# CAN: A CAUSAL ADVERSARIAL NETWORK FOR LEARNING OBSERVATIONAL AND INTERVENTIONAL DISTRIBUTIONS

A PREPRINT

**Raha Moraffah**
Department of Computer Science
Arizona State University
Tempe, AZ 85281
Raha.Moraffah@asu.edu

**Bahman Moraffah**
School of Electrical, Computer, and Energy Engineering
Arizona state university
Tempe AZ, 85287
Bahman.Moraffah@asu.edu

**Mansooreh Karami**
Department of Computer Science
Arizona State University
Tempe, AZ 85281
mkarami@asu.edu

**Adrienne Raglin**
Army Research Lab
2800 Powder Mill Rd
Adelphi, MD 20783
adrienne.raglin2.civ@mail.mil

**Huan Liu**
Department of Computer Science
Arizona State University
Tempe, AZ 85281
hliu@asu.edu

August 27, 2020

**ABSTRACT**

We propose a generative Causal Adversarial Network (CAN) for learning and sampling from observational (conditional) and interventional distributions. In contrast to the existing CausalGAN which requires the causal graph for the labels to be given, our proposed framework learns the causal relations from the data and generates samples accordingly. In addition to the relationships between labels, our model also learns the label-pixel and pixel-pixel dependencies and incorporate them in sample generation. The proposed CAN comprises a two-fold process namely Label Generation Network (LGN) and Conditional Image Generation Network (CIGN). The LGN is a novel GAN architecture which learns and samples from the causal graph over multi-categorical labels. The sampled labels are then fed to CIGN, a new conditional GAN architecture, which learns the relationships amongst labels and pixels and pixels themselves and generates samples based on them. This framework additionally provides an intervention mechanism which enables the model to generate samples from interventional distributions. We quantitatively and qualitatively assess the performance of CAN and empirically show that our model is able to generate both interventional and observational samples without having access to the causal graph for the application of face generation on CelebA data.

## 1 Introduction

Generative Adversarial Networks (GANs) [1] are ubiquitous tools for non-parametric sampling from complicated and high-dimensional distributions. GANs have achieved promising results in generating sharp-looking and realistic images and videos [2, 3]. They are also exploited to generate samples from categorical distributions [4] as well as text data [5]. One well-known extension of GAN is conditional GAN (cGAN) which enables sampling from conditional distributions. cGANs are designed to generate a random variable (e.g. image) given instances of another sets of random variables (e.g. labels). Several frameworks for cGAN with impressive performance have been proposed [6, 7]. Traditional cGANs assume the labels are independent and changing one label does not affect the distribution of other labels. However, this assumption does not hold in many real-world cases and in fact labels often have causal relationships with each other [8]. Ignoring such dependencies could result in generating unrealistic samples. However, questions such as *"what if the person had mustache or was bald?"* are interesting questions and could lead into generating interesting samples that do not belong to the observed data. In order to model causal relations and answer "what if?" questions, causal inference

provides powerful tools, i.e. Structural Equation Models (SEMs), as a way of encoding causal relationships, and intervention mechanisms [9]. Intervention on one variable is different from conditioning on it in the sense that the latter affects the distributions of both ancestors and descendants of the variable in the causal graph whereas the intervention only affects the distribution of descendants. For instance, given $gender \rightarrow mustache$ as a part of the causal graph, conditioning on mustache changes the distribution of the gender and thus, we only see males with mustache in the generated samples. Intervening on mustache on the other hand does not affect the gender and hence, we expect to see both males and females with mustache in our samples. the distribution resulted from intervening on some variables is called the interventional distribution. To consider dependencies between labels and generate samples from interventional distribution, CausalGAN [8] proposes an adversarial training framework to learn a causal generative model based on a given causal graph. Despite promising results, this framework requires the causal graph of the labels to be given which cannot be satisfied in most real-world cases. Moreover, CausalGAN only considers the causal relationships between the labels and does not take the label-pixel and pixel-pixel relations into account. Furthermore, since the label generator in CausalGAN contains one neural net per each node in the causal graph of the labels, the model cannot be scaled when the number of labels is big. To address these problems, we propose a novel generative **C**ausal **A**dvesarial **N**etwork (CAN) which aims to: 1) learn the causal relations between labels from the data instead of considering it to be given and generate labels accordingly; 2) learn the label-pixel and pixel-pixel relations and consider them in generating images and 3) generate samples from interventional and observational distributions. Particularly, CAN is a 2-fold framework which consists of a Label Generation Network, a novel GAN, and Conditional Image Generation Network, a novel cGAN. The Label Generation Network is trained to learn the causal graph over the labels from the data and generates samples from the learned graph. The labels are then fed to the Conditional Image Generation Network, an extension of AC-GAN [7] which is designed to take in a set of labels, learn the label-pixel and pixel-pixel relationships from the data and generate the images accordingly. We also propose an intervention mechanism which enables the framework to sample from both conditional and interventional distributions. This allows the framework to generate samples which may not exist in the data (e.g. images of bald women). Illustrative examples can be found in Figure 2. Our contributions can be summerized as follows: 1) We propose a novel architecture called generative Causal Adversarial Network, which is capable of generating high quality and diverse samples from observational and interventional distributions without requiring the causal graph to be known. Specifically, the model learns the label-label, label-pixel and pixel-pixel causal dependencies from the data and generate samples based on the learned relationships; 2) We propose an intervention mechanism for our proposed framework which enables generating samples from interventional distributions using generative adversarial networks; and 3) Our extensive quantitative and qualitative experiments show the superiority of our model over existing cGANs as well as the existing CausalGAN. We also demonstrate sampling from interventional distribution through a face generation application on CelebA data.

