactual instances of observed data, that can work even under highly confounded data (~ 95% confounding) and provides a mechanism to marginalize the causal mechanisms responsible for confounding.
- We modify conditional CycleGAN to improve the quality of generated counterfactuals. Our method is computationally efficient and easy to implement.
- Following previous work, we perform extensive experiments on well-known benchmarks – three MNIST variants and CelebA datasets – to showcase the usefulness of our proposed methodology in improving the accuracy of a downstream classifier.

2 Related Work

Counterfactual Inference: [27], in his seminal text on causality, provided a three-step procedure for generation of a counterfactual data instance, given an observed instance: (i) Abduction: abduct/recover the values of exogenous noise variables; (ii) Action: perform the required intervention; and (iii) Prediction: generate the counterfactual instance. One however needs access to the underlying structural causal model (SCM) to perform the above steps for counterfactual generation. Since real-world data do not come with an underlying SCM, many recent efforts have focused on modeling the underlying causal mechanisms generating data under various assumptions [21, 18, 7, 47, 30, 41, 3, 25].

Generating Counterfactuals by Learning ICMs: In a more recent effort, assuming any real-world image is generated with three independent causal mechanisms for shape, texture, background, and a composition mechanism of the first three, [34] developed Counterfactual Generative Networks (CGN) that generate counterfactual images of a given image. CGN trains three Generative Adversarial Networks (GANs) \[13\] to learn shape, texture, background mechanisms and combine these three mechanisms using a composition mechanism \(g\) as \(g(\text{shape, texture, background}) = \text{shape} \odot \text{texture} \odot \text{background}\).
Disentanglement and Data Augmentation: The spurious correlations among generative factors have been considered in disentanglement [10][35]. The general idea in these efforts is to separate the causal predictive features from non-causal/spurious predictive features to predict an outcome. Our goal is different from disentanglement, and we focus on the performance of a downstream classifier instead of separating the sources of generative factors. Traditional data augmentation methods such as rotation, scaling, corruption, etc. [15][8][44][43] do not consider the causal generative process and hence they can not remove the confounding in the images via data augmentation.

A similar effort to our paper is by [11] who use CycleGAN to generate counterfactual data points. However, they focus on the performance of a subgroup (a subset of data with specific properties) which is different from our goal of controlling confounding in the entire dataset. Another recent work by [39] considers spurious correlations among generative factors and uses CycleGAN to generate counterfactual images. Compared to these efforts, rather than using CycleGAN directly, we propose a CycleGAN-based architecture that is optimized for controlled generation using contrastive losses.

3 Information Theoretic Measure of Confounding

Background and Problem Formulation: Let \( \{Z_1, Z_2, \ldots, Z_n\} \) be a set of random variables denoting the generative factors of an observed data point \( X \), and \( Y \) be the label of the observation \( X \). Each generative factor \( Z_i \) (e.g., color) can take on a value form a discrete set of values \( \{z_1, \ldots, z^d_i\} \) (e.g., red, green etc.). Let the set \( S = \{Y, Z_1, \ldots, Z_n\} \) generates \( N \) real-world observations \( \{X_i\}_{i=1}^N \) through an unknown causal mechanism \( g \) (Fig. 1). Each \( X_i \) can be thought of as an observation generated using the causal mechanism \( g \) with certain intervention on the variables in the set \( S \). Variables in \( S \) may potentially be confounded by a set of confounders \( C = \{C_1, \ldots, C_m\} \) that denote real-world confounding such as selection bias. Let \( D \) be the dataset of real-world observations along with corresponding values taken by \( \{Y, Z_1, \ldots, Z_n\} \). From a causal effect perspective, each variable in \( S \) has a direct causal influence on the observation \( X \) (e.g., the causal edge \( Z_i \rightarrow X \)) and also has non-causal influence on \( X \) via the confounding variables \( C_1, \ldots, C_m \) (e.g., \( Z_i \leftarrow C_j \rightarrow Z_k \rightarrow X \) for some \( C_j \) and \( Z_k \)). These paths via the confounding variables, in which there is an incoming arrow to the variables in \( S \), are also referred to as backdoor paths [20].

In any downstream application where \( D \) is used to train a model (e.g., classification), it is desirable to minimize or remove the effect of confounding variable to ensure that a model is not exploiting the spurious correlations in the data to arrive at a decision. In this paper, we present a method to remove the effect of such confounding variables using counterfactual data augmentation. We first study the relationship between confounding and the correlation between a pair of generative factors.

