

CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training

Murat Kocaoglu ^{*1,a}, Christopher Snyder ^{*1,b}, Alexandros G. Dimakis^{1,c} and Sriram Vishwanath^{1,d}

¹Department of Electrical and Computer Engineering, The University of Texas at Austin, USA
^a mkocaoglu@utexas.edu ^b 22csnyder@gmail.com ^c dimakis@austin.utexas.edu ^d sriram@austin.utexas.edu

September 18, 2017

Abstract

We propose an adversarial training procedure for learning a causal implicit generative model for a given causal graph. We show that adversarial training can be used to learn a generative model with true observational and interventional distributions if the generator architecture is consistent with the given causal graph. We consider the application of generating faces based on given binary labels where the dependency structure between the labels is preserved with a causal graph. This problem can be seen as learning a causal implicit generative model for the image and labels. We devise a two-stage procedure for this problem. First we train a causal implicit generative model over binary labels using a neural network consistent with a causal graph as the generator. We empirically show that Wasserstein GAN can be used to output discrete labels. Later we propose two new conditional GAN architectures, which we call CausalGAN and CausalBEGAN. We show that the optimal generator of the CausalGAN, given the labels, samples from the image distributions conditioned on these labels. The conditional GAN combined with a trained causal implicit generative model for the labels is then an implicit causal generative network over the labels and the generated image. We show that the proposed architectures can be used to sample from observational and interventional image distributions, even for interventions which do not naturally occur in the dataset.

1 Introduction

Generative adversarial networks are neural generative models that can be trained using backpropagation to mimic sampling from very high dimensional nonparametric distributions [13]. A *generator* network models the sampling process through feedforward computation. The generator output is constrained and refined through the feedback by a competitive "adversary network", that attempts to discriminate between the generated and real samples. In the application of sampling from a distribution over images, a generator, typically a neural network, outputs an image given independent noise variables. The objective of the generator is to maximize the loss of the discriminator (convince the discriminator that it outputs images from the real data distribution). GANs have shown tremendous success in generating samples from distributions such as image and video [37] and have even been proposed for language translation [38].

*Equal contribution.

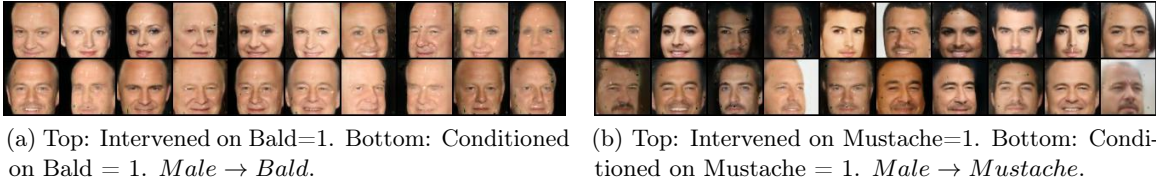


Figure 1: Observational and interventional image samples from CausalBEGAN. Our architecture can be used to sample not only from the joint distribution (conditioned on a label) but also from the interventional distribution, e.g., under the intervention $\text{do}(\text{Mustache} = 1)$. The resulting distributions are clearly different, as is evident from the samples outside the dataset, e.g., *females with mustaches*.

One extension idea for GANs is to enable sampling from the class conditional data distributions by feeding labels to the generator. Various neural network architectures have been proposed for solving this problem [27, 30, 1]. As far as we are aware of, in all of these works, the class labels are chosen independently from one another. Therefore, choosing one label does not affect the distribution of the other labels. As a result, these architectures do not provide the functionality to condition on a label, and sample other labels and the image. For concreteness consider a generator trained to output images of birds when given the *color* and *species* labels. On one hand, if we feed the generator *color=blue*, since *species* label is independent from the *color* label, we are likely to see blue eagles as well as blue jays. However, we do not expect to see any blue eagles when conditioned on *color=blue* in any dataset of real bird images. Similarly, consider a generator trained to output face images given the *gender* and *mustache* labels. When labels are chosen independently from one another, images generated under *mustache = 1* should contain both males and females, which is clearly different than conditioning on *mustache = 1*. The key for understanding and unifying these two notions, conditioning and being able to sample from distributions different than the dataset’s is to use *causality*.

We can think of generating an image conditioned on labels as a causal process: Labels determine the image distribution. The generator is a functional map from labels to image distributions. This is consistent with a simple causal graph "*Labels cause the Image*", represented with the graph $L \rightarrow G$, where L is the set of labels and G is the generated image. Using a finer model, we can also include the causal graph between the labels. Using the notion of causal graphs, we are interested in extending the previous work on conditional image generation by

- (i) capturing the dependence and
 - (ii) capturing the causal effect
- between labels and the image.

As an example, consider the causal graph between gender (G) and mustache (M) labels. The causal relation is clearly *gender causes mustache*¹, shown with the graph $G \rightarrow M$. Conditioning on *gender=male*, we expect to see males with or without mustaches, based on the fraction of males with mustaches in the population. When we condition on *mustache = 1*, we expect to sample from males only since the population does not contain females with mustaches. In addition to sampling from conditional distributions, causal models allow us to sample from various different distributions called *interventional distributions*, which we explain next.

From a causal lens, using independent labels corresponds to using an empty causal graph between

¹In reality, there may be confounder variables, i.e., variables that affect both, which are not observable. In this work, we ignore this effect by assuming the graph has causal sufficiency, i.e., there does not exist unobserved variables that cause more than one observable variable.

the labels. However in practice the labels are not independent and even have clear causal connections (e.g., *gender* causes *mustache*). Using an empty causal graph instead of the true causal graph, and setting a label to a particular value is equivalent to *intervening* on that label in the original causal graph, but also ignoring the way it affects other variables. An intervention is an experiment which fixes the value of a variable, without affecting the rest of the causal mechanism, which is different from conditioning. An intervention on a variable affects its descendant variables in the causal graph. But unlike conditioning, it does not affect the distribution of its ancestors. For example, instead of the causal graph *Gender causes Mustache*, if we used the empty causal graph between the same labels, intervening on *Gender = Female* would create females with mustaches, whereas with the correct causal graph, it should only yield females without mustaches since setting the *Gender* variable will affect all the variables that are downstream, e.g., *mustache*. See Figure 1 for a sample of our results which illustrate this concept on the *bald* and *mustache* variables. Similarly, for generating birds with the causal graph *Species causes color*, intervening on *color = blue* allows us to sample blue eagles (which do not exist) whereas conditioning on *color = blue* does not.

An *implicit* generative model [28] is a mechanism that can sample from a probability distribution but cannot provide likelihoods for data points. In this work we propose *causal implicit* generative models (CiGM): mechanisms that can sample not only from probability distributions but also from *conditional* and *interventional* distributions. We show that when the generator structure inherits its neural connections from the causal graph, GANs can be used to train causal implicit generative models. We use WassersteinGAN to train a causal implicit generative model for image labels, as part of a two-step procedure for training a causal implicit generative model for the images and image labels. For the second step, we propose a novel conditional GAN architecture and loss function called the CausalGAN. We show that the optimal generator can sample from the correct conditional and interventional distributions, which is summarized by the following theorem.

Theorem 1 (Informal). *Let $G(l, z)$ be the output of the generator for a given label l and latent vector z . Let G^* be the global optimal generator for the loss function in (5), when the rest of the network is trained to optimality. Then the generator samples from the conditional image distribution given the label, i.e., $p_g(G(l, Z) = x) = p_{data}(X = x|L = l)$, where p_{data} is the data probability density function over the image and the labels, p_g is the probability density function induced by the random variable Z , and X is the image random variable.*

The following corollary states that the trained causal implicit generative model for the labels concatenated with CausalGAN is a causal implicit generative model for the labels and image.

Corollary 1. *Suppose $C : Z_1 \rightarrow \mathcal{L}$ is a causal implicit generative model for the causal graph $D = (\mathcal{V}, E)$ where \mathcal{V} is the set of image labels and the observational joint distribution over these labels is strictly positive. Let $G : \mathcal{L} \times Z_2 \rightarrow \mathcal{I}$ be the class conditional generator that can sample from the image distribution conditioned on the given label combination $L \in \mathcal{L}$. Then $G(C(Z_1), Z_2)$ is a causal implicit generative model for the causal graph $D' = (\mathcal{V} \cup \{Image\}, E \cup \{(V_1, Image), (V_2, Image), \dots, (V_n, Image)\})$.*

In words, the corollary states the following: Consider a causal graph D' on the image labels and the *image* variable, where every *label* causes the *image*. Then combining an implicit causal generative model for the induced subgraph on the *labels* with a conditional generative model for the *image* given the *labels* yields a causal implicit generative model for D' .

Our contributions are as follows:

- We observe that adversarial training can be used after simply structuring the generator architecture based on the causal graph to train a causal implicit generative model.

- We empirically show how simple GAN training can be adapted using WassersteinGAN to learn a graph-structured generative model that outputs *essentially discrete*² labels.
- We consider the problem of conditional and interventional sampling of images given a causal graph over binary labels. We propose a two-stage procedure to train a causal implicit generative model over the binary labels and the image. As part of this procedure, we propose a novel conditional GAN architecture and loss function. We show that the global optimal generator³ provably samples from the class conditional distributions.
- We propose a natural but nontrivial extension of BEGAN to accept labels: using the same motivations for margins as in BEGAN [4], we arrive at a "margin of margins" term, which cannot be neglected. We show empirically that this model, which we call CausalBEGAN, produces high quality images that capture the image labels.
- We evaluate our causal implicit generative model training framework on the labeled CelebA data [23]. We show that the combined architecture generates images that can capture both the observational and interventional distributions over images and labels jointly⁴. We show the surprising result that CausalGAN and CausalBEGAN can produce high-quality label-consistent images *even for label combinations realized under interventions that never occur during training*, e.g., "woman with mustache".

2 Related Work

Using a generative adversarial network conditioned on the image labels has been proposed before: In [27], authors propose to extend generative adversarial networks to the setting where there is extra information, such as labels. The label of the image is fed to both the generator and the discriminator. This architecture is called conditional GAN. In [7], authors propose a new architecture called InfoGAN, which attempts to maximize a variational lower bound of mutual information between the labels given to the generator and the image. In [30], authors propose a new conditional GAN architecture, which performs well on higher resolution images. A class label is given to the generator. Image from the dataset is also chosen conditioned on this label. In addition to deciding if the image is real or fake, the discriminator has to also output an estimate of the class label.

Using causal principles for deep learning and using deep learning techniques for causal inference has been recently gaining attention. In [26], authors observe the connection between conditional GAN layers, and structural equation models. Based on this observation, they use CGAN [27] to learn the causal direction between two variables from a dataset. In [25], the authors propose using a neural network in order to discover the causal relation between image class labels based on static images. In [3], authors propose a new regularization for training a neural network, which they call causal regularization, in order to assure that the model is predictive in a causal sense. In a very recent work [5], authors point out the connection of GANs to causal generative models. However they see image as a cause of the neural net weights, and do not use labels.

BiGAN [9] and ALI [10] improve the standard GAN framework to provide the functionality of learning the mapping from image space to latent space. In CoGAN [22] the authors learn a joint distribution given samples from marginals by enforcing weight sharing between generators. This can, for example, be used to learn the joint distribution between image and labels. It is not clear, however, if this approach will work when the generator is structured via a causal graph. SD-GAN [8] is an architecture which splits the latent space into "Identity" and "Observation" portions. To

²Each of the generated labels are sharply concentrated around 0 and 1.

³Global optimal after the remaining network is trained to optimality.

⁴Our code is available at <https://github.com/mkocaoglu/CausalGAN>

generate faces of the same person, one can then fix the identity portion of the latent code. This works well for datasets where each identity has multiple observations. Authors in [1] use conditional GAN of [27] with a one-hot encoded vector that encodes the age interval. A generator conditioned on this one-hot vector can then be used for changing the age attribute of a face image. Another application of generative models is in compressed sensing: Authors in [6] give compressed sensing guarantees for recovering a vector, if the data lies close to the output of a trained generative model.