## 2 Causal Adversarial Network (CAN) Framework

In this section, we introduce our proposed generative **C**ausal **A**dversarial **N**etwork (CAN), which aims to: 1) learn the label-label, label-pixel and pixel-pixel causal dependencies and generate samples according to them; 2) enable sampling from both observational (conditional) and interventional distributions. Our proposed framework consists of a Label Generation Network (LGN) and a Conditional Image Generation Network (CIGN). The proposed LGN is a GAN-based framework which learns the causal graph over the labels from the data and samples from it given a noise input. The sampled labels along with a random noise are then fed to the CIGN, a novel extension of cGAN which learns the causal relations between labels and pixels and pixels themselves and generates samples accordingly. To this end, we propose to modify the GAN's generator by integrating the Structural Equation Model (SEM) of the data into the generator and generating samples from it. The parameters of the SEM, the generator and discriminator are learned via adversarial training.

### 2.1 Proposed framework

In this section, we demonstrate the general idea of embedding the SEM in the GAN's architecture using one of the widely used variant of GAN, WGAN-GP [10] and introduce an intervention mechanism for this framework. We then describe how this idea contributes to components of CAN and how these components can be combined to form CAN.

#### 2.1.1 WGAN-GP

A Generative Adversarial Network (GAN) [1] consists of two neural networks namely generator and discriminator (critic) competing against each other. The generator takes in a random noise vector $z$ and generates a fake sample. The discriminator receives a real or a fake sample and determines whether it is synthesized by the generator or drawn from the real distribution. Different loss metrics have been proposed for GANs. For instance, Wasserstein-GAN (WGAN) [11] uses Earth-Mover (a.k.a Wasserstein-1) distance to compare the real and generated distributions. This metric is

specifically suitable due to its convergence properties and correlation with the perceptual quality of generated images. In this paper, we adopt an improved and more stable version of WGAN, WGAN-GP, which enforces the Lipschitz constraint required by the objective of WGAN by adding a gradient penalty term. The discriminator and generator losses of WGAN-GP are respectively defined as:

$$\begin{aligned}\mathcal{L}_D &= \underbrace{\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)]}_{\text{Critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x}) - 1\|)^2]}_{\text{gradient penalty}}, \\ \mathcal{L}_G &= -\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})],\end{aligned} \tag{1}$$

where D is a set of 1-Lipschitz functions, $\mathbb{P}_r$ and $\mathbb{P}_g$ denote the real and model distributions, and $\mathbb{P}_{\hat{x}}$ is a distribution obtained by randomly interpolating between real and generated images. The generator learned via this objective is a deterministic transformation from an easy-to-sample independent distribution to the target random variable. Once the parameters are learned, the generator can be used to perform non-parametric sampling from marginal distributions $P(x)$ with this generative process $\tilde{x} = G(z;\theta_g)\,, z \sim p(z)$. The $G(z;\theta_g)$ is the generator parameterized with a neural network with parameters $\theta_g$ and trained via adversarial training.The graphical model of this process is defined as $z \rightarrow x$ which assumes that every variable in $x$ depends on every variable of $z$. However, in reality, only a few variables in z affect each variable of x. To discover these relationships from the data and generate samples based on them, we propose to learn causal relationships between variables of x by embedding a Structural Equation Model (SEM) in the generator's structure, learn its parameters along with other parameters of the model and generate sample using ancestral sampling.

### 2.1.2 Integrating SEM into the GAN's Generator

In this section, we explain the process of integrating the SEM into the GAN's generator by first formally introducing the SEM and then explaining the process of embedding it into GAN.

Let $X \in \mathbb{R}^n$ be a sample from the joint distribution of n variables and $A \in \mathbb{R}^{n\times n}$ be a weighted adjacency matrix of the DAG in which each node corresponds to a variable in X. The linear Structural Equation Model (SEM) is defined as:

$$X = A^T X + Z$$

where $Z \in \mathbb{R}^n$ is a random noise. In other words, in a SEM, each variable is defined as a (linear) function of its parents in the causal DAG and a noise variable. The child-parent relationships are encoded via the adjacency matrix $A$. If the causal DAG is sorted in topological order, ancestral sampling is done by first sampling a random noise Z and then solving the following triangular system:

$$X = (I - A^T)^{-1} Z$$

This equation is a linear deterministic transformation from $Z$ to $X$. GANs generators are often non-linear transformations of noise. To add non-linearity, inspired by recent work on learning non-linear SEM [12, 13], we propose to replace traditional generator ($G(Z;\theta_g)$) with the following:

$$X = G((I - A^T)^{-1} Z; \theta_g, A) \tag{2}$$

where G is the generator in the traditional GAN and is instantiated based on the type of data. In this paper, we call G the non-linear transformation function. Equation (2) demonstrates a mapping from noise to data space by taking causal relations into account. The parameter A is simultaneously learned along with all other parameters of the model via adversarial training.

Optimizing the adversarial loss does not guarantee the learned A to be a DAG as required by SEMs. A graph needs to satisfy the acyclicity constraint to be a DAG. To satisfy this condition, we impose an equality constraint which guarantees that a graph is acyclic if and only if for any $\beta > 0$ [12]:

$$tr[(I + \beta A \odot A)^n] - n = 0 \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$ is the adjacency matrix of the graph, n is number of nodes, "tr" represents trace of a matrix and $\odot$ is element-wise multiplication.