Definition 3.1. (Directed Information [31][40].) In a causal directed acyclic graph (DAG) \( G = (V, E) \), where \( V \) denotes the set of variables and \( E \) denotes the set of directed edges denoting the direction of causal influence among the variables in \( V \), the directed information from a variable \( Z_i \in V \) to another variable \( Z_j \in V \) is denoted by \( I(Z_i \rightarrow Z_j) \). It is defined as follows:

\[
I(Z_i \rightarrow Z_j) := D_{KL}(p(Z_i|Z_j)||p(Z_i|\text{do}(Z_j))) = \mathbb{E}_{p(Z_i, Z_j)} \log \frac{p(Z_i|Z_j)}{p(Z_i|\text{do}(Z_j))}
\]

Using Definition 3.1, it is easy to see that the variables \( Z_i \) and \( Z_j \) are unconfounded if and only if \( I(Z_i \rightarrow Z_j) = 0 \). Non zero directed information \( I(Z_j \rightarrow Z_i) \) entails that, \( p(Z_i|Z_j) \neq p(Z_i|\text{do}(Z_j)) \) and hence the presence of confounding (if there is no confounder, \( p(Z_i|Z_j) \) should be equal to \( p(Z_i|\text{do}(Z_j)) \)). Also, it is important to note that the directed information is not symmetric (i.e., \( I(Z_i \rightarrow Z_j) \neq I(Z_j \rightarrow Z_i) \) [17]). We use this fact in defining the measure of confounding below. Since we need to quantify the notion of confounding (as opposed to no confounding), we use directed information to quantify confounding as defined below.

Definition 3.2. (An Information Theoretic Measure of Confounding.) In a causal directed acyclic graph (DAG) \( G = (V, E) \), where \( V \) denotes the set of variables and \( E \) denotes the set of directed edges denoting the direction of causal influence among the variables in \( V \), the amount of confounding between a pair of variables \( Z_i \in V \) and \( Z_j \in V \) is equal to \( I(Z_i \rightarrow Z_j) + I(Z_j \rightarrow Z_i) \).
We propose a way to systematically generate counterfactual data that can marginalize the effect of another confounding edge w.r.t. the confounding effect of the confounding edge

\[ M \]

This mapping \( Z \) and \( Z \) for which \( \text{intervention} \) generating the counterfactual inference: learning the un-

due to augmentation has no spurious correlations between generative factors. If we observe spurious correlation between two generative factors \( Z_i, Z_j \) when they take on the values \( z_i \) and \( z_j \) respectively, generating counterfactual instances w.r.t. \( Z_j \) with the intervention \( \text{do}(Z_j = z'_j) \) and adding the counterfactual instances to original data breaks the correlation between \( Z_i, Z_j \). We now present our algorithm to generate counterfactual images in a systematic manner.

4 CONIC: Methodology

We propose a way to systematically generate counterfactual data that can marginalize the effect of any confounding edge \( C_i \rightarrow Z_j \) in Fig. 1(a) as explained below.

Removing The Confounding Effect of \( C_i \rightarrow Z_j \): In the causal graphs of the form 1(a), for paths of the form \( Z_j \leftarrow C_i \rightarrow Z_i \), we call the edges \( C_i \rightarrow Z_j \) and \( C_i \rightarrow Z_i \) as confounding edges since together, their existence is the reason for confounding in the data. Also, let \( (z_j^p, z_i^p) \) is one pair of attribute values taken by the variable pair \( (Z_j, Z_i) \) under extreme confounding (e.g., in the training set of colored MNIST dataset, correlation coefficient of 0.99 between color and digit is observed such that whenever color is red, digit is 7 etc.). To remove the effect of the confounding edge \( C_i \rightarrow Z_j \) w.r.t. the another confounding edge \( C_i \rightarrow Z_i \) (recall that confounding between \( Z_j, Z_i \) is present if only if there exists a pair of causal arrows \( C_i \rightarrow Z_j \) and \( C_i \rightarrow Z_i \) for some \( i \); due to this reason we consider the confounding effect of the confounding edge \( C_i \rightarrow Z_j \) w.r.t. another confounding edge \( C_i \rightarrow Z_i \), we consider two subsets \( T_1, T_2 \) of the observational data \( D \) which are constructed as follows. \( T_1 \) consists of the set of instances for which \( Z_j \neq z_j^p \) and \( Z_i = z_i^p \); \( T_2 \) consists of the set of instances for which \( Z_j = z_j^p \) and \( Z_i \neq z_i^p \). The size of \( T_1 \) is usually much smaller than the size of \( T_2 \) because of high correlation between \( Z_j \) and \( Z_i \) (e.g., there are more red 7’s than non-red 7’s).