3 Background

3.1 Causality Basics

In this section, we give a brief introduction to causality. Specifically, we use Pearl’s framework [31], i.e., structural causal models, which uses structural equations and directed acyclic graphs between random variables to represent a causal model. We explain how causal principles apply to our framework through examples. For a more detailed treatment of the subject with more of the technical details, see [31].

Consider two random variables X, Y . Within the structural causal modeling framework and under the causal sufficiency assumption⁵, X *causes* Y simply means that there exists a function f and some unobserved random variable E , independent from X , such that $Y = f(X, E)$. Unobserved variables are also called *exogenous*. The causal graph that represents this relation is $X \rightarrow Y$. In general, a causal graph is a directed acyclic graph implied by the structural equations: The parents of a node in the causal graph represent the *causes* of that variable. The causal graph can be constructed from the structural equations as follows: The parents of a variable are those that appear in the structural equation that determines the value of that variable.

Formally, a structural causal model is a tuple $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P}_E(\cdot))$ that contains a set of functions $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$, a set of random variables $V = \{X_1, X_2, \dots, X_n\}$, a set of exogenous random variables $\mathcal{E} = \{E_1, E_2, \dots, E_n\}$, and a probability distribution over the exogenous variables \mathcal{P}_E ⁶. The set of observable variables \mathcal{V} has a joint distribution implied by the distributions of \mathcal{E} , and the functional relations \mathcal{F} . This distribution is the projection of \mathcal{P}_E onto the set of variables \mathcal{V} and is shown by \mathcal{P}_V . The causal graph D is then the directed acyclic graph on the nodes \mathcal{V} , such that a node X_j is a parent of node X_i if and only if X_j is in the domain of f_i , i.e., $X_i = f_i(X_j, S, E_i)$, for some $S \subset V$. The set of parents of variable X_i is shown by Pa_i . D is then a Bayesian network for the induced joint probability distribution over the observable variables \mathcal{V} . We assume causal sufficiency: Every exogenous variable is a direct parent of at most one observable variable.

An *intervention*, is an operation that changes the underlying causal mechanism, hence the corresponding causal graph. An intervention on X_i is denoted as $do(X_i = x_i)$. It is different from conditioning on $X_i = x$ in the following way: An intervention removes the connections of node X_i to its parents, whereas conditioning does not change the causal graph from which data is sampled. The interpretation is that, for example, if we set the value of X_i to 1, then it is no longer determined through the function $f_i(Pa_i, E_i)$. An intervention on a set of nodes is defined similarly. The joint distribution over the variables after an intervention (post-interventional distribution) can be calculated as follows: Since D is a Bayesian network for the joint distribution, the observational distribution can be factorized as $P(x_1, x_2, \dots, x_n) = \prod_{i \in [n]} \Pr(x_i | Pa_i)$, where the nodes in Pa_i are assigned to the corresponding values in $\{x_i\}_{i \in [n]}$. After an intervention on a set of nodes

⁵In a causally sufficient system, every unobserved variable affects no more than a single observed variable.

⁶The definition provided here assumes causal sufficiency, i.e., there are no exogenous variables that affect more than one observable variable. Under causal sufficiency, Pearl’s model assumes that the distribution over the exogenous variables is a product distribution, i.e., exogenous variables are mutually independent.

$X_S := \{X_i\}_{i \in S}$, i.e., $do(X_S = \mathbf{s})$, the post-interventional distribution is given by $\prod_{i \in [n] \setminus S} \Pr(x_i | Pa_i^S)$, where Pa_i^S is the shorthand notation for the following assignment: $X_j = x_j$ for $X_j \in Pa_i$ if $j \notin S$ and $X_j = \mathbf{s}(j)$ if $j \in S$ ⁷.

In general it is not possible to identify the true causal graph for a set of variables without performing experiments or making additional assumptions. This is because there are multiple causal graphs that lead to the same joint probability distribution even for two variables [36]. This paper does not address the problem of learning the causal graph: We assume the causal graph is given to us, and we learn a causal model, i.e., the functions and the distributions of the exogenous variables comprising the structural equations⁸. There is significant prior work on learning causal graphs that could be used before our method, see e.g. [11, 17, 18, 16, 35, 24, 12, 32, 21, 20, 19]. When the true causal graph is unknown we can use any feasible graph, i.e., any Bayesian network that respects the conditional independencies present in the data. If only a few conditional independencies are known, a richer model (i.e., a denser Bayesian network) can be used, although a larger number of functional relations should be learned in that case. We explore the effect of the used Bayesian network in Section 8. If the used Bayesian network has edges that are inconsistent with the true causal graph, our conditional distributions will be correct, but the interventional distributions will be different.

4 Causal Implicit Generative Models

Implicit generative models [28] are used to sample from a probability distribution without an explicit parameterization. Generative adversarial networks are arguably one of the most successful examples of implicit generative models. Thanks to an adversarial training procedure, GANs are able to produce realistic samples from distributions over a very high dimensional space, such as images. To sample from the desired distribution, one samples a vector from a known distribution, such as Gaussian or uniform, and feeds it into a feedforward neural network which was trained on a given dataset. Although implicit generative models can sample from the data distribution, they do not provide the functionality to sample from interventional distributions. *Causal implicit generative models* provide a way to sample from both observational and interventional distributions.

We show that generative adversarial networks can also be used for training causal implicit generative models. Consider the simple causal graph $X \rightarrow Z \leftarrow Y$. Under the causal sufficiency assumption, this model can be written as $X = f_X(N_X), Y = f_Y(N_Y), Z = f_Z(X, Y, N_Z)$, where f_X, f_Y, f_Z are some functions and N_X, N_Y, N_Z are jointly independent variables. The following simple observation is useful: *In the GAN training framework, generator neural network connections can be arranged to reflect the causal graph structure.* Consider Figure 2b. The feedforward neural networks can be used to represent the functions f_X, f_Y, f_Z . The noise terms can be chosen as independent, complying with the condition that (N_X, N_Y, N_Z) are jointly independent. Hence this feedforward neural network can be used to represent the causal graph $X \rightarrow Z \leftarrow Y$ if f_X, f_Y, f_Z are within the class of functions that can be represented with the given family of neural networks.

The following proposition is well known in the causality literature. It shows that given the true causal graph, two causal models that have the same observational distribution have the same interventional distribution for any intervention.

Proposition 1. *Let $\mathcal{M}_1 = (D_1 = (V, E), N_1, \mathcal{F}_1, \mathcal{P}_{N_1}(\cdot)), \mathcal{M}_2 = (D_2 = (V, E), N_2, \mathcal{F}_2, \mathcal{Q}_{N_2}(\cdot))$ be two causal models. If $\mathcal{P}_V(\cdot) = \mathcal{Q}_V(\cdot)$, then $\mathcal{P}_V(\cdot | do(S)) = \mathcal{Q}_V(\cdot | do(S))$*

⁷With slight abuse of notation, we use $\mathbf{s}(j)$ to represent the value assigned to variable X_j by the intervention rather than the j th coordinate of \mathbf{s}

⁸Even when the causal graph is given, there will be many different sets of functions and exogenous noise distributions that explain the observed joint distribution for that causal graph. We are learning one such model.

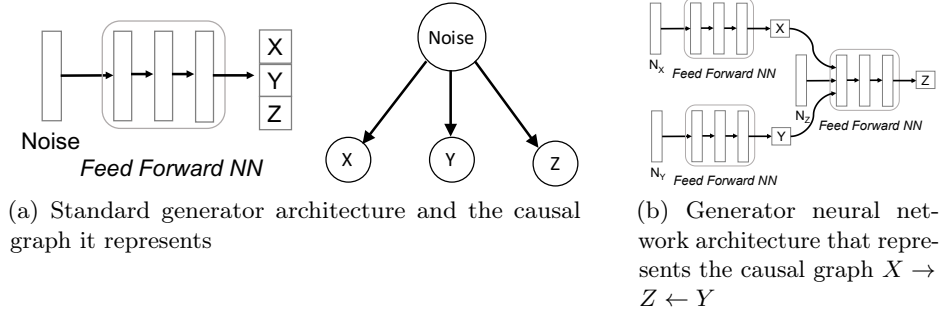


Figure 2: (a) The causal graph implied by the standard generator architecture, feedforward neural network. (b) A neural network implementation of the causal graph $X \rightarrow Z \leftarrow Y$: Each feed forward neural net captures the function f in the structural equation model $V = f(Pa_V, E)$.

Proof. Note that D_1 and D_2 are the same causal Bayesian networks [31]. Interventional distributions for causal Bayesian networks can be directly calculated from the conditional probabilities and the causal graph. Thus, \mathcal{M}_1 and \mathcal{M}_2 have the same interventional distributions. \square

We have the following definition, which ties a feedforward neural network with a causal graph:

Definition 1. Let $Z = \{Z_1, Z_2, \dots, Z_m\}$ be a set of mutually independent random variables. A feedforward neural network G that outputs the vector $G(Z) = [G_1(Z), G_2(Z), \dots, G_n(Z)]$ is called **consistent** with a causal graph $D = ([n], E)$, if $\forall i \in [n]$, \exists a set of layers f_i such that $G_i(Z)$ can be written as $G_i(Z) = f_i(\{G_j(Z)\}_{j \in Pa_i}, Z_{S_i})$, where Pa_i are the set of parents of i in D , and $Z_{S_i} := \{Z_j : j \in S_i\}$ are collections of subsets of Z such that $\{S_i : i \in [n]\}$ is a partition of $[m]$.

Based on the definition, we say a feedforward neural network G with output

$$G(Z) = [G_1(Z), G_2(Z), \dots, G_n(Z)], \quad (1)$$

is a causal implicit generative model for the causal model $\mathcal{M} = (D = ([n], E), N, \mathcal{F}, \mathcal{P}_N(\cdot))$ if G is consistent with the causal graph D and $\Pr(G(Z) = \mathbf{x}) = \mathcal{P}_V(\mathbf{x}), \forall \mathbf{x}$.

We propose using adversarial training where the generator neural network is consistent with the causal graph according to Definition 1. This notion is illustrated in Figure 2b.

5 Causal Generative Adversarial Networks

Causal implicit generative models can be trained given a causal graph and samples from a joint distribution. However, for the application of image generation with binary labels, we found it difficult to simultaneously learn the joint label and image distribution⁹. For these applications, we focus on dividing the task of learning a causal implicit generative causal model into two subtasks: First, learn the causal implicit generative model over a small set of variables. Then, learn the remaining set of variables conditioned on the first set of variables using a conditional generative network. For this training to be consistent with the causal structure, every node in the first set should come before any node in the second set with respect to the partial order of the causal graph. We assume that the problem of generating images based on the image labels inherently contains a causal graph similar to the one given in Figure 3, which makes it suitable for a two-stage training: First, train a generative

⁹Please see the Appendix for our primitive result using this naive attempt.

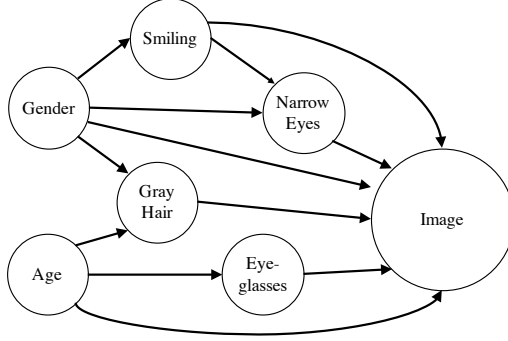


Figure 3: A plausible causal model for image generation.

model over the labels, then train a generative model for the images conditioned on the labels. As we show next, our new architecture and loss function (CausalGAN) assures that the optimum generator outputs the label conditioned image distributions. Under the assumption that the joint probability distribution over the labels is strictly positive¹⁰, combining pretrained causal generative model for labels with a label-conditioned image generator gives a causal implicit generative model for images. The formal statement for this corollary is postponed to Section 6.