We therefore combine this equality constraint with our adversarial loss. The objectives of the modified WGAN-GP with embedded SEM are given as:

$$\begin{aligned}\mathcal{L}_D &= \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x}) - 1\|)^2], \\ \mathcal{L}_G &= -\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] = -\mathop{\mathbb{E}}_{z\sim\mathbb{P}_z}[D(G(z;\theta,A))] \\ \text{s.t.}\quad & tr[(I + \beta A \odot A)^n] - n = 0\end{aligned} \tag{4}$$

This objective is solved with augmented Lagrangian approach [14]. More details can be found in the Appendix. In the following, we propose an intervention mechanism for our designed framework.

### 2.1.3 Intervention Mechanism for Sampling from Interventional Distributions

In a SEM, each variable $x_i$ can be written as a deterministic function of its parents ($pa(x_i)$) and a noise variable ($z_i$). Intervention on variable $x_i = f_i(pa(x_i), z_i)$ in the system is accomplished by replacing the function $f_i$ with the desired value where the rest of the system remain unchanged [9]. In terms of graphical models, this is equivalent to removing all incoming edges to the node in the causal DAG and replacing the value of that node with the interventional value. Doing so makes the distribution of the ancestors of the node unchanged and only changes the distribution of the descendants. To model this process, we consider two sources of input for each node in the causal DAG: 1)values of the parent nodes and a noise variable and 2)interventional value for that node. In the case of generating purely observational data, the values of parents and a random noise are sampled and used to calculate the value of the node using the equation $x_i = f_i(pa(x_i), z_i)$. Whereas, in the case of sampling from interventional distribution, the interventional value is selected as the value of the node and that value is propagated to the descendants of that node in the graph. This selection process is implemented via a mask vector $\alpha_1$ which performs as selector and for each node either selects the value of parents, their corresponding weights in the SEM and a random noise and calculates the final value the node using them or selects the interventional value and sets it as the value of node. More formally, we extend our mapping function defined in equation (2) to enable sampling from both observational and interventional distributions as follows:

$$X = G((I - \alpha_1 \odot A^T)^{-1}(\alpha_1 \odot Z + (1 - \alpha_1) \odot C)) \tag{5}$$

where $C \in \mathbb{R}^n$ is a vector of interventional values for nodes in the causal DAG where $C_i$ corresponds to the desired interventional value for the $i - th$ node of the graph and $\alpha_1 \in \mathbb{R}^n$ is a selector mask which selects source of inputs for each node in the graph. If $\alpha_{1i} = 0$ then $x_i = f_i(pa(x_i), z_i)$ and if $\alpha_{1i} = 1$ then $x_i$ is forced to be equal to $C_i$, i.e., $x_i = C_i$.

It is worth mentioning that we use the intervention mechanism to sample from interventional distribution at inference time when the SEM has already been learned. Now, we discuss each component of the CAN framework and how the method proposed in section 2.1.2 can be used to model them.

### 2.1.4 Label Generation Network

We introduce the Label Generation Network (LGN) whose aim is to learn the causal dependencies between the labels and generate samples based on them. To achieve this, We directly use the generator proposed in equation (2). However, the generator non-linear transformation function (G), the discriminator (D) and the loss criterion still need to be designed according to the type of the data the GAN is designed to generate.

Labels in commonly used datasets are often of discrete nature with multiple categorical variables. Therefore the choice of $G$, $D$ and the objective loss need to be suitable for this data type. For the G, we employ the architecture proposed in [4] for multi-categorical variables. The architecture consists of multiple fully connected layers followed by a layer with one fully connected output layer per each categorical variable (i.e. label) in the sample. A softmax activation is then applied on top of this layer. Finally, the outputs of the softmax are concatenated with each other and generate the final output of the generator. The discriminator is a set of fully connected layers. For the adversarial loss, we utilize WGAN-GP's loss, which has proven to be capable of generating discrete samples using continuous generators [10]. We learn the parameters of LGN by optimizing the adversarial loss in equation (4).

### 2.1.5 Conditional Image Generation Network

We propose a novel cGAN architecture called Conditional Image Generation Network (CIGN). In traditional cGANs, labels and random noise are simply concatenated and fed to the generator to synthesize fake samples. These models assume all labels affect all pixels in the image. This assumption is not true in reality as labels only affect specific pixels in the image. To learn these relations and consider them while generating samples, we propose a new architecture of cGAN's generator such that the new model is able to learn the label-pixel and pixel-pixel causal dependencies. While learning pixel-pixel criterion can be accomplished by directly using the framework in section 2.1.2, that model is specifically designed for sampling from marginal distributions and does not support conditioning on additional information (labels). To address the issue, we model conditioning with our intervention mechanism. Despite differences between conditioning and intervening on a variable, i.e, conditioning affects the distributions of both ancestors and descendants and intervening only affects the descendants in the causal graph, these two concepts operate the same for the variables who have no ancestors in the graph. Since in image generation process the assumption is that labels cause the image, $L \rightarrow G$, the labels are the ancestors of the pixels of the image and conditioning on them is equivalent to intervention. This subtle observation allows us to use the framework in section 2.1.2 for cGAN generator with minor modifications. Particularly, In the generator, we have a causal graph of labels and pixels, but assume the values of labels in the that graph are already given to the model by intervening on them. Hence, we only need to learn the label-pixel and pixel-pixel relations. For the choice of G, broadly speaking, the network comprises a series of 'deconvolution'