Now, we learn a mapping \( M \) from the set \( T_1 \) to the set \( T_2 \) that changes the attribute \( Z_j \) while fixing the value of \( Z_i \) at \( z_i^p \). That is, for any given instance \( X \in T_1 \), for which \( Z_j \neq z_j^p \), \( M \) maps \( X \) to a different instance \( X' \) in which the value of the generative factor \( Z_j \) is changed to \( z_j^p \) (e.g., \( M \) takes red 9 as input and returns red 7 as output). This mapping \( M \) can be thought of as a function performing the 3-step counterfactual inference: learning the underlying generative factors, performing the intervention \( \text{do}(Z_j = z_j^p) \) and then generating the counterfactual instance \( X' \). Now, given an instance \( X \) for which \( Z_j \neq z_j^p \) and \( Z_i \neq z_i^p \), using \( M \), we can generate counterfactual instance \( X' \) in which \( Z_j = z_j^p \) and \( Z_i \neq z_i^p \). These counterfactual instances, when augmented with the original observed dataset \( D \), removes the effect of the confounding edge \( C_i \rightarrow Z_j \) w.r.t. the edge \( C_i \rightarrow Z_i \). That is, the counterfactual instances, when augmented with original data, breaks the correlation between \( Z_j \) and \( Z_i \). This process can now be repeated systematically for each confounding edge to generate

<table>
<thead>
<tr>
<th>Algorithm 1: Counterfactual Generation to Remove the Effect of Confounding Edge ( C_i \rightarrow Z_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Result:</strong> Counterfactual images that remove the confounding effect caused by the edge ( C_i \rightarrow Z_j )</td>
</tr>
<tr>
<td><strong>Input:</strong> ( D = {X_i}_{i=1}^N ), ( \text{Nodes} = {Z_i</td>
</tr>
<tr>
<td><strong>Initialize:</strong> cf_images = []</td>
</tr>
<tr>
<td>for each ( Z_i \in \text{Nodes} ) do</td>
</tr>
<tr>
<td>( T_1 = {X \in D</td>
</tr>
<tr>
<td>( T_2 = {X \in D</td>
</tr>
<tr>
<td>( M = \text{conditionalCycleGAN}(T_1, T_2) )</td>
</tr>
<tr>
<td>( \text{Factural}_\text{Imgs} = {X \in D</td>
</tr>
<tr>
<td>( \text{Counterfactuals} = M(\text{Factural}_\text{Imgs}) )</td>
</tr>
<tr>
<td>cf_images.append(Counterfactuals)</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>return cf_images</td>
</tr>
</tbody>
</table>
counterfactual instances that remove the spurious correlations. The overall procedure to generate counterfactual instances is summarized in Algorithm 1.

Earlier works use CycleGAN to generate counterfactual images that differ from original image by a single attribute feature \[39\] [11]. Given two domains/sets of images that differ w.r.t. only one generative factor \(Z_j\), a CycleGAN can learn to translate between the two domains by changing the attribute value of \(Z_j\). In this case, one can think of CycleGAN as a function performing the required intervention \(Z_j\) and generating counterfactual instance without modeling the underlying causal process. Concretely, CycleGAN is an architecture used to perform unsupervised domain translation using unpaired images. In a CycleGAN, a generator \(G_1\) first transforms a given image \(X\) from a domain/set \(T_1\) into \(X'\) so that \(X'\) appears to come from another domain/set \(T_2\) such that certain features from input \(X\) are preserved in the output \(X'\). A discriminator \(D_{T_2}\) then classifies whether the translated image \(X'\) is original (i.e., sampled from \(T_2\)) or fake (i.e., generated by \(G_1\)). A second generator \(G_2\) transforms the image \(X'\) back to original image \(X\) to ensure that \(G_1\) is using the contents of \(X\) to generate \(X'\). The same procedure is repeated to translate images from domain \(T_2\) into domain \(T_1\). The loss function of CycleGAN can be written as \(L_{cycle\ GAN} = L_{GAN}(G_1, D_{T_2}, X, X') + L_{GAN}(G_2, D_{T_1}, X', X) + L_{cycle}(G_1, G_2)\) Where \(L_{GAN}\) is simple Generative Adversarial Network (GAN) [12] loss and \(L_{cycle}\) is cycle consistency loss measuring how well the output of \(G_2\) is matching with the original input \(X\). For example, \(L_{cycle}(G_1, G_2) = E_{X~T}[||G_2(G_1(X)) - X||_1]\) can ensure that \(G_2(G_1(X)) = X\). We use conditional variant of CycleGAN to leverage the supervision in terms of attribute values. For each generator, along with input, we also feed a desired target attribute as shown in the Figure 2.