5.1 Causal Implicit Generative Model for Binary Labels

Here we describe the adversarial training of a causal implicit generative model for binary labels. This generative model, which we call the *Causal Controller*, will be used for controlling which distribution the images will be sampled from when intervened or conditioned on a set of labels. As in Section 4, we structure the Causal Controller network to sequentially produce labels according to the causal graph.

Since our theoretical results hold for binary labels, we prefer a generator which can sample from an essentially discrete label distribution¹¹. However, the standard GAN training is not suited for learning a discrete distribution due to the properties of Jensen-Shannon divergence. To be able to sample from a discrete distribution, we employ WassersteinGAN [2]. We used the model of [15], where the Lipschitz constraint on the gradient is replaced by a penalty term in the loss.

5.2 CausalGAN Architecture

As part of the two-step process proposed in Section 4 of learning a causal implicit generative model over the labels *and* the image variables, we design a new conditional GAN architecture to generate the images based on the labels of the Causal Controller. Unlike previous work, our new architecture and loss function assures that the optimum generator outputs the label conditioned image distributions. We use a pretrained Causal Controller which is not further updated.

Labeler and Anti-Labeler: We have two separate labeler neural networks. *The Labeler* is trained to estimate the labels of images in the dataset. *The Anti-Labeler* is trained to estimate the labels of the images which are sampled from the generator. The label of a generated image is the label produced by the Causal Controller.

¹⁰This assumption does not hold in the CelebA dataset: $\Pr(Male = 0, Mustache = 1) = 0$. However, we will see that the trained model is able to extrapolate to these interventional distributions when the CausalGAN model is not trained for very long.

¹¹Ignoring the theoretical considerations, adding noise to transform the labels artificially into continuous targets also works. However we observed better empirical convergence with this technique.

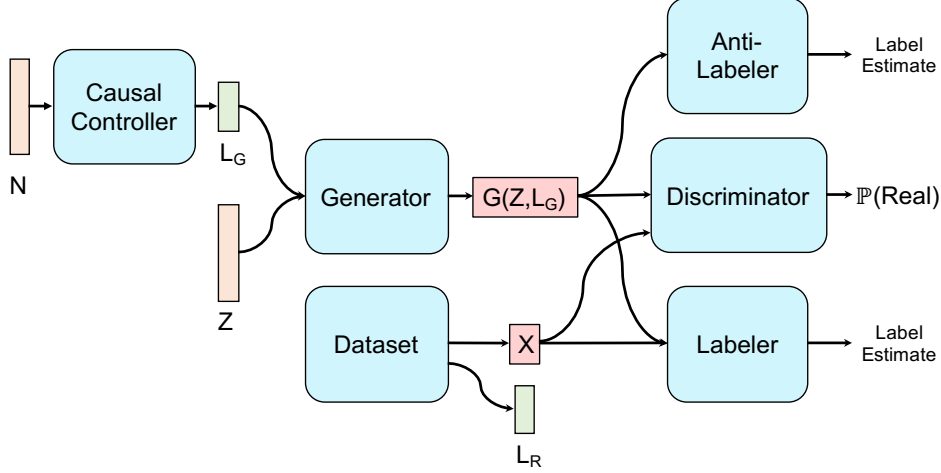


Figure 4: CausalGAN architecture.

Generator: The objective of the generator is 3-fold: producing realistic images by competing with the discriminator, capturing the labels it is given in the produced images by minimizing the Labeler loss, and avoiding drifting towards unrealistic image distributions that are easy to label by maximizing the Anti-Labeler loss. For the optimum Causal Controller, Labeler, and Anti-Labeler, we will later show that the optimum generator samples from the same distribution as the class conditional images.

The most important distinction of CausalGAN with the existing conditional GAN architectures is that it uses an Anti-Labeler network in addition to a Labeler network. Notice that the theoretical guarantee we develop in Section 6 does not hold when Anti-Labeler network is not used. Intuitively, the Anti-Labeler loss discourages the generator network to generate only few typical faces for a fixed label combination. This is a phenomenon that we call *label-conditioned mode collapse*. In the literature, minibatch-features are one of the most popular techniques used to avoid mode-collapse [34]. However, the diversity within a batch of images due to different label combinations can make this approach ineffective for combatting label-conditioned mode collapse. We observe that this intuition carries over to practice.

Loss Functions

We present the results for a single binary label l . For the more general case of d binary labels, we have an extension where the labeler and the generator losses are slightly modified. We explain this extension in the supplementary material in Section 10.4 along with the proof that the optimal generator samples from the class conditional distribution given the d -dimensional label vector. Let $\mathbb{P}(l = 1) = \rho$. We use $p_g^0(x) := \mathbb{P}(G(z, l) = x | l = 0)$ and $p_g^1(x) := \mathbb{P}(G(z, l) = x | l = 1)$. $G(\cdot)$, $D(\cdot)$, $D_{LR}(\cdot)$, and $D_{LG}(\cdot)$ are the mappings due to generator, discriminator, Labeler, and Anti-Labeler respectively.

The generator loss function of CausalGAN contains label loss terms, the GAN loss in [13], and an added loss term due to the discriminator. With the addition of this term to the generator loss, we will be able to prove that the optimal generator outputs the class conditional image distribution. This result will also be true for multiple binary labels.

For a fixed generator, Anti-Labeler solves the following optimization problem:

$$\max_{D_{LG}} \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(D_{LG}(x))] + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(1 - D_{LG}(x))]. \quad (2)$$

The Labeler solves the following optimization problem:

$$\max_{D_{LR}} \rho \mathbb{E}_{x \sim p_{\text{data}}^1(x)} [\log(D_{LR}(x))] + (1 - \rho) \mathbb{E}_{x \sim p_{\text{data}}^0(x)} [\log(1 - D_{LR}(x))]. \quad (3)$$

For a fixed generator, the discriminator solves the following optimization problem:

$$\max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{1 - D(x)}{D(x)} \right) \right]. \quad (4)$$

For a fixed discriminator, Labeler and Anti-Labeler, generator solves the following optimization problem:

$$\begin{aligned} \min_G & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{1 - D(x)}{D(x)} \right) \right] \\ & - \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(D_{LR}(X))] - (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(1 - D_{LR}(X))] \\ & + \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(D_{LG}(X))] + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(1 - D_{LG}(X))]. \end{aligned} \quad (5)$$

Remark: Although the authors in [13] have the additive term $\mathbb{E}_{x \sim p_g(x)} [\log(1 - D(X))]$ in the definition of the loss function, in practice they use the term $\mathbb{E}_{x \sim p_g(x)} [-\log(D(X))]$. It is interesting to note that this is the extra loss terms we need for the global optimum to correspond to the class conditional image distributions under a label loss.

5.3 CausalBEGAN Architecture

In this section, we propose a simple, but non-trivial extension of BEGAN where we feed image labels to the generator. One of the central contributions of BEGAN [4] is a control theory-inspired boundary equilibrium approach that encourages generator training only when the discriminator is near optimum and its gradients are the most informative. The following observation helps us carry the same idea to the case with labels: Label gradients are most informative when the image quality is high. Here, we introduce a new loss and a set of margins that reflect this intuition.

Formally, let $\mathcal{L}(x)$ be the average L_1 pixel-wise autoencoder loss for an image x , as in BEGAN. Let $\mathcal{L}_{sq}(u, v)$ be the squared loss term, i.e., $\|u - v\|_2^2$. Let (x, l_x) be a sample from the data distribution, where x is the image and l_x is its corresponding label. Similarly, $G(z, l_g)$ is an image sample from the generator, where l_g is the label used to generate this image. Denoting the space of images by \mathcal{I} , let $G : \mathbb{R}^n \times \{0, 1\}^m \mapsto \mathcal{I}$ be the generator. As a naive attempt to extend the original BEGAN loss formulation to include the labels, we can write the following loss functions:

$$\begin{aligned} Loss_D &= \mathcal{L}(x) - \mathcal{L}(\text{Labeler}(G(z, l))) + \mathcal{L}_{sq}(l_x, \text{Labeler}(x)) - \mathcal{L}_{sq}(l_g, \text{Labeler}(G(z, l_g))), \\ Loss_G &= \mathcal{L}(G(z, l_g)) + \mathcal{L}_{sq}(l_g, \text{Labeler}(G(z, l_g))). \end{aligned} \quad (6)$$

However, this naive formulation does not address the use of margins, which is extremely critical in the BEGAN formulation. Just as a better trained BEGAN discriminator creates more useful gradients for image generation, a better trained Labeler is a prerequisite for meaningful gradients. This motivates an additional margin-coefficient tuple (b_2, c_2) , as shown in (7,8).

The generator tries to jointly minimize the two loss terms in the formulation in (6). We empirically observe that occasionally the image quality will suffer because the images that best exploit the Labeler network are often not obliged to be realistic, and can be noisy or misshapen. Based on this, label loss seems unlikely to provide useful gradients unless the image quality remains good. Therefore we encourage the generator to incorporate label loss only when the *image quality margin*

b_1 is large compared to the *label margin* b_2 . To achieve this, we introduce a new *margin of margins* term, b_3 . As a result, the margin equations and update rules are summarized as follows, where $\lambda_1, \lambda_2, \lambda_3$ are learning rates for the coefficients.

$$\begin{aligned} b_1 &= \gamma_1 * \mathcal{L}(x) - \mathcal{L}(G(z, l_g)). \\ b_2 &= \gamma_2 * \mathcal{L}_{sq}(l_x, \text{Labeler}(x)) - \mathcal{L}_{sq}(l_g, \text{Labeler}(G(z, l_g))). \end{aligned} \quad (7)$$

$$\begin{aligned} b_3 &= \gamma_3 * \text{relu}(b_1) - \text{relu}(b_2). \\ c_1 &\leftarrow \text{clip}_{[0,1]}(c_1 + \lambda_1 * b_1). \\ c_2 &\leftarrow \text{clip}_{[0,1]}(c_2 + \lambda_2 * b_2). \end{aligned} \quad (8)$$

$$\begin{aligned} c_3 &\leftarrow \text{clip}_{[0,1]}(c_3 + \lambda_3 * b_3). \\ \text{Loss}_D &= \mathcal{L}(x) - c_1 * \mathcal{L}(G(z, l_g)) + \mathcal{L}_{sq}(l_x, \text{Labeler}(x)) - c_2 * \mathcal{L}_{sq}(l_g, G(z, l_g)). \\ \text{Loss}_G &= \mathcal{L}(G(z, l_g)) + c_3 * \mathcal{L}_{sq}(l_g, \text{Labeler}(G(z, l_g))). \end{aligned} \quad (9)$$

One of the advantages of BEGAN is the existence of a monotonically decreasing scalar which can track the convergence of the gradient descent optimization. Our extension preserves this property as we can define

$$\mathcal{M}_{\text{complete}} = \mathcal{L}(x) + |b_1| + |b_2| + |b_3|, \quad (10)$$

and show that $\mathcal{M}_{\text{complete}}$ decreases progressively during our optimizations. See Figure 21 in the Appendix.

6 Theoretical Guarantees for CausalGAN

In this section, we show that the best CausalGAN generator for the given loss function outputs the class conditional image distribution when Causal Controller outputs the real label distribution and labelers operate at their optimum. We show this result for the case of a single binary label $l \in \{0, 1\}$. The proof can be extended to multiple binary variables, which we explain in the supplementary material in Section 10.4. As far as we are aware of, this is the first conditional generative adversarial network architecture with this guarantee.