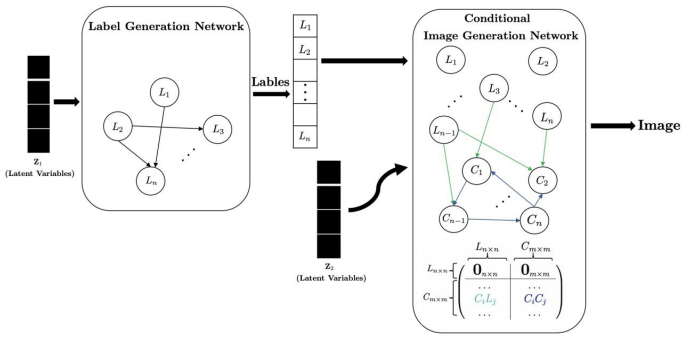


Figure 1: CAN framework at a glance: The framework consists of a LGN followed by CIGN. The LGN learns the causal relations between labels and samples from them. The labels are fed to CIGN which learns the label-pixel and pixel-pixel relations to generate samples.

layers and the discriminator network is a series of 'convolutional' layers. In order to learn the parameters, we modify a variation of AC-GAN's objective based on WGAN-GP loss to support multiple labels. Formally the objectives of our proposed CIGN are defined as:

$$\begin{aligned}
\mathcal{L}_D &= \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x}) - 1\|)^2] - \mathcal{L}_C, \\
\mathcal{L}_G &= -\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathcal{L}_C = -\mathop{\mathbb{E}}_{z\sim\mathbb{P}_z}[D(G(z;\theta,A))] - \mathcal{L}_C \\
\text{s.t.} \quad & tr[(I + \beta A \odot A)^n] - n = 0
\end{aligned} \tag{6}$$

where $\mathcal{L}_C$ is the log-likelihood of the correct class defined as $\mathcal{L}_C = \mathbb{E}[\log P(C = c\,|X_{\text{real}})] + \mathbb{E}[\log P(C = c\,|X_{\text{fake}})]$. These objective are solved via adversarial training.

More details on the architecture of both LGN and CIGN are discussed in the Appendix. Finally, the LGN and CIGN combined together are called CAN. The labels are first sampled from their observational or interventional distributions via LGN. The sampled labels are then fed to the CIGN and generate samples given the labels. Figure 1 displays an overview of our proposed CAN framework.

## 3 Experiment

In this section, we evaluate the performance of the proposed CAN from both quantitative and qualitative perspectives. In our experiments, we aim to answer the following questions: 1) Is CAN capable of generating samples from both conditional and interventional distributions?; 2) Are the images generated by CAN diverse, of high quality and close to the real distribution?; 3) Are labels generated by LGN component of CAN high quality?; and 4) Is CAN able to learn the causal graph from the data? To answer these, we perform three main types of experiments: image generation evaluation, label generation evaluation, and validation of the causal graph learned by the CAN. More details on the experimental settings can be found in the Appendix.

### 3.1 Image Generation Evaluation

Here, we seek to answer the first two experimental questions, i.e. examining CAN's ability to generate samples from observational and interventional distributions and assessing the quality of the (observational) images generated by CAN. Evaluating the performance of GANs is a challenging task [15]. Following state-of-the-arts [7, 8, 2], we assess the performance of our proposed framework through both qualitative and quantitative evaluations. To demonstrate the effectiveness of CAN, we train the model on CelebA, a large-scale face attributes dataset [16], which includes 202,599 images of faces of celebrities along with 40 attribute annotations per each image. Even though CAN is capable of training on all 40 attributes, o make a fair comparison with previous work [8], the results are presented on 9 selected labels namely "Bald","Eyeglasses","Male","Mouth-Slightly-Open","Mustache","Narrow-Eyes","Smiling","Wearing-Lipstick", and "Young".

**Qualitative Evaluation of CAN**

Here we verify the ability of CAN in generating samples from both interventional and observational distributions in Figure 2. The difference between conditional and interventional distributions are justified by adding/removing certain

(a) (b)

Figure 2: Results of CAN for intervening and conditioning on labels. Top two rows show the results for intervention (i.e. intervention on label=0 and label=1). Bottom two rows are the results for conditioning (i.e. condition on label=0 and label=1) : (a) shows the results for "Bald" label. Since the causal graph is expected to be $Male \rightarrow Bald$, in the interventional samples we have both bald males and females. However, in the conditional samples, only bald males can be found. (b) shows the results on label "Mustache". The results are in compliance with the expected causal graph $Male \rightarrow Mustache$. Therefore, in the interventional samples we have both males and females with mustache. However, in the conditional samples, only males with mustache can be found.

feature to the image through intervention or conditioning. Note that conditional sampling from labels is implemented through rejection sampling. Due to the space limitation, we only showcase our results for two labels, i.e., *Bald* and *Mustache*. The results for all other labels can be found in the Appendix.

**Quantitative Evaluation of CAN**

In this experiment, we quantitatively asses the quality of the (observational) images generated by CAN and compare them with the following baselines: 1) AC-GAN [7]: The state-of-the-art cGAN, where the discriminator is designed to predict the class label as well as authenticity of an image. AC-GAN discards the label-label, label-pixel and pixel-pixels causal relations. In our experiments we implement the AC-GAN withWGAN-GP loss and modify the architecture to accept multiple labels; 2) CausalGAN [8]: The state-of-the-art GAN designed for sampling from both interventional and conditional distributions based on a known causal graph. CausalGAN only considers causal relations between labels. CausalGAN requires the causal graph over labels to be know and ignore label-pixel and pixel-pixel relations.