To improve the quality of counterfactual images generated by conditional CycleGAN under extreme confounding, we propose a modification to conditional CycleGAN as detailed below. As discussed earlier, \(X'\), the output of \(G_1\), can be thought of as a counterfactual image of \(X\). When changing the feature \(Z_j\) of \(X\), we keep the feature \(Z_i\) fixed. That is, the representation for \(Z_j\) in both \(X\) and \(X'\) should be different and the representation for \(Z_i\) in both \(X\) and \(X'\) should be same. To ensure this, as shown in Figure 2 along with two generators \(G_1, G_2\) and a discriminator \(D_{T_2}\) that are part of conditional CycleGAN, we add two pre-trained discriminators \(L_1, L_2\) (shown in green color in Fig. 2). \(L_1\) takes two images \(X, X'\) as input and returns high penalty if the representation of \(Z_j\) is similar in \(X, X'\) and small penalty otherwise. \(L_2\) takes two images \(X, X'\) as input and returns high penalty if the representation of \(Z_i\) is different and small penalty otherwise. Thus, our overall objective to generate good quality counterfactual images is to train the modified conditional CycleGAN by minimizing the following objective.

\[
L_{conic} = L_{cycle\ GAN} - L_{con}(L_1(X), L_1(G_1(X))) + L_{con}(L_2(X), L_2(G_1(X))) \\
- L_{con}(L_1(X'), L_1(G_2(X'))) + L_{con}(L_2(X'), L_2(G_2(X'))) \tag{2}
\]

Where \(L_{con}\) is the contrastive loss [14]. For a pair of images \((X, X')\), \(L_{con}\) defined \(L_{con}(X, X') = AD^2 + (1 - A)\max(\epsilon - D, 0)^2\) Where \(A = 1\) if \(X, X'\) belong to same class (or have same attribute values), \(A = 0\) if \(X, X'\) belong to different classes (or have different attribute values). \(D\) is the distance between the representations of \(X, X'\) (e.g., Euclidean distance), \(\epsilon\) is the margin of error allowed between two representations of the images of different classes. \(L_1\) and \(L_2\) are pre-trained models and the parameters of \(L_1\) and \(L_2\) are fixed. That is, the loss values returned by \(L_{con}\) are only used to update the trainable parameters of conditional CycleGAN.

A Downstream Task - Image Classification: To measure the goodness of counterfactual generation, we consider the classification task on the unconfounded test set as a downstream task. Let \(D^{aug} = \{(X_i, Y_i)\}_{i=1}^M\) be the augmented dataset of original data \(D\) and corresponding counterfactual data points. Using \(D^{aug}\), we minimize \(L_{aug} := E_{(X, y) \sim D^{aug}}[\ell(f_\theta(X), y)]\). Where \(\ell\) is cross entropy loss.
We showed the relationship between confounding and correlation in the causal processes considered. Now, we present the experimental results on both synthetic (MNIST variants) and real world (CelebA) datasets. Having access to the ground truth generative factors (i.e., \( Z_1, \ldots, Z_n \)), we artificially create confounding in the training data and we leave test data to be unconfounded (i.e., no correlation among generative factors). We compare CONIC with various baselines including Empirical Risk Minimizer (ERM), Conditional GAN (CGAN) [12], Conditional VAE (CVAE) [20], Conditional-\( \beta \)-VAE (C-\( \beta \)-VAE) [16], AugMix [15], CutMix [43], Invariant Risk Minimization (IRM) [11], and Counterfactual Generative Networks (CGN) [34].