6.1 CausalGAN with Single Binary Label

First, we find the optimal discriminator for a fixed generator. Note that in (4), the terms that the discriminator can optimize are the same as the GAN loss in [13]. Hence the optimal discriminator behaves the same as in the standard GAN. Then, the following lemma from [13] directly applies to our discriminator:

Proposition 2 ([13]). *For fixed G , the optimal discriminator D is given by*

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}. \quad (11)$$

Second, we identify the optimal Labeler and Anti-Labeler. We have the following lemma:

Lemma 1. *The optimum Labeler has $D_{LR}(x) = \mathbb{P}_r(l = 1|x)$.*

Proof. Please see the supplementary material. □

Similarly, we have the corresponding lemma for Anti-Labeler:

Lemma 2. *For a fixed generator with $x \sim p_g$, the optimum Anti-Labeler has $D_{LG}(x) = \mathbb{P}_g(l = 1|x)$.*

Proof. Proof is the same as the proof of Lemma 1. \square

Define $C(G)$ as the generator loss for when discriminator, Labeler and Anti-Labeler are at their optimum. Then, we show that the generator that minimizes $C(G)$ outputs class conditional image distributions.

Theorem 2 (Theorem 1 formal for single binary label). *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g^0 = p_{data}^0$ and $p_g^1 = p_{data}^1$, i.e., if and only if given a label l , generator output $G(z, l)$ has the class conditional image distribution $p_{data}(x|l)$.*

Proof. Please see the supplementary material. \square

Now we can show that our two stage procedure can be used to train a causal implicit generative model for any causal graph where the *Image* variable is a sink node, captured by the following corollary:

Corollary 2. *Suppose $C : \mathcal{Z}_1 \rightarrow \mathcal{L}$ is a causal implicit generative model for the causal graph $D = (\mathcal{V}, E)$ where \mathcal{V} is the set of image labels and the observational joint distribution over these labels are strictly positive. Let $G : \mathcal{L} \times \mathcal{Z}_2 \rightarrow \mathcal{I}$ be the class conditional GAN that can sample from the image distribution conditioned on the given label combination $L \in \mathcal{L}$. Then $G(C(\mathcal{Z}_1), \mathcal{Z}_2)$ is a causal implicit generative model for the causal graph $D' = (\mathcal{V} \cup \{\text{Image}\}, E \cup \{(V_1, \text{Image}), (V_2, \text{Image}), \dots (V_n, \text{Image})\})$.*

Proof. Please see the supplementary material. \square

6.2 Extensions to Multiple Labels

In Theorem 2 we show that the optimum generator samples from the class conditional distributions given a single binary label. Our objective is to extend this result to the case with d binary labels.

First we show that if the Labeler and Anti-Labeler are trained to output 2^d scalars, each interpreted as the posterior probability of a particular label combination given the image, then the minimizer of $C(G)$ samples from the class conditional distributions *given d labels*. This result is shown in Theorem 3 in the supplementary material. However, when d is large, this architecture may be hard to implement. To resolve this, we propose an alternative architecture, which we implement for our experiments: We extend the single binary label setup and use cross entropy loss terms for each label. This requires Labeler and Anti-Labeler to have only d outputs. However, although we need the generator to capture the joint label posterior given the image, this only assures that the generator captures each label’s posterior distribution, i.e., $p_r(l_i|x) = p_g(l_i|x)$ (Proposition 3). This, in general, does not guarantee that the class conditional distributions will be true to the data distribution. However, for many joint distributions of practical interest, where *the set of labels are completely determined by the image*¹², we show that this guarantee implies that the joint label posterior will be true to the data distribution, implying that the optimum generator samples from the class conditional distributions. Please see Section 10.5 for the formal results and more details.

7 Implementation

In this section, we explain the differences between implementation and theory, along with other implementation details for both CausalGAN and CausalBEGAN.

¹²The dataset we are using arguably satisfies this condition.

7.1 Pretraining Causal Controller for Face Labels

In this section, we explain the implementation details of the Wasserstein Causal Controller for generating face labels. We used the total variation distance (TVD) between the distribution of generator and data distribution as a metric to decide the success of the models.

The gradient term used as a penalty is estimated by evaluating the gradient at points interpolated between the real and fake batches. Interestingly, this Wasserstein approach gives us the opportunity to train the Causal Controller to output (almost) discrete labels (See Figure 7a). In practice though, we still found benefit in rounding them before passing them to the generator.

The generator architecture is structured in accordance with Section 4 based on the causal graph in Figure 5, using uniform noise as exogenous variables and 6 layer neural networks as functions mapping parents to children. For the training, we used 25 Wasserstein discriminator (critic) updates per generator update, with a learning rate of 0.0008.

7.2 Implementation Details for CausalGAN

In practice, we use stochastic gradient descent to train our model. We use *DCGAN* [33], a convolutional neural net-based implementation of generative adversarial networks, and extend it into our Causal GAN framework. We have expanded it by adding our Labeler networks, training a Causal Controller network and modifying the loss functions appropriately. Compared to DCGAN an important distinction is that we make 6 generator updates for each discriminator update on average. The discriminator and labeler networks are concurrently updated in a single iteration.

Notice that the loss terms defined in Section 5.2 contain a single binary label. In practice we feed a d -dimensional label vector and need a corresponding loss function. We extend the Labeler and Anti-Labeler loss terms by simply averaging the loss terms for every label. The i^{th} coordinates of the d -dimensional vectors given by the labelers determine the loss terms for label i . Note that this is different than the architecture given in Section 10.4, where the discriminator outputs a length- 2^d vector and estimates the probabilities of all label combinations given the image. Therefore this approach does not have the guarantee to sample from the class conditional distributions, if the data distribution is not restricted. However, for the type of labeled image dataset we use in this work, where labels seem to be completely determined given an image, this architecture is sufficient to have the same guarantees. For the details, please see Section 10.5 in the supplementary material.

Compared to the theory we have, another difference in the implementation is that we have swapped the order of the terms in the cross entropy expressions for labeler losses. This has provided sharper images at the end of the training.

7.3 Usage of Anti-Labeler in CausalGAN

An important challenge that comes with gradient-based training is the use of Anti-Labeler. We observe the following: In the early stages of the training, Anti-Labeler can very quickly minimize its loss, if the generator falls into label-conditioned mode collapse. Recall that we define label-conditioned mode-collapse as the problem of generating few typical faces when a label is fixed. For example, the generator can output the same face when *Eyeglasses* variable is set to 1. This helps generator to easily satisfy the label loss term we add to our loss function. Notice that however, if label-conditioned mode collapse occurs, Anti-Labeler will very easily estimate the true labels given an image, since it is always provided with the same image. Hence, maximizing the Anti-Labeler loss in the early stages of the training helps generator to avoid label-conditioned mode collapse with our loss function.

In the later stages of the training, due to the other loss terms, generator outputs realistic images, which drives Anti-Labeler to act similar to Labeler. Thus, maximizing Anti-Labeler loss and minimizing Labeler loss become contradicting tasks. This moves the training in a direction where labels are captured less and less by the generator, hence losing the conditional image generation property.

Based on these observations, we employ the following loss function for the generator in practice:

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \mathcal{L}_{LabelerR} - e^{-t/T} \mathcal{L}_{LabelerG}, \quad (12)$$

where the terms are GAN loss term, loss of Labeler and loss of Anti-Labeler respectively (see first second and third lines of (5)). t is the number of iterations in the training and T is the time constant of the exponential decaying coefficient for the Anti-Labeler loss. $T = 3000$ is chosen for the experiments, which corresponds to roughly 1 epoch of training.

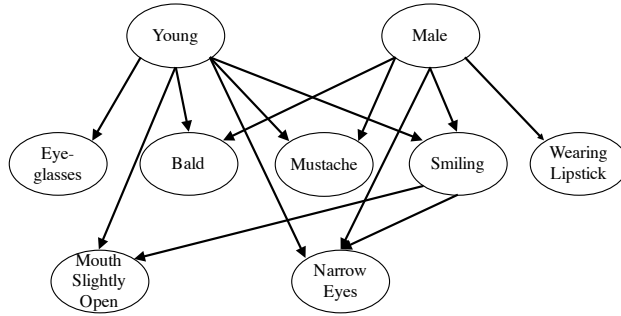


Figure 5: The causal graph used for simulations for both CausalGAN and CausalBEGAN, called Causal Graph 1 (G1). We also add edges (see Appendix Section 10.6) to form the complete graph "cG1". We also make use of the graph rcG1, which is obtained by reversing the direction of every edge in cG1.

7.4 Conditional Image Generation for CausalBEGAN

The labels input to CausalBEGAN are taken from the Causal Controller. We use very few parameter tunings. We use the same learning rate (0.00008) for both the generator and discriminator and do 1 update of each simultaneously (calculating the for each before applying either). We simply use $\gamma_1 = \gamma_2 = \gamma_3 = 0.5$. We do not expect the model to be very sensitive to these parameter values, as we achieve good performance without hyperparameter tweaking. We do use customized margin learning rates $\lambda_1 = 0.001$, $\lambda_2 = 0.00008$, $\lambda_3 = 0.01$, which reflect the asymmetry in how quickly the generator can respond to each margin. For example c_2 can have much more "spiky", fast responding behavior compared to others even when paired with a smaller learning rate, although we have not explored this parameter space in depth. In these margin behaviors, we observe that the best performing models have all three margins "active": near 0 while frequently taking small positive values.

8 Results

8.1 Dependence of GAN Behavior on Causal Graph

In Section 4 we showed how a GAN could be used to train a causal implicit generative model by incorporating the causal graph into the generator structure. Here we investigate the behavior and

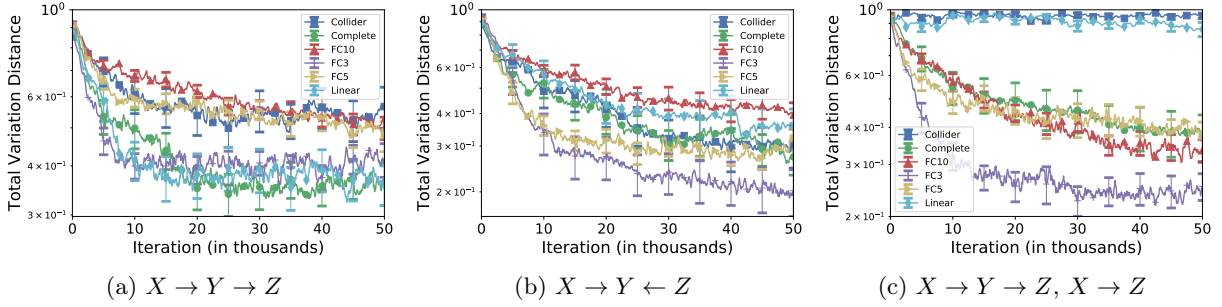


Figure 6: Convergence in total variation distance of generated distribution to the true distribution for causal implicit generative model, when the generator is structured based on different causal graphs. (a) Data generated from line graph $X \rightarrow Y \rightarrow Z$. The best convergence behavior is observed when the true causal graph is used in the generator architecture. (b) Data generated from collider graph $X \rightarrow Y \leftarrow Z$. Fully connected layers may perform better than the true graph depending on the number of layers. Collider and complete graphs performs better than the line graph which implies the wrong Bayesian network. (c) Data generated from complete graph $X \rightarrow Y \rightarrow Z, X \rightarrow Z$. Fully connected with 3 layers performs the best, followed by the complete and fully connected with 5 and 10 layers. Line and collider graphs, which implies the wrong Bayesian network does not show convergence behavior.