In order to measure the visual quality of synthesized images, we calculate Fréchet Inception distance (FID) which compares the activations between real and fake samples [17]. However, FID measures the quality of samples from marginal distributions and is not specifically designed for conditional GANs. Since our method can be considered as an extension of cGAN, we utilize GAN-train and GAN-test metrics proposed by [18] which are specifically designed to evaluate cGANs. These two metrics are based on classification accuracy and approximate the recall (diversity) and precision (quality of the image) of cGANs, respectively. Since these metrics are originally proposed for the case where each image exactly has one label, to be able to apply them to our model, we extend them by using Hamming score (a.k.a. label-based accuracy) which is widely used for multi-label classification instead of accuracy. We also report the GAN-train and GAN-test for each label individually along with details on these metrics calculations in the Appendix. Table 1 illustrates the FID, GAN-train and GAN-test scores achieved by CAN and the baselines. These results show in addition to being capable of generating samples from interventional and observational distributions, the CAN framework is also able to generate high quality and diverse images and outperforms the baselines.

Table 1: The FID, GAN-train and GAN-test scores for CAN and baselines. Analyzing the FID (lower is better) and GAN-train and GAN-test (higher is better) demonstrates the effectiveness of CAN

| Model | FID | GAN-train | GAN-test |
|---|---|---|---|
| CAN | 4.95 | 0.65 | 0.62 |
| CausalGAN | 20.32 | 0.58 | 0.45 |
| AC-GAN | 12.58 | 0.60 | 0.42 |

Table 2: Evaluation of the quality of labels on Child dataset

| | Child | | |
|---|---|---|---|
| Model | $\text{MSE}_p$ | $\text{MSE}_f$ | $\text{MSE}_a$ |
| CAN | $5.1 \times 10^{-4} \pm 8 \times 10^{-5}$ | $1.4 \times 10^{-3} \pm 4 \times 10^{-4}$ | $3.4 \times 10^{-4} \pm 7 \times 10^{-5}$ |
| MC-WGAN-GP | $9.8 \times 10^{-4} \pm 5 \times 10^{-5}$ | $1.8 \times 10^{-3} \pm 4 \times 10^{-4}$ | $4.2 \times 10^{-4} \pm 8 \times 10^{-5}$ |

Table 3: Evaluation of the quality of labels on CelebA-Label dataset

| | CelebA-Label | | |
|---|---|---|---|
| Model | $\text{MSE}_p$ | $\text{MSE}_f$ | $\text{MSE}_a$ |
| CAN | $6.9 \times 10^{-4} \pm 10^{-5}$ | $2.2 \times 10^{-4} \pm 5 \times 10^{-5}$ | $1.4 \times 10^{-5} \pm 3 \times 10^{-5}$ |
| MC-WGAN-GP | $6 \times 10^{-4} \pm 10^{-5}$ | $6 \times 10^{-4} \pm 10^{-4}$ | $1.3 \times 10^{-5} \pm 3 \times 10^{-5}$ |

### 3.2 Label Generation Evaluation

In this section, we evaluate the quality of generated labels by assessing the performance of LGN. We split the data into 90% training and 10% test. We utilize three Mean Squared Error (MSE) based metrics, i.e. $\text{MSE}_p = \text{MSE}(p_{\text{test}}, p_{\text{sample}})$, $\text{MSE}_f = \text{MSE}(f_{\text{train}}, f_{\text{sample}})$ and $\text{MSE}_a = \text{MSE}(a_{\text{train}}, a_{\text{sample}})$ used in [19, 4] to evaluate the generated multi-categorical samples. In these metrics, $p_{\text{test}}$ and $p_{\text{sample}}$ are vectors of frequencies of ones corresponding to real samples in the test and generated samples, respectively. $f_{\text{train}}$ and $f_{\text{sample}}$ are the f-1 scores of prediction of each dimension of the vector representing a sample using a logistic regression model trained on the real train set and generated samples. $a_{\text{train}}$ and $a_{\text{sample}}$ are the accuracy of predicting each categorical variable in the multi-categorical sample using real train and generated samples. We compare our model with MC-WGAN-GP [4], a GAN for multi categorical data. This model discards the causal relations. We demonstrate our results on three datasets with multiple categorical variables : Child and Alram datasets [20] and All 9 labels used previously from CelebA data. We report the results for Child and Celeba-label data in tables 2 and 3 and the results for Alarm are shown in the Appendix.

### 3.3 Validating the causal graph learned by the model

Since the performance of the CAN highly depends on the quality of the causal graph learned by the model from the data, in this section we evaluate the causal graph generated by CAN on two commonly used datasets for causal discovery: Child and Alram [20]. We utilize traditional metrics for structure learning, i.e., structural Hamming distance (SHD) and number of edges discovered [21, 12], and report the numbers. Since these datasets are discrete, we use LGN to learn the causal DAG. We compare our results with DAG-GNN [12], a variational autoencoder parameterized by a graph neural network for causal structure learning. As shown in table 4, CAN outperforms the baseline in most of the cases, which indicates the capability of the model in learning causal graph.