**MNIST Variants:** We construct the following three synthetic datasets based on MNIST dataset [22] and its colored, texture, and morpho (where the digit thickness is controlled; Fig. 3) variants [1, 2, 34]: (i) colored morpho MNIST (CM-MNIST), (ii) double colored morpho MNIST (DCM-MNIST), and (iii) wildlife morpho MNIST (WLM-MNIST). We consider extreme confounding among generative factors as explained below. For the experimental results shown in Table 1 in the training set of CM-MNIST dataset, the correlation coefficient between digit label and digit color \( r(label, \text{color}) \) is 0.95 and the digits from 0 to 4 are thin and digits from 5 to 9 are thick (see Figure 3). That is, \( r(label, \text{thin}) = 1 \) if the digit is in \([0,1,2,3,4]\) else \( r(label, \text{thick}) = 1 \). In the training set of DCM-MNIST dataset, digit label, digit color, and background color jointly take a fixed set of values 95% of the time. That is, \( r(label, \text{color}) = r(color, \text{background}) = r(label, \text{background}) = 0.95 \) and the digits from 0 to 4 are thin and digits from 5 to 9 are thick. In the training set of WLM-MNIST dataset digit shape, digit texture, and background texture jointly take a fixed set of attribute values 95% of the time and the digits from 0 to 4 are thin and digits from 5 to 9 are thick. Table 1 shows the results in which CONIC outperforms all the baselines. See Appendix for comparison of augmented images by various baselines. Coninc uses only 10000, 15000, 15000 counterfactual images in CM-MNIST, DCM-MNIST, and WLM-MNIST experiments respectively to get improved performance.

**CelebA:** Unlike MNIST variants, CelebA [23] dataset implicitly contains confounding (e.g., the percentage of males with blond hair is different from the percentage of females with blond hair, in addition to the difference in the total number of males and females in the dataset). In this experiment, we consider the performance of a classifier trained on the augmented data that predicts hair color given an image. Our test set is the set of males with blond hair. We train models on the train set and test the performance on the set of males with blond hair. Since the number of males with blond hair is very low in the dataset (approximately 4% of males have blond hair), we show that the augmenting the train set with only 10000 images of males with blond hair improves the performance over baselines (see Table 1) whereas other baselines require more than 50000 augmented images to get minor improvement over ERM. Given a male image with non-blond hair, CONIC generates the counterfactual image with blond hair without changing the male attribute (see Appendix for sample counterfactual images). We also note that the deterministic models such as CGN fail when they are applied to a different task where the number and type of generative factors are not fixed and are difficult to separate (e.g., CelebA). CGN results in table 1 are obtained with only 1000 counterfactual images as augmented data points. When we increase the number of counterfactual instances, performance of CGN reduces further.

## 5 Experiments and Results

### Table 1: Test set accuracy results on MNIST variants and CelebA

<table>
<thead>
<tr>
<th>Model</th>
<th>CM-MNIST</th>
<th>DCM-MNIST</th>
<th>WLM-MNIST</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>46.41±0.81</td>
<td>43.31±2.30</td>
<td>28.28±0.70</td>
<td>70.64±6.93</td>
</tr>
<tr>
<td>CGAN</td>
<td>41.86±1.79</td>
<td>30.66±3.86</td>
<td>17.50±0.85</td>
<td>70.99±2.35</td>
</tr>
<tr>
<td>CVAE</td>
<td>49.58±1.50</td>
<td>41.99±1.10</td>
<td>34.19±1.58</td>
<td>71.50±1.82</td>
</tr>
<tr>
<td>C−( \beta )-VAE</td>
<td>51.22±1.00</td>
<td>51.58±2.56</td>
<td>33.90±1.87</td>
<td>74.29±0.65</td>
</tr>
<tr>
<td>AugMix</td>
<td>47.36±0.01</td>
<td>44.85±0.02</td>
<td>26.30±1.30</td>
<td>71.93±4.64</td>
</tr>
<tr>
<td>CutMix</td>
<td>20.44±1.22</td>
<td>23.10±2.98</td>
<td>12.08±1.59</td>
<td>73.66±0.76</td>
</tr>
<tr>
<td>IRM</td>
<td>55.25±0.89</td>
<td>49.71±0.71</td>
<td>50.26±0.48</td>
<td>72.30±2.71</td>
</tr>
<tr>
<td>CGN</td>
<td>42.15±3.89</td>
<td>47.50±2.18</td>
<td>43.84±0.25</td>
<td>69.25±0.29</td>
</tr>
<tr>
<td>CONIC</td>
<td>65.57±0.34</td>
<td>92.41±0.26</td>
<td>77.72±1.00</td>
<td>79.56±1.28</td>
</tr>
</tbody>
</table>

We studied the adverse effects of confounding in observational data on the performance of a classifier. We showed the relationship between confounding and correlation in the causal processes considered, and we proposed a methodology to remove the correlation between the target variable and generative factors.
factors that works even when the dataset is highly confounded. Specifically, we proposed a counterfactual data augmentation method that systematically removes the confounding effect rather than addressing the confounding problem through random augmentations. Using the generated counterfactuals leads to substantial increase in a downstream classifier’s accuracy. That said, we observed that the counterfactual quality can still be improved, which will be interesting future work.