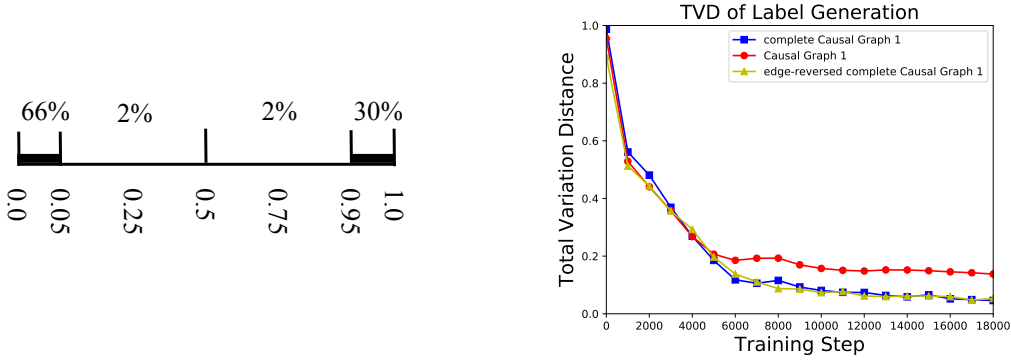
convergence of causal implicit generative models when the true data distribution arises from another (possibly distinct) causal graph.

We consider causal implicit generative model convergence on synthetic data whose three features $\{X, Y, Z\}$ arise from one of three causal graphs: "line" $X \rightarrow Y \rightarrow Z$, "collider" $X \rightarrow Y \leftarrow Z$, and "complete" $X \rightarrow Y \rightarrow Z, X \rightarrow Z$. For each node a (randomly sampled once) cubic polynomial in $n + 1$ variables computes the value of that node given its n parents and 1 uniform exogenous variable. We then repeat, creating a new synthetic dataset in this way for each causal model and report the averaged results of 20 runs for each model.

For each of these data generating graphs, we compare the convergence of the joint distribution to the true joint in terms of the total variation distance, when the generator is structured according to a line, collider, or complete graph. For completeness, we also include generators with no knowledge of causal structure: $\{fc3, fc5, fc10\}$ are fully connected neural networks that map uniform random noise to 3 output variables using either 3, 5, or 10 layers respectively.

The results are given in Figure 6. Data is generated from line causal graph $X \rightarrow Y \rightarrow Z$ (left panel), collider causal graph $X \rightarrow Y \leftarrow Z$ (middle panel), and complete causal graph $X \rightarrow Y \rightarrow Z, X \rightarrow Z$ (right panel). Each curve shows the convergence behavior of the generator distribution, when generator is structured based on each one of these causal graphs. We expect convergence when the causal graph used to structure the generator is capable of generating the joint distribution due to the true causal graph: as long as we use the correct Bayesian network, we should be able to fit to the true joint. For example, complete graph can encode all joint distributions. Hence, we expect complete graph to work well with all data generation models. Standard fully connected layers correspond to the causal graph with a latent variable causing all the observable variables. Ideally, this model should be able to fit to any causal generative model. However, the convergence behavior of adversarial training across these models is unclear, which is what we are exploring with Figure 6.

For the line graph data $X \rightarrow Y \rightarrow Z$, we see that the best convergence behavior is when line graph is used in the generator architecture. As expected, complete graph also converges well, with



(a) Essentially Discrete Range of Causal Controller

(b) TVD vs. No. of Iters in CelebA Labels

Figure 7: (a) A number line of unit length binned into 4 unequal bins along with the percent of Causal Controller ($G1$) samples in each bin. Results are obtained by sampling the joint label distribution 1000 times and forming a histogram of the scalar outputs corresponding to any label. Note that our Causal Controller output labels are approximately discrete even though the input is a continuum (uniform). The 4% between 0.05 and 0.95 is not at all uniform and almost zero near 0.5. (b) Progression of total variation distance between the Causal Controller output with respect to the number of iterations: Causal Graph 1 is used in the training with Wasserstein loss.

slight delay. Similarly, fully connected network with 3 layers show good performance, although surprisingly fully connected with 5 and 10 layers perform much worse. It seems that although fully connected can encode the joint distribution in theory, in practice with adversarial training, the number of layers should be tuned to achieve the same performance as using the true causal graph. Using the wrong Bayesian network, the collider, also yields worse performance.

For the collider graph, surprisingly using a fully connected generator with 3 and 5 layers shows the best performance. However, consistent with the previous observation, the number of layers is important, and using 10 layers gives the worst convergence behavior. Using complete and collider graphs achieves the same decent performance, whereas line graph, a wrong Bayesian network, performs worse than the two.

For the complete graph, fully connected 3 performs the best, followed by fully connected 5, 10 and the complete graph. As we expect, line and collider graphs, which cannot encode all the distributions due to a complete graph, performs the worst and does not actually show any convergence behavior.

8.2 Wasserstein Causal Controller on CelebA Labels

We test the performance of our Wasserstein Causal Controller on a subset of the binary labels of CelebA dataset. We use the causal graph given in Figure 5.

For causal graph training, first we verify that our Wasserstein training allows the generator to learn a mapping from continuous uniform noise to a discrete distribution. Figure 7a shows where the samples, averaged over all the labels in Causal Graph 1, from this generator appears on the real line. The result emphasizes that the proposed Causal Controller outputs an almost discrete distribution: 96% of the samples appear in 0.05–neighborhood of 0 or 1. Outputs shown are *unrounded* generator outputs.

A stronger measure of convergence is the total variational distance (TVD). For Causal Graph 1 ($G1$), our defined completion ($cG1$), and $cG1$ with arrows reversed ($rcG1$), we show convergence of TVD with training (Figure 7b). Both $cG1$ and $rcG1$ have TVD decreasing to 0, and TVD for $G1$

assymptotes to around 0.14 which corresponds to the incorrect conditional independence assumptions that G1 makes. This suggests that any given complete causal graph will lead to a nearly perfect implicit causal generator over labels and that bayesian partially incorrect causal graphs can still give reasonable convergence.

8.3 CausalGAN Results

In this section, we train the whole CausalGAN together using a pretrained Causal Controller network. The results are given in Figures 8a-12a. The difference between intervening and conditioning is clear through certain features. We implement conditioning through rejection sampling. See [29, 14] for other works on conditioning for implicit generative models.



(a) Intervening vs Conditioning on Mustache, Top: Intervene Mustache=1, Bottom: Condition Mustache=1

Figure 8: Intervening/Conditioning on Mustache label in Causal Graph 1. Since $Male \rightarrow Mustache$ in Causal Graph 1, we do not expect $do(Mustache = 1)$ to affect the probability of $Male = 1$, i.e., $\mathbb{P}(Male = 1|do(Mustache = 1)) = \mathbb{P}(Male = 1) = 0.42$. Accordingly, the top row shows both males and females with mustaches, even though the generator never sees the label combination $\{Male = 0, Mustache = 1\}$ during training. The bottom row of images sampled from the conditional distribution $\mathbb{P}(.|Mustache = 1)$ shows only male images because in the dataset $\mathbb{P}(Male = 1|Mustache = 1) \approx 1$.



(a) Intervening vs Conditioning on Bald, Top: Intervene Bald=1, Bottom: Condition Bald=1

Figure 9: Intervening/Conditioning on Bald label in Causal Graph 1. Since $Male \rightarrow Bald$ in Causal Graph 1, we do not expect $do(Bald = 1)$ to affect the probability of $Male = 1$, i.e., $\mathbb{P}(Male = 1|do(Bald = 1)) = \mathbb{P}(Male = 1) = 0.42$. Accordingly, the top row shows both bald males and bald females. The bottom row of images sampled from the conditional distribution $\mathbb{P}(.|Bald = 1)$ shows only male images because in the dataset $\mathbb{P}(Male = 1|Bald = 1) \approx 1$.

8.4 CausalBEGAN Results

In this section, we train CausalBEGAN on CelebA dataset using Causal Graph 1. The Causal Controller is pretrained with a Wasserstein loss and used for training the CausalBEGAN.

To first empirically justify the need for the margin of margins we introduced in (9) (c_3 and b_3), we train the same CausalBEGAN model setting $c_3 = 1$, removing the effect of this margin. We show



(a) Intervening vs Conditioning on Wearing Lipstick, Top: Intervene Wearing Lipstick=1, Bottom: Condition Wearing Lipstick=1

Figure 10: Intervening/Conditioning on Wearing Lipstick label in Causal Graph 1. Since $Male \rightarrow WearingLipstick$ in Causal Graph 1, we do not expect $do(Wearing Lipstick = 1)$ to affect the probability of $Male = 1$, i.e., $\mathbb{P}(Male = 1|do(Wearing Lipstick = 1)) = \mathbb{P}(Male = 1) = 0.42$. Accordingly, the top row shows both males and females who are wearing lipstick. However, the bottom row of images sampled from the conditional distribution $\mathbb{P}(.|Wearing Lipstick = 1)$ shows only female images because in the dataset $\mathbb{P}(Male = 0|Wearing Lipstick = 1) \approx 1$.



(a) Intervening vs Conditioning on Mouth Slightly Open, Top: Intervene Mouth Slightly Open=1, Bottom: Condition Mouth Slightly Open=1

Figure 11: Intervening/Conditioning on Mouth Slightly Open label in Causal Graph 1. Since $Smiling \rightarrow MouthSlightlyOpen$ in Causal Graph 1, we do not expect $do(Mouth Slightly Open = 1)$ to affect the probability of $Smiling = 1$, i.e., $\mathbb{P}(Smiling = 1|do(Mouth Slightly Open = 1)) = \mathbb{P}(Smiling = 1) = 0.48$. However on the bottom row, conditioning on $Mouth Slightly Open = 1$ increases the proportion of smiling images ($0.48 \rightarrow 0.76$ in the dataset), although 10 images may not be enough to show this difference statistically.



(a) Intervening vs Conditioning on Narrow Eyes, Top: Intervene Narrow Eyes=1, Bottom: Condition Narrow Eyes=1

Figure 12: Intervening/Conditioning on Narrow Eyes label in Causal Graph 1. Since $Smiling \rightarrow Narrow Eyes$ in Causal Graph 1, we do not expect $do(Narrow Eyes = 1)$ to affect the probability of $Smiling = 1$, i.e., $\mathbb{P}(Smiling = 1|do(Narrow Eyes = 1)) = \mathbb{P}(Smiling = 1) = 0.48$. However on the bottom row, conditioning on $Narrow Eyes = 1$ increases the proportion of smiling images ($0.48 \rightarrow 0.59$ in the dataset), although 10 images may not be enough to show this difference statistically.

that the image quality for rare labels deteriorates. Please see Figure 20 in the appendix. Then for the labels *Wearing Lipstick*, *Mustache*, *Bald*, and *Narrow Eyes*, we illustrate the difference between

interventional and conditional sampling when the label is 1. (Figures 13a-16a).



(a) Intervening vs Conditioning on Mustache, Top: Intervene Mustache=1, Bottom: Condition Mustache=1

Figure 13: Intervening/Conditioning on Mustache label in Causal Graph 1. Since $Male \rightarrow Mustache$ in Causal Graph 1, we do not expect $do(Mustache = 1)$ to affect the probability of $Male = 1$, i.e., $\mathbb{P}(Male = 1|do(Mustache = 1)) = \mathbb{P}(Male = 1) = 0.42$. Accordingly, the top row shows both males and females with mustaches, even though the generator never sees the label combination $\{Male = 0, Mustache = 1\}$ during training. The bottom row of images sampled from the conditional distribution $\mathbb{P}(.|Mustache = 1)$ shows only male images because in the dataset $\mathbb{P}(Male = 1|Mustache = 1) \approx 1$.



(a) Intervening vs Conditioning on Bald, Top: Intervene Bald=1, Bottom: Condition Bald=1

Figure 14: Intervening/Conditioning on Bald label in Causal Graph 1. Since $Male \rightarrow Bald$ in Causal Graph 1, we do not expect $do(Bald = 1)$ to affect the probability of $Male = 1$, i.e., $\mathbb{P}(Male = 1|do(Bald = 1)) = \mathbb{P}(Male = 1) = 0.42$. Accordingly, the top row shows both bald males and bald females. The bottom row of images sampled from the conditional distribution $\mathbb{P}(.|Bald = 1)$ shows only male images because in the dataset $\mathbb{P}(Male = 1|Bald = 1) \approx 1$.