Table 4: Results on child and alarm dataset: SHD (the lower the better) and number of edges are calculated with respect to the groundtruth

| | Child | | Alarm | |
|---|---|---|---|---|
| Model | SHD | #Edges | SHD | #Edges |
| CAN | 241 | 228 | 61 | 42 |
| DAG-GNN | 242 | 225 | 65 | 48 |

## 4 Related work

Conditional GAN (cGAN) is a type of GAN which allows conditional image generation by incorporating additional information such as class labels into both generator and discriminator. The first cGAN proposes to feed labels and images to both generator and discriminator [6]. Different variations of cGAN have been proposed to improve the conditional image generation. AC-GAN [7] proposes to generate high resolution images by modifying the discriminator to predict

the class label of the images as well as their authenticity. In [22] authors propose a projection based discriminator to consider the side information in the underlying probabilistic model. In another attempt to combine labels with images, authors propose to learn the joint distribution of labels and images [23, 24]. However, All aforementioned frameworks assume the labels are independent. To address this issue, CausalGAN [8] proposes an adversarial training procedure to generate conditional images given a causal graph for the labels. Despite considering the causal relations between the labels, the performance of CausalGAN is limited by the requirement of knowing the causal graph. Moreover, CausalGAN only considers the causal relations between labels and discard the label-pixel and pixel-pixel relations. Causal discovery is the task of identifying the causal relationships between a set of variables. With the surge of continuous constraints for learning DAGs [21, 12], neural networks have been leveraged to perform causal discovery from observational data. For instance, Yu et al. [12] propose a graph neural network based model to recover the causal DAG. Other works also utilize neural nets to discover causal DAGs by exploiting the non-linear relations between variables [13], using reinforcement learning to search for the best fitting DAG [25] and proposing a mask gradient-based approach to learn the DAG [26]. In [27], authors propose to leverage the cGAN to discover the direction of causal relationships between only two variables. Causality has been also used to improve the deep neural nets. In [28], authors explore the connection between GANs and causal generative models by considering the image as a cause of neural network's weights. In [29], Authors propose a neural net based approach to discover the causal relations between a label and a static image by discovering the causal relations directions. Non of these works use causality to generate interventional and conditional images.

## 5 Conclusions

In this paper, we propose a generative Causal Adversarial Network (CAN) which learns the label-label, label-pixel and pixel-pixel causal relations from the data and generates samples accordingly. The framework consists of a LGN and CIGN. The labels sampled from LGN are fed to the CIGN and together they can be used to generate samples from observational and interventional distributions. We validate the performance of our model through comprehensive experiments.

## Appendix

### A.1. Optimization Algorithm using Augmented Lagrangian Multiplier

In order to solve the objective described in equation (4) section 2.1.2 of the main text, we leverage augmented Lagrangian approach [14] which is widely used to solve the nonlinear equality-constrained problems. We define the Augmented Lagrangian for equation (4) as:

$$\begin{aligned}\mathcal{L}_D &= \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x}) - 1\|)^2], \\ \mathcal{L}_G &= - \mathop{\mathbb{E}}_{z\sim\mathbb{P}_z}[D(G(z;\theta,A))] + \bar{\lambda} c(A) + \frac{\rho}{2}\left\|c(A)\right\|^2,\end{aligned} \tag{7}$$

where $c(A) = tr[(I + \beta A \odot A)^n] - n$ is the equality constraint and $\bar{\lambda}$ and $\rho > 0$ are the Lagrange multiplier and the quadratic penalty weight, respectively. Note that we only impose the constraint when training the generator. We alternate between optimizing the discriminator and the generator. To optimize the objectives in the aforementioned equation, we use stochastic optimization solvers such as ADAM optimizer [30] or RMSprop [31].

We update $\bar{\lambda}$ in each step with the following rule:

$$\bar{\lambda} = \bar{\lambda} + \rho c(A)$$

### A.2. LGN and CIGN Network Architectures

In this section, we describe the details of the architectures of G (the nonlinear transformation function used in the generator) and D (discriminator) for the models trained on each dataset.

For all experiments, we use ADAM [30] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to optimize the LGN network and RMSprop [31] to optimize the CIGN. The learning rate for both generator and discriminator in LGN is 0.001. The learning rates for the generator and discriminator of CIGN are set to 0.001 and 0.0002, respectively. All LGN networks are trained for 250 epochs and the CIGN network is trained for 300 epochs. We set the hyperparameter $\lambda$ (gradient penalty coefficient) to 1. It is worth mentioning that the dimension of the input noise variable to the generator (z) is not necessarily the

same as the dimension of the data and could be much smaller. Moreover, in both generator and discriminator of LGN network, each categorical variable is represented as a one-hot encoded vector and each sample, a multi-categorical variable, consists of multiple one-hot encoded vectors concatenated with each other. For the generators, we mostly use normal rectified linear units (ReLU) as nonlinearity and for the discriminators, we utilize leaky rectified linear units (LReLU) [32] with the negative slope 0.2.

In what follows, we thoroughly discuss the details of the architectures of the models trained on each dataset. Table 5 summarizes the notations we used to descibe the architectures of LGN and CIGN.

Table 5: Notations used to describe the architectures of LGN and CIGN

| **Notation** | **Definition** |
|---|---|
| DECONV | Deconvolutional layer |
| CONV | Standard convolutional layer |
| FC | Fully-connected layer |
| N | Number of output channels |
| K | Kernel size |
| S | Stride size |
| P | Padding size |
| BN | Batch normalization |

### A.2.1. CelebA Dataset

**CelebA Data Preprocessing:** The CelebFaces Attributes dataset [16] is a dataset of 202,599 faces of celebrities annotated with 40 attributes. To be consistent with the previous work [8], we select the following 9 attributes to perform our experiments on: "Bald","Eyeglasses","Male","Mouth-Slightly-Open","Mustache","Narrow-Eyes","Smiling","Wearing-Lipstick", and "Young". We preprocess the images by first cropping the $178 \times 218$ images to $178 \times 178$ and then resizing them to $64 \times 64$.

**Network Architecture:** Table 6 and 7 show the architectures of LGN and CIGN trained on CelebA data. For the CIGN Network, we input a 9 dimensional vector containing the class labels along with 128 noise variables, which together results in a 137 dimensional input. For the input to the LGN, we use 9 noise variables.