References


Appendix

A Proof of Propositions

Proposition 3.1. In causal processes of the form \(1(a)\), \(p(Z_i|\text{do}(Z_j))\) is equal to \(p(Z_i)\).

Proof. In causal processes of the form \(1(a)\), let \(C'\) denote the set of all confounding variables that are part of some backdoor path from \(Z_i\) to \(Z_j\). That is \(C' = \{ C| Z_i \leftarrow C \to Z_j \}\) for some \(i, j\). Then we can evaluate the quantity \(p(Z_i|\text{do}(Z_j))\) as

\[
    p(Z_i|\text{do}(Z_j)) = \sum_{C'} p(Z_i|Z_j, C') p(C') = \sum_{C'} p(Z_i|C') p(C') = \sum_{C'} p(Z_i, C') = p(Z_i)
\]

Where the first equality is because of the adjustment formula \(26\) and the second equality is because of the fact that \(Y\) is a collider in causal graph \(1(a)\) and hence conditioned on \(C'\), \(Z_i\) is independent of \(Z_j\).

Proposition 3.2. For causal generative processes of the form \(1(a)\), the correlation between a pair of generative factors \((Z_i, Z_j)\) is proportional to the amount of confounding between \(Z_i\) and \(Z_j\).

Proof. Expanding the quantity \(I(Z_i \rightarrow Z_j) + I(Z_j \rightarrow Z_i)\), we get the following,

\[
    \begin{align*}
    I(Z_i \rightarrow Z_j) + I(Z_j \rightarrow Z_i) &= \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i|Z_j)}{p(Z_i|\text{do}(Z_j))} \right) \right] + \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_j|Z_i)}{p(Z_j|\text{do}(Z_i))} \right) \right] \\
    &= \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i|Z_j)p(Z_j|Z_i)}{p(Z_i|\text{do}(Z_j))p(Z_j|\text{do}(Z_i))} \right) \right] = \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i|Z_j)p(Z_j|Z_i)}{p(Z_i)p(Z_j)} \right) \right] \\
    &= \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i|Z_j)p(Z_j|Z_i)}{p(Z_i)p(Z_j)} \right) \right] = \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i)p(Z_j)}{p(Z_i)p(Z_j)} \right) \right] \\
    &= 2 \times \mathbb{E}_{Z_i,Z_j} \left[ \log \left( \frac{p(Z_i)p(Z_j)}{p(Z_i)p(Z_j)} \right) \right] = 2 \times I(Z_i; Z_j)
    \end{align*}
\]

Where the third equality is due to Proposition \(3.1\). Since non-zero mutual information implies positive correlation, we see that the amount of confounding between \(Z_i\) and \(Z_j\) is directly proportional to the correlation between \(Z_i\) and \(Z_j\). Hence, we use the correlation as a measure of confounding between generative factors in the causal processes of the form \(1(a)\).

B Time Complexity Analysis:

Apart from its simple methodology, CONIC brings additional advantages in terms of computing time required to train the model that generates counterfactual images. As shown in Table 2, the time required to run our method to generate counterfactual images w.r.t. a generative factor \(Z_j\) is significantly less than CGN that learns deterministic causal mechanisms as discussed in Section \(2\). Even though we used CycleGAN in this work, for the cases where the number of generative factors are more, StarGAN \(5\) can be used to minimize the time required to learn the mappings from one domain to another domain \(89\) \(11\).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CONIC</th>
<th>CGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM-MNIST</td>
<td>2.76 ± 0.19</td>
<td>103 ± 1.50</td>
</tr>
<tr>
<td>DCM-MNIST</td>
<td>2.22 ± 0.01</td>
<td>103 ± 2.04</td>
</tr>
<tr>
<td>WLM-MNIST</td>
<td>1.22 ± 0.01</td>
<td>111 ± 2.50</td>
</tr>
</tbody>
</table>

Table 2: Run time (in minutes) of CONIC compared to CGN on MNIST variants
Figure 3: Left: sample thin morpho MNIST images and corresponding labels. Right: Sample thick morpho MNIST images and corresponding labels.

Figure 4: Top: CelebA original images of males with non-blond hair color. Bottom: Counterfactual images of males with blond hair generated using Algorithm 1.
Figure 5: Sample images from MNIST variants and augmented images by various methods.