(a) Intervening vs Conditioning on Mouth Slightly Open, Top: Intervene Mouth Slightly Open=1, Bottom: Condition Mouth Slightly Open=1

Figure 15: Intervening/Conditioning on Mouth Slightly Open label in Causal Graph 1. Since $Smiling \rightarrow MouthSlightlyOpen$ in Causal Graph 1, we do not expect $do(Mouth Slightly Open = 1)$ to affect the probability of $Smiling = 1$, i.e., $\mathbb{P}(Smiling = 1|do(Mouth Slightly Open = 1)) = \mathbb{P}(Smiling = 1) = 0.48$. However on the bottom row, conditioning on $Mouth Slightly Open = 1$ increases the proportion of smiling images ($0.48 \rightarrow 0.76$ in the dataset), although 10 images may not be enough to show this difference statistically.



(a) Intervening vs Conditioning on Narrow Eyes, Top: Intervene Narrow Eyes=1, Bottom: Condition Narrow Eyes=1

Figure 16: Intervening/Conditioning on Narrow Eyes label in Causal Graph 1. Since $Smiling \rightarrow Narrow\ Eyes$ in Causal Graph 1, we do not expect $do(Narrow\ Eyes = 1)$ to affect the probability of $Smiling = 1$, i.e., $\mathbb{P}(Smiling = 1 | do(Narrow\ Eyes = 1)) = \mathbb{P}(Smiling = 1) = 0.48$. However on the bottom row, conditioning on $Narrow\ Eyes = 1$ increases the proportion of smiling images ($0.48 \rightarrow 0.59$ in the dataset), although 10 images may not be enough to show this difference statistically. As a rare artifact, in the dark image in the third column the generator appears to rule out the possibility of $Narrow\ Eyes = 0$ instead of demonstrating $Narrow\ Eyes = 1$.

9 Conclusion

We proposed a novel generative model with label inputs. In addition to being able to create samples *conditional* on labels, our generative model can also sample from the *interventional* distributions. Our theoretical analysis provides provable guarantees about correct sampling under such interventions and conditionings. The difference between these two sampling mechanisms is the key for causality. Interestingly, causality leads to generative models that are more creative since they can produce samples that are different from their training samples in multiple ways. We have illustrated this point for two models (CausalGAN and CausalBEGAN) and numerous label examples.

Acknowledgements

We thank Ajil Jalal for the helpful discussions.

References

- [1] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *arXiv pre-print*, 2017.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *arXiv pre-print*, 2017.
- [3] Mohammad Taha Bahadori, Krzysztof Chalupka, Edward Choi, Robert Chen, Walter F. Stewart, and Jimeng Sun. Causal regularization. In *arXiv pre-print*, 2017.
- [4] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. In *arXiv pre-print*, 2017.
- [5] Michel Besserve, Naji Shajarisales, Bernhard Schölkopf, and Dominik Janzing. Group invariance principles for causal generative models. In *arXiv pre-print*, 2017.
- [6] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed sensing using generative models. In *ICML 2017*, 2017.
- [7] Yan Chen, Xi Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of NIPS 2016*, Barcelona, Spain, December 2016.

- [8] Chris Donahue, Akshay Balsubramani, Julian McAuley, and Zachary C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. In *arXiv pre-print*, 2017.
- [9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.
- [10] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017.
- [11] Frederick Eberhardt. Phd thesis. *Causation and Intervention (Ph.D. Thesis)*, 2007.
- [12] Jalal Etesami and Negar Kiyavash. Discovering influence structure. In *IEEE ISIT*, 2016.
- [13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS 2014*, Montreal, CA, December 2014.
- [14] Matthew Graham and Amos Storkey. Asymptotically exact inference in differentiable generative models. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 499–508, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.
- [15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *arXiv pre-print*, 2017.
- [16] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014.
- [17] Patrik O Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Proceedings of NIPS 2008*, 2008.
- [18] Antti Hyttinen, Frederick Eberhardt, and Patrik Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013.
- [19] Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *ICML’17*, 2017.
- [20] Murat Kocaoglu, Alexandros G. Dimakis, Sriram Vishwanath, and Babak Hassibi. Entropic causal inference. In *AAAI’17*, 2017.
- [21] Ioannis Kontoyiannis and Maria Skoularidou. Estimating the directed information and testing for causality. *IEEE Trans. Inf. Theory*, 62:6053–6067, Aug. 2016.
- [22] Ming-Yu Liu and Tuzel Oncel. Coupled generative adversarial networks. In *Proceedings of NIPS 2016*, Barcelona, Spain, December 2016.
- [23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [24] David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin. Towards a learning theory of cause-effect inference. In *Proceedings of ICML 2015*, 2015.
- [25] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Schölkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of CVPR 2017*, Honolulu, CA, July 2017.
- [26] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *arXiv pre-print*, 2016.
- [27] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. In *arXiv pre-print*, 2016.
- [28] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. In *arXiv pre-print*, 2016.
- [29] Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 489–498, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

- [30] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *arXiv pre-print*, 2016.
- [31] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.
- [32] Christopher Quinn, Negar Kiyavash, and Todd Coleman. Directed information graphs. *IEEE Trans. Inf. Theory*, 61:6887–6909, Dec. 2015.
- [33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv pre-print*, 2015.
- [34] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS’16*, 2016.
- [35] Karthikeyan Shanmugam, Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. In *NIPS 2015*, 2015.
- [36] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. A Bradford Book, 2001.
- [37] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Proceedings of NIPS 2016*, Barcelona, Spain, December 2016.
- [38] Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Adversarial neural machine translation. In *arXiv pre-print*, 2017.

10 Appendix

10.1 Proof of Lemma 1

The proof follows the same lines as in the proof for the optimal discriminator. Consider the objective

$$\begin{aligned} & \rho \mathbb{E}_{x \sim p_{\text{data}}^1(x)} [\log(D_{LR}(x))] + (1 - \rho) \mathbb{E}_{x \sim p_{\text{data}}^0(x)} [\log(1 - D_{LR}(x))] \\ &= \int \rho p_r(x|l=1) \log(D_{LR}(x)) + (1 - \rho) p_r(x|l=0) \log(1 - D_{LR}(x)) dx \end{aligned} \quad (13)$$

Since $0 < D_{LR} < 1$, D_{LR} that maximizes (3) is given by

$$D_{LR}^*(x) = \frac{\rho p_r(x|l=1)}{p_r(x|l=1)\rho + p_r(x|l=0)(1-\rho)} = \frac{\rho p_r(x|l=1)}{p_r(x)} = p_r(l=1|x) \quad (14)$$

10.2 Proof of Theorem 2

Define $C(G)$ as the generator loss for when discriminator, Labeler and Anti-Labeler are at their optimum. p_{data} , p_r , \mathbb{P}_{data} and \mathbb{P}_r are used exchangeably for the data distribution. Then we have,

$$\begin{aligned} C(G) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D^*(x))] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D^*(x))] - \mathbb{E}_{x \sim p_g(x)} [\log(D^*(x))] \\ &\quad - (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(1 - D_{LR}(x))] - \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(D_{LR}(x))] \\ &\quad + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(1 - D_{LG}(x))] + \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(D_{LG}(x))] \\ &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D^*(x))] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D^*(x))] - \mathbb{E}_{x \sim p_g(x)} [\log(D^*(x))] \\ &\quad - (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(\mathbb{P}_r(l=0|x))] - \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(\mathbb{P}_r(l=1|x))] \\ &\quad + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(\mathbb{P}_g(l=0|x))] + \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(\mathbb{P}_g(l=1|x))] \\ &= \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\log \left(\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right) \right] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{p_g(x)}{p_{\text{data}}(x)} \right) \right] \\ &\quad - (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(\mathbb{P}_r(l=0|x))] - \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(\mathbb{P}_r(l=1|x))] \\ &\quad + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} [\log(\mathbb{P}_g(l=0|x))] + \rho \mathbb{E}_{x \sim p_g^1(x)} [\log(\mathbb{P}_g(l=1|x))] \end{aligned} \quad (15)$$

Using Bayes' rule, we can write $\mathbb{P}(l=1|x) = \frac{\mathbb{P}(x|l=1)\rho}{\mathbb{P}(x)}$ and $\mathbb{P}(l=0|x) = \frac{\mathbb{P}(x|l=0)(1-\rho)}{\mathbb{P}(x)}$. Then we have the following:

$$\begin{aligned} C(G) &= -1 + KL(p_r \parallel \frac{p_r + p_g}{2}) + KL(p_g \parallel p_r) + H(\rho) \\ &\quad + (1 - \rho) KL(p_g^0 \parallel p_r^0) + \rho KL(p_g^1 \parallel p_r^1) - (1 - \rho) KL(p_g^0 \parallel p_r) - \rho KL(p_g^1 \parallel p_r) \\ &\quad + (1 - \rho) \mathbb{E}_{x \sim p_g^0(x)} \left[\log \left(\frac{p_g^0(1-\rho)}{p_g} \right) \right] + \rho \mathbb{E}_{x \sim p_g^1(x)} \left[\log \left(\frac{p_g^1\rho}{p_g} \right) \right], \end{aligned}$$

where $H(\rho)$ stands for the binary entropy function. Notice that we have

$$\begin{aligned}
& -(1-\rho)KL(p_g^0 \parallel p_r) - \rho KL(p_g^1 \parallel p_r) \\
&= -(1-\rho) \int p_g^0(x) \log(p_g^0(x)) dx - \rho \int p_g^1 \log(p_g^1(x)) dx + (1-\rho) \int p_g^0(x) \log(p_r(x)) dx \\
&\quad + \rho \int p_g^1(x) \log(p_r(x)) dx \\
&= -(1-\rho) \int p_g^0(x) \log(p_g^0(x)) dx - \rho \int p_g^1 \log(p_g^1(x)) dx + \int p_g(x) \log(p_r(x)) dx \\
&= -(1-\rho) \int p_g^0(x) \log(p_g^0(x)) dx - \rho \int p_g^1 \log(p_g^1(x)) dx - KL(p_g \parallel p_r) + \int p_g(x) \log(p_g(x)) dx.
\end{aligned}$$

Also notice that we have

$$\begin{aligned}
& (1-\rho) \mathbb{E}_{x \sim p_g^0(x)} \left[\log \left(\frac{p_g^0(1-\rho)}{p_g} \right) \right] + \rho \mathbb{E}_{x \sim p_g^1(x)} \left[\log \left(\frac{p_g^1 \rho}{p_g} \right) \right] \\
&= - \int p_g(x) \log(p_g(x)) dx - H(\rho) + (1-\rho) \int p_g^0(x) \log(p_g^0(x)) dx + \rho \int p_g^1(x) \log(p_g^1(x)) dx
\end{aligned}$$

Substituting this into the above equation and combining terms, we get

$$C(G) = -1 + KL(p_r \parallel \frac{p_r + p_g}{2}) + (1-\rho)KL(p_g^0 \parallel p_r^0) + \rho KL(p_g^1 \parallel p_r^1)$$

Observe that for $p_g^0 = p_r^0$ and $p_g^1 = p_r^1$, we have $p_g = p_r$, yielding $C(G) = -1$. Finally, since KL divergence is always non-negative we have $C(G) \geq -1$, concluding the proof. \square

10.3 Proof of Corollary 2

Since C is a causal implicit generative model for the causal graph D , by definition it is consistent with the causal graph D . Since in a conditional GAN, generator G is given the noise terms and the labels, it is easy to see that the concatenated generator neural network $G(C(Z_1), Z_2)$ is consistent with the causal graph D' , where $D' = (\mathcal{V} \cup \{Image\}, E \cup \{(V_1, Image), (V_2, Image), \dots (V_n, Image)\})$. Assume that C and G are perfect, i.e., they sample from the true label joint distribution and conditional image distribution. Then the joint distribution over the generated labels and image is the true distribution since $\mathbb{P}(Image, Label) = \mathbb{P}(Image|Label)\mathbb{P}(Label)$. By Proposition 1, the concatenated model can sample from the true observational and interventional distributions. Hence, the concatenated model is a causal implicit generative model for graph D' .