Table 6: LGN Network Architecture for CelebA

| **Discriminator** D | **Generator** G |
|---|---|
| Input $\in \mathbb{R}^{18}$ | Input $\in \mathbb{R}^{9}$ |
| FC-(N100), LReLU(0.2) | FC-(N100), ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N1) | FC-(N100), BN, ReLU |
| | 9×FC-(N2), BN, ReLU |
| | 9× Softmax |

### A.2.2. Child and Alarm Datasets

**Child Dataset:** Child data [20] is a commonly used dataset in causal discovery. This dataset consists of 5000 multi-categorical samples where each sample comprises 20 categorical features. Each category is in the range of 1-5.

**Alarm Dataset:** A widely used multi-categorical data in causal discovery with 5000 samples. Each sample consists of 37 categorical features and each category ranges from 1 to 3.

Table 7: CIGN Network Architecture for CelebA

| **Discriminator** D | **Generator** G |
|---|---|
| Input $64 \times 64$ Color image | Input $\in \mathbb{R}^{137}$ |
| CONV-(N64, K$4 \times 4$, S2, P1), LReLU(0.2) | FC-(N1024), BN, ReLU |
| CONV-(N128, K$4 \times 4$, S2, P1), BN, LReLU(0.2) | FC-(N$128 \times 16 \times 16$), BN, ReLU |
| FC-(N1024), BN, LReLU(0.2) | DECONV-(N64, K$4 \times 4$, S2, P1), BN, ReLU |
| GAN-disc: FC-(N1)<br>Class-disc: FC-(N9) | DECONV-(N3, K$4 \times 4$, S2, P1), Tanh |

Table 8: LGN Network Architecture for Child dataset

| **Discriminator** D | **Generator** G |
|---|---|
| Input $\in \mathbb{R}^{60}$ | Input $\in \mathbb{R}^{20}$ |
| FC-(N100), LReLU(0.2) | FC-(N100), ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | $8\times$ FC-(N2)<br>$8\times$ FC-(N3)<br>$1\times$ FC-(N4)<br>$2\times$ FC-(N5)<br>$1\times$ FC-(N6) |
| FC-(N1) | $20\times$ Softmax |

Table 9: LGN Network Architecture for Alarm dataset

| **Discriminator** D | **Generator** G |
|---|---|
| Input $\in \mathbb{R}^{105}$ | Input $\in \mathbb{R}^{37}$ |
| FC-(N100), LReLU(0.2) | FC-(N100), ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | FC-(N100), BN, ReLU |
| FC-(N100), LReLU(0.2) | $13\times$ FC-(N2)<br>$17\times$ FC-(N3)<br>$7\times$ FC-(N4) |
| FC-(N1) | $37\times$ Softmax |

(a) (b)

Figure 3: Results of CAN for intervening and conditioning on labels : (a) shows the results for "Wearing-Lipstick" label. Since we expect to have $Male \rightarrow Wearing-Lipstick$, we do not expect intervening on Wearing-Lipstick to affect the distribution of Male, i.e., $\mathbb{P}$(Male $= 1$|do(Wearing-Lipstick=1)) $= \mathbb{P}$(Male $= 1$) $= 0.41$. Thus, in the interventional samples we observe both males and females wearing lipstick. On the other hand, conditioning on Wearing-Lipstick affects the distribution of Male, i.e., $\mathbb{P}$(Male $= 1$|Wearing-Lipstick=1) $\approx 0$. Hence, in the conditional samples, only females with lipstick can be found. (b) shows the results for "Mustache" label. Since in the learned causal graph we have $Wearing - Lipstick \rightarrow Mustache$, intervening on Mustache should not affect the distribution of Wearing-Lipstick, i.e., $\mathbb{P}$(Wearing-Lipstick $= 1$|do(Mustache=1)) $= \mathbb{P}$(Wearing-Lipstick $= 1$) $= 0.47$. This explains the existence of people with mustache both with and without lipstick in the interventional samples. Note that conditioning on Mustache affects the distribution of Wearing-Lipstick, i.e., $\mathbb{P}$(Wearing-Lipstick $= 1$|Mustache=1) $\approx 0$. Hence, no person with mustache is seen wearing a lipstick in the conditional samples.

(a) (b)

Figure 4: Results of CAN for intervening and conditioning on labels : (a) shows the results for "Bald" label. According to the causal graph, we have $Young \rightarrow Bald$, therefore, intervening on Bald does not affect Young, i.e., $\mathbb{P}$(Young $=$ 1|do(Bald=1)) $= \mathbb{P}$(Young $= 1$) $= 0.77$. However, conditioning on Bald changes the $\mathbb{P}$(Young $= 1$) from 0.77 to 0.23. (b) shows the results for "Mustache" label. Since in the learned causal graph we have $Male \rightarrow Mustache$, intervening on Mustache should not affect the distribution of Male, i.e., $\mathbb{P}$(Male $= 1$|do(Mustache=1)) $= \mathbb{P}$(Male $= 1$) $= 0.41$. However, we have $\mathbb{P}$(Male $= 1$|Mustache=1) $\approx 1$), which means the conditional samples should only contain males whereas in the interventional samples we have both males and females with mustache.

**Network Architecture:** In our experiments, we only train the LGN Network on Child and Alarm datasets. Table 8 and 9 illustrate the architecture of the LGN Network trained on Child and Alarm data, respectively. We use 20 noise variables as input to the LGN for Child data. The input to the LGN trained on Alarm data consists of 37 noise variables.

## A.3. Additional Experimental Results

### A.3.1. Additional Qualitative Results

In this section, we present additional images generated by CAN from both observational (conditional) and interventional distributions in Figures 3 and 4. Note that in these figures, top two rows show the results for intervention (i.e., intervention on label=0 and label=1, respectively) and bottom two rows are the results for conditioning (i.e., condition on label=0 and label=1, respectively).

Table 10: The GAN-train and GAN-test scores for CAN and baselines for all 9 labels in CelebA separately. Analyzing GAN-train and GAN-test (higher is better) demonstrates the effectiveness of CAN

| Label | Model | GAN-train | GAN-test |
|---|---|---|---|
| Bald | CAN | 97.72 | 87.67 |
| | CausalGAN | 91.11 | 59.43 |
| | AC-GAN | 91.15 | 57.96 |
| Eyeglasses | CAN | 96.01 | 91.62 |
| | CausalGAN | 92.31 | 87.96 |
| | AC-GAN | 95.24 | 87.05 |
| Male | CAN | 86.56 | 88.12 |
| | CausalGAN | 85.24 | 76.40 |
| | AC-GAN | 87.00 | 70.70 |
| Mouth-Slightly-Open | CAN | 79.26 | 80.00 |
| | CausalGAN | 75.43 | 72.36 |
| | AC-GAN | 79.41 | 70.85 |
| Mustache | CAN | 90.15 | 90.74 |
| | CausalGAN | 82.66 | 59.32 |
| | AC-GAN | 88.58 | 57.98 |
| Narrow-Eyes | CAN | 82.33 | 82.90 |
| | CausalGAN | 61.14 | 50.23 |
| | AC-GAN | 69.01 | 52.51 |
| Smiling | CAN | 82.86 | 86.28 |
| | CausalGAN | 71.83 | 72.41 |
| | AC-GAN | 78.55 | 71.26 |
| Wearing-Lipstick | CAN | 80.07 | 85.50 |
| | CausalGAN | 72.30 | 59.96 |
| | AC-GAN | 71.09 | 55.52 |
| Young | CAN | 67.37 | 64.34 |
| | CausalGAN | 65.93 | 63.14 |
| | AC-GAN | 76.39 | 61.59 |

### A.3.2.Quantitative Evaluation: Evaluation Metrics Details

**GAN-train and GAN-test:** In this section, we explain the details of the two metrics, namely GAN-train and GAN-test, which are used to evaluate the proposed CIGN network. More specifically, to calculate GAN-train, a classification network is trained with images generated by the GAN model, and then its performance is evaluated on a test set of real-world images. To calculate the GAN-test, on the other hand, a classification network is trained on the real-world data and tested on the images generated by the GAN. In our experiments, for the classification network, we train Resnet-18 [33] with learning rate 0.001 for 20 epochs. We also extend the original architecture to perform multi-label classification. Since our results are for multiple labels, we report both the hamming score, a label-based accuracy metric for multi-label classification (Table 1 in the main text) as well as classification accuracy for each label. The classification accuracy per each label is summerized in table 10. The results show that our proposed CAN outperforms both baselines, i.e. CausalGAN and AC-GAN and demonstrate the effectiveness of our model in generating high-quality and diverse images compared to the baselines which do not consider the causal relations amongst pixels and labels .

### A.3.3. Label Generation Evaluation: Alarm Data Results

Results for the evaluation of samples generated for Alarm data by LGN are shown is table 11. The results show that CAN outperforms MC-WGAN-GP which does not consider the causal relations between features of a sample and generates high-quality samples.

Table 11: Evaluation of the quality of labels on Alarm dataset

| | Alarm | | |
|---|---|---|---|
| Model | $\text{MSE}_p$ | $\text{MSE}_f$ | $\text{MSE}_a$ |
| CAN | $4.1 \times 10^{-4} \pm 2 \times 10^{-5}$ | $4.9 \times 10^{-3} \pm 2 \times 10^{-3}$ | $3.3 \times 10^{-4} \pm 2 \times 10^{-5}$ |
| MC-WGAN-GP | $4.8 \times 10^{-4} \pm 5 \times 10^{-5}$ | $7.1 \times 10^{-3} \pm 2 \times 10^{-3}$ | $6.3 \times 10^{-4} \pm 7 \times 10^{-5}$ |

## References


[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[4] Ramiro Camino, Christian Hammerschmidt, and Radu State. Generating multi-categorical samples with generative adversarial networks. *arXiv preprint arXiv:1807.01202*, 2018.

[5] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[6] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[7] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[8] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

[9] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[11] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[12] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. *arXiv preprint arXiv:1904.10098*, 2019.

[13] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

[14] Dimitri P Bertsekas. Nonlinear programming. athena scientific. *Belmont, MA*, 1999.

[15] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[18] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2018.

[19] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

[20] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[21] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, pages 9472–9483, 2018.

[22] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[23] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[24] Yunchen Pu, Shuyang Dai, Zhe Gan, Weiyao Wang, Guoyin Wang, Yizhe Zhang, Ricardo Henao, and Lawrence Carin. Jointgan: Multi-domain joint distribution learning with generative adversarial nets. *arXiv preprint arXiv:1806.02978*, 2018.

[25] Shengyu Zhu and Zhitang Chen. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.

[26] Ignavier Ng, Zhuangyan Fang, Shengyu Zhu, and Zhitang Chen. Masked gradient-based causal structure learning. *arXiv preprint arXiv:1910.08527*, 2019.

[27] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

[28] Michel Besserve, Naji Shajarisales, Bernhard Schölkopf, and Dominik Janzing. Group invariance principles for causal generative models. *arXiv preprint arXiv:1705.02212*, 2017.

[29] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6979–6987, 2017.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[32] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.