10.4 CausalGAN Architecture and Loss for Multiple Labels

In this section, we explain the modifications required to extend the proof to the case with multiple binary labels, or a label variable with more than 2 states in general. p_{data} , p_r , \mathbb{P}_{data} and \mathbb{P}_r are used exchangeably for the data distribution in the following.

Consider Figure 4 in the main text. Labeler outputs the scalar $D_{LR}(x)$ given an image x . With the given loss function in (3), i.e., when there is a single binary label l , when we show in Section 10.1 that the optimum Labeler $D_{LR}^*(x) = p_r(l=1|X=x)$. We first extend the Labeler objective as follows: Suppose we have d binary labels. Then we allow the Labeler to output a 2^d dimensional

vector $D_{LR}(x)$, where $D_{LR}(x)[i]$ is the i^{th} coordinate of this vector. The Labeler then solves the following optimization problem:

$$\max_{D_{LR}} \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{p_r^j} \log(D_{LR}(x)[j]), \quad (16)$$

where $p_r^j(x) := \mathbb{P}_r(X = x | l = j)$ and $\rho_j = \mathbb{P}_r(l = j)$. We have the following Lemma:

Lemma 3. *Consider a Labeler D_{LR} that outputs the 2^d -dimensional vector $D_{LR}(x)$ such that $\sum_{j=1}^{2^d} D_{LR}(x)[j] = 1$, where $x \sim p_r(x, l)$. Then the optimum Labeler with respect to the loss in (16) has $D_{LR}^*(x)[j] = p_r(l = j | x)$.*

Proof. Suppose $p_r(l = j | x) = 0$ for a set of (label, image) combinations. Then $p_r(x, l = j) = 0$, hence these label combinations do not contribute to the expectation. Thus, without loss of generality, we can consider only the combinations with strictly positive probability. We can also restrict our attention to the functions D_{LR} that are strictly positive on these (label, image) combinations; otherwise, loss becomes infinite, and as we will show we can achieve a finite loss. Consider the vector $D_{LR}(x)$ with coordinates $D_{LR}(x)[j]$ where $j \in [2^d]$. Introduce the discrete random variable $Z_x \in [2^d]$, where $\mathbb{P}(Z_x = j) = D_{LR}(x)[j]$. The Labeler loss can be written as

$$\min -\mathbb{E}_{(x,l) \sim p_r(x,l)} \log(\mathbb{P}(Z_x = j)) \quad (17)$$

$$= \min \mathbb{E}_{x \sim p_r(x)} KL(L_x \parallel Z_x) - H(L_x), \quad (18)$$

where L_x is the discrete random variable such that $\mathbb{P}(L_x = j) = \mathbb{P}_r(l = j | x)$. $H(L_x)$ is the Shannon entropy of L_x , and it only depends on the data. Since KL divergence is greater than zero and $p(x)$ is always non-negative, the loss is lower bounded by $-H(L_x)$. Notice that this minimum can be achieved by satisfying $\mathbb{P}(Z_x = j) = \mathbb{P}_r(l = j | x)$. Since KL divergence is minimized if and only if the two random variables have the same distribution, this is the unique optimum, i.e., $D_{LR}^*(x)[j] = \mathbb{P}_r(l = j | x)$. □

The lemma above simply states that the optimum Labeler network will give the posterior probability of a particular label combination, given the observed image. In practice, the constraint that the coordinates sum to 1 could be satisfied by using a softmax function in the implementation. Next, we have the corresponding loss function and lemma for the Anti-Labeler network. The Anti-Labeler solves the following optimization problem

$$\max_{D_{LG}} \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{p_g^j} \log(D_{LG}(x)[j]), \quad (19)$$

where $p_g^j(x) := \mathbb{P}(G(z, l) = x | l = j)$ and $\rho_j = \mathbb{P}(l = j)$. We have the following Lemma:

Lemma 4. *The optimum Anti-Labeler has $D_{LG}^*(x)[j] = \mathbb{P}_g(l = j | x)$.*

Proof. The proof is the same as the proof of Lemma 3, since Anti-Labeler does not have control over the joint distribution between the generated image and the labels given to the generator, and cannot optimize the conditional entropy of labels given the image under this distribution. □

For a fixed discriminator, Labeler and Anti-Labeler, the generator solves the following optimization problem:

$$\begin{aligned}
& \min_G \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{1 - D(x)}{D(x)} \right) \right] \\
& - \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} [\log(D_{LR}(X)[j])] \\
& + \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} [\log(D_{LG}(X)[j])] .
\end{aligned} \tag{20}$$

We then have the following theorem, that shows that the optimal generator samples from the class conditional image distributions given a particular label combination:

Theorem 3 (Theorem 1 formal for multiple binary labels). *Define $C(G)$ as the generator loss for when discriminator, Labeler and Anti-Labeler are at their optimum obtained from (20). The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g^j = p_{\text{data}}^j, \forall j \in [2^d]$, i.e., if and only if given a d -dimensional label vector l , generator samples from the class conditional image distribution, i.e., $\mathbb{P}(G(z, l) = x) = p_{\text{data}}(x|l)$.*

Proof. Substituting the optimum values for the Discriminator, Labeler and Anti-Labeler networks, we get the virtual training criterion $C(G)$ as

$$\begin{aligned}
C(G) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D^*(x))] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D^*(x))] - \mathbb{E}_{x \sim p_g(x)} [\log(D^*(x))] \\
& - \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log(D_{LR}^*(x)[j]) \\
& + \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log(D_{LG}^*(x)[j]) \\
&= \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\log \left(\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right) \right] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{p_g(x)}{p_{\text{data}}(x)} \right) \right] \\
& - \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log(p_r(l = j|X = x)) \\
& + \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log(p_g(l = j|X = x))
\end{aligned} \tag{21}$$

Using Bayes' rule, we can write $\mathbb{P}(l = j|x) = \frac{\mathbb{P}(x|l=j)\rho_j}{\mathbb{P}(x)}$. Then we have the following:

$$\begin{aligned}
C(G) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\log \left(\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right) \right] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{p_g(x)}{p_{\text{data}}(x)} \right) \right] \\
&\quad - \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log \left(\frac{p_r^j(x) \rho_j}{p_r(x)} \right) \\
&\quad + \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log \left(\frac{p_g^j(x) \rho_j}{p_g(x)} \right), \\
&= -1 + KL(p_r \parallel \frac{p_r + p_g}{2}) + KL(p_g \parallel p_r) + H(l) \\
&\quad + \sum_{j=1}^{2^d} \rho_j KL(p_g^j \parallel p_r^j) - \sum_{j=1}^{2^d} \rho_j KL(p_g^j \parallel p_r) \\
&\quad + \sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log \left(\frac{p_g^j(x) \rho_j}{p_g(x)} \right).
\end{aligned}$$

Notice that we have

$$\begin{aligned}
&- \sum_{j=1}^{2^d} \rho_j KL(p_g^j \parallel p_r) \\
&= - \sum_{j=1}^{2^d} \rho_j \int p_g^j \log(p_g^j) dx - KL(p_g \parallel p_r) + \int p_g(x) \log(p_g(x)) dx
\end{aligned}$$

Also notice that we have

$$\begin{aligned}
&\sum_{j=1}^{2^d} \rho_j \mathbb{E}_{x \sim p_g^j(x)} \log \left(\frac{p_g^j(x) \rho_j}{p_g(x)} \right) \\
&= - \int p_g(x) \log(p_g(x)) dx - H(l) + \sum_{j=1}^{2^d} \rho_j \int p_g^j(x) \log(p_g^j(x)) dx
\end{aligned}$$

Substituting this into the above equation and combining terms, we get

$$C(G) = -1 + KL(p_r \parallel \frac{p_r + p_g}{2}) + \sum_{j=1}^{2^d} \rho_j KL(p_g^j \parallel p_r^j)$$

Observe that for $p_g^j = p_r^j, \forall j \in [d]$, we have $p_g = p_r$, yielding $C(G) = -1$. Finally, since KL divergence is always non-negative we have $C(G) \geq -1$, concluding the proof. \square

10.5 Alternate CausalGAN Architecture for d Labels

In this section, we provide the theoretical guarantees for the implemented CausalGAN architecture with d labels. Later we show that these guarantees are sufficient to prove that the global optimal generator samples from the class conditional distributions for a practically relevant class of distributions.

First, let us restate the loss functions more formally. Note that $D_{LR}(x), D_{LG}(x)$ are d -dimensional vectors. The Labeler solves the following optimization problem:

$$\max_{D_{LR}} \rho_j \mathbb{E}_{x \sim p_r^{j1}} \log(D_{LR}(x)[j]) + (1 - \rho_j) \mathbb{E}_{x \sim p_r^{j0}} \log(1 - D_{LR}(x)[j]). \quad (22)$$

where $p_r^{j0}(x) := \mathbb{P}(X = x | l_j = 0)$, $p_r^{j1}(x) := \mathbb{P}(X = x | l_j = 1)$ and $\rho_j = \mathbb{P}(l_j = 1)$. For a fixed generator, the Anti-Labeler solves the following optimization problem:

$$\max_{D_{LG}} \rho_j \mathbb{E}_{p_g^{j1}} \log(D_{LG}(x)[j]) + (1 - \rho_j) \mathbb{E}_{p_g^{j0}} \log(1 - D_{LG}(x)[j]), \quad (23)$$

where $p_g^{j0}(x) := \mathbb{P}_g(x | l_j = 0)$, $p_g^{j1}(x) := \mathbb{P}_g(x | l_j = 1)$. For a fixed discriminator, Labeler and Anti-Labeler, the generator solves the following optimization problem:

$$\begin{aligned} \min_G & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left(\frac{1 - D(x)}{D(x)} \right) \right] \\ & - \frac{1}{d} \sum_{j=1}^d \rho_j \mathbb{E}_{x \sim p_g^{j1}(x)} [\log(D_{LR}(X)[j])] - (1 - \rho_j) \mathbb{E}_{x \sim p_g^{j0}(x)} [\log(1 - D_{LR}(X)[j])] \\ & + \frac{1}{d} \sum_{j=1}^d \rho_j \mathbb{E}_{x \sim p_g^{j1}(x)} [\log(D_{LG}(X)[j])] + (1 - \rho_j) \mathbb{E}_{x \sim p_g^{j0}(x)} [\log(1 - D_{LG}(X)[j])]. \end{aligned} \quad (24)$$

We have the following proposition, which characterizes the optimum generator, for optimum Labeler, Anti-Labeler and Discriminator:

Proposition 3. *Define $C(G)$ as the generator loss for when discriminator, Labeler and Anti-Labeler are at their optimum obtained from (24). The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g(x | l_i) = p_r(x | l_i), \forall i \in [d]$ and $p_g(x) = p_r(x)$.*

Proof. Proof follows the same lines as in the proof of Theorem 2 and Theorem 3 and is omitted. \square

Thus we have

$$p_r(x, l_i) = p_g(x, l_i), \forall i \in [d] \text{ and } p_r(x) = p_g(x). \quad (25)$$

However, this does not in general imply $p_r(x, l_1, l_2, \dots, l_d) = p_g(x, l_1, l_2, \dots, l_d)$, which is equivalent to saying the generated distribution samples from the class conditional image distributions. To guarantee the correct conditional sampling given all labels, we introduce the following assumption: We assume that the image x determines all the labels. This assumption is very relevant in practice. For example, in the CelebA dataset, which we use, the label vector, e.g., whether the person is a male or female, with or without a mustache, can be thought of as a deterministic function of the image. When this is true, we can say that $p_r(l_1, l_2, \dots, l_n | x) = p_r(l_1 | x) p_r(l_2 | x) \dots p_r(l_n | x)$.

We need the following lemma, where kronecker delta function refers to the functions that take the value of 1 only on a single point, and 0 everywhere else:

Lemma 5. *Any discrete joint probability distribution, where all the marginal probability distributions are kronecker delta functions is the product of these marginals.*

Proof. Let $\delta_{\{x=u\}}$ be the kronecker delta function which is 1 if $x = u$ and is 0 otherwise. Consider a joint distribution $p(X_1, X_2, \dots, X_n)$, where $p(X_i) = \delta_{\{X_i=u_i\}}, \forall i \in [n]$, for some set of elements $\{u_i\}_{i \in [n]}$. We will show by contradiction that the joint probability distribution is zero everywhere except at (u_1, u_2, \dots, u_n) . Then, for the sake of contradiction, suppose for some $v = (v_1, v_2, \dots, v_n) \neq$

(u_1, u_2, \dots, u_n) , $p(v_1, v_2, \dots, v_n) \neq 0$. Then $\exists j \in [n]$ such that $v_j \neq u_j$. Then we can marginalize the joint distribution as

$$p(v_j) = \sum_{X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n} p(X_1, \dots, X_{j-1}, v_j, X_{j+1}, \dots, X_n) > 0, \quad (26)$$

where the inequality is due to the fact that the particular configuration (v_1, v_2, \dots, v_n) must have contributed to the summation. However this contradicts with the fact that $p(X_j) = 0, \forall X_j \neq u_j$. Hence, $p(\cdot)$ is zero everywhere except at (u_1, u_2, \dots, u_n) , where it should be 1. \square

We can now simply apply the above lemma on the conditional distribution $p_g(l_1, l_2, \dots, l_d|x)$. Proposition 3 shows that the image distributions and the marginals $p_g(l_i|x)$ are true to the data distribution due to Bayes' rule. Since the vector (l_1, \dots, l_n) is a deterministic function of x by assumption, $p_r(l_i|x)$ are kronecker delta functions, and so are $p_g(l_i|x)$ by Proposition 3. Thus, since the joint $p_g(x, l_1, l_2, \dots, l_d)$ satisfies the condition that every marginal distribution $p(l_i|x)$ is a kronecker delta function, then it must be a product distribution by Lemma 5. Thus we can write

$$p_g(l_1, l_2, \dots, l_d|x) = p_g(l_1|x)p_g(l_2|x) \dots p_g(l_n|x).$$

Then we have the following chain of equalities.

$$\begin{aligned} p_r(x, l_1, l_2, \dots, l_d) &= p_r(l_1, \dots, l_n|x)p_r(x) \\ &= p_r(l_1|x)p_r(l_2|x) \dots p_r(l_n|x)p_r(x) \\ &= p_g(l_1|x)p_g(l_2|x) \dots p_g(l_n|x)p_g(x) \\ &= p_g(l_1, l_2, \dots, l_d|x)p_g(x) \\ &= p_g(x, l_1, l_2, \dots, l_d). \end{aligned}$$

Thus, we also have $p_r(x|l_1, l_2, \dots, l_n) = p_g(x|l_1, l_2, \dots, l_n)$ since $p_r(l_1, l_2, \dots, l_n) = p_g(l_1, l_2, \dots, l_n)$, concluding the proof that the optimum generator samples from the class conditional image distributions.

10.6 Additional Simulations for Causal Controller

First, we evaluate the effect of using the wrong causal graph on an artificially generated dataset. Figure 17 shows the scatter plot for the two coordinates of a three dimensional distribution. As we observe, using the correct graph gives the closest scatter plot to the original data, whereas using the wrong Bayesian network, collider graph, results in a very different distribution.

Second, we expand on the causal graphs used for experiments for the CelebA dataset. The graph Causal Graph 1 (G1) is as illustrated in Figure 5. The graph cG1, which is a completed version of G1, is the complete graph associated with the ordering: Young, Male, Eyeglasses, Bald, Mustache, Smiling, Wearing Lipstick, Mouth Slightly Open, Narrow Eyes. For example, in cG1 Male causes Smiling because Male comes before Smiling in the ordering. The graph rcG1 is associated with the reverse ordering.

Next, we check the effect of using the incorrect Bayesian network for the data. The causal graph G1 generates Male and Young independently, which is incorrect in the data. Comparison of pairwise distributions in Table 1 demonstrate that for G1 a reasonable approximation to the true distribution is still learned for {Male, Young} jointly. For cG1 a nearly perfect distributional approximation is learned. Furthermore we show that despite this inaccuracy, both graphs G1 and cG1 lead to Causal

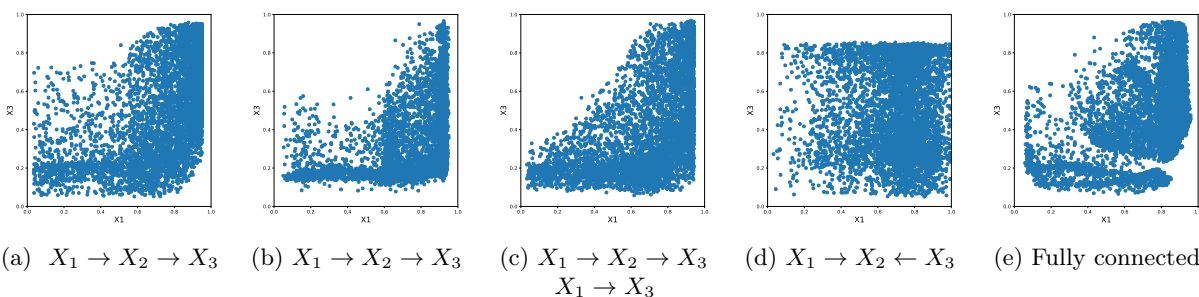


Figure 17: Synthetic data experiments: (a) Scatter plot for actual data. Data is generated using the causal graph $X_1 \rightarrow X_2 \rightarrow X_3$. (b) Generated distribution when generator causal model is $X_1 \rightarrow X_2 \rightarrow X_3$. (c) Generated distribution when generator causal model is $X_1 \rightarrow X_2 \rightarrow X_3$. (d) Generated distribution when generator causal model is $X_1 \rightarrow X_2 \leftarrow X_3$. (e) Generated distribution when generator is from a fully connected last layer of a 5 layer FF neural net.

Label Pair		Male	
		0	1
Young	0	0.140.07	0.090.15
	1	0.470.51	0.29[0.27](0.26)
Mustache	0	0.610.58	0.340.38
	1	0.000.00	0.040.04

Table 1: Pairwise marginal distribution for select label pairs when Causal Controller is trained on $G1$ in plain text, its completion $cG1$ [square brackets], and the true pairwise distribution(in parentheses). Note that $G1$ treats Male and Young labels as independent, but does not completely fail to generate a reasonable (product of marginals) approximation. Also note that when an edge is added $Young \rightarrow Male$, the learned distribution is nearly exact. Note that both graphs contain the edge $Male \rightarrow Mustache$ and so are able to learn that women have no mustaches.

Controllers that never output the label combination $\{\text{Female}, \text{Mustache}\}$, which will be important later.

Wasserstein GAN in its original form (with Lipshitz discriminator) assures convergence in distribution of the Causal Controller output to the discretely supported distribution of labels. We use a slightly modified version of Wasserstein GAN with a penalized gradient[15]. We first demonstrate that learned outputs actually have "approximately discrete" support. In Figure 7a, we sample the joint label distribution 1000 times, and make a histogram of the (all) scalar outputs corresponding to any label.

Although Figure 7b demonstrates conclusively good convergence for both graphs, TVD is not always intuitive. For example, "how much can each marginal be off if there are 9 labels and the TVD is 0.14?". To expand upon Figure 2 where we showed that the causal controller learns the correct distribution for a pairwise subset of nodes, here we also show that both Causal Graph 1 ($G1$) and the completion we define ($cG1$) allow training of very reasonable marginal distributions for all labels (Table 1) that are not off by more than 0.03 for the worst label. $\mathbb{P}_D(L = 1)$ is the probability that the label is 1 in the dataset, and $\mathbb{P}_G(L = 1)$ is the probability that the generated label is (around a small neighborhood of) 1.

Label, L	$\mathbb{P}_{G1}(L = 1)$	$\mathbb{P}_{cG1}(L = 1)$	$\mathbb{P}_D(L = 1)$
Bald	0.02244	0.02328	0.02244
Eyeglasses	0.06180	0.05801	0.06406
Male	0.38446	0.41938	0.41675
Mouth Slightly Open	0.49476	0.49413	0.48343
Mustache	0.04596	0.04231	0.04154
Narrow Eyes	0.12329	0.11458	0.11515
Smiling	0.48766	0.48730	0.48208
Wearing Lipstick	0.48111	0.46789	0.47243
Young	0.76737	0.77663	0.77362

Table 2: Marginal distribution of pretrained Causal Controller labels when Causal Controller is trained on Causal Graph 1(P_{G1}) and its completion(P_{cG1}), where $cG1$ is the (nonunique) largest DAG containing $G1$ (see appendix). The third column lists the actual marginal distributions in the dataset

10.7 Additional Simulations for CausalGAN

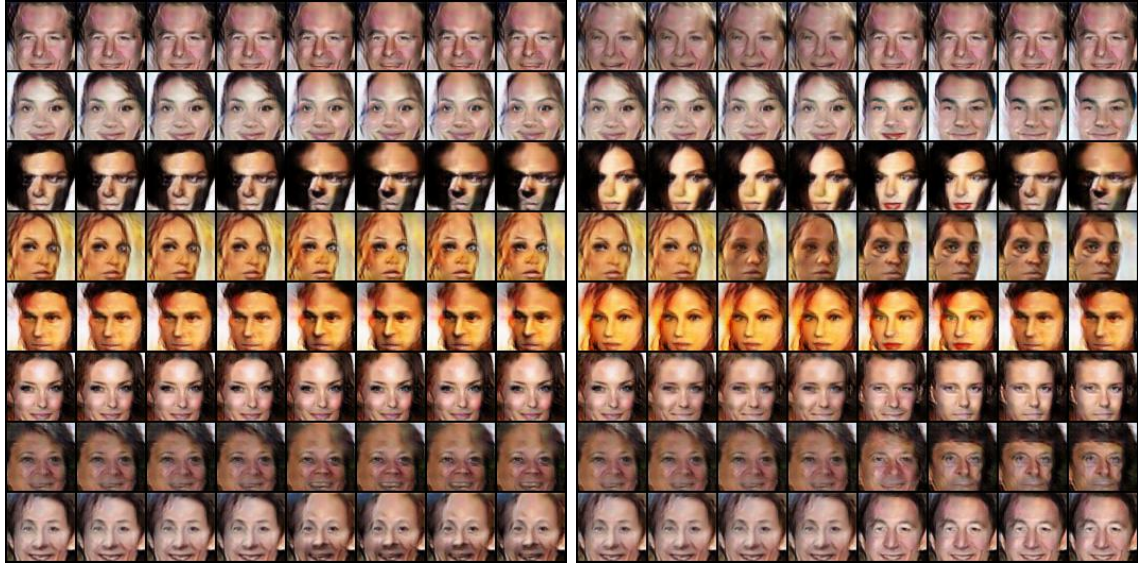
In this section, we provide additional simulations for CausalGAN. In Figures 18a-18d, we show the conditional image generation properties of CausalGAN by sweeping a single label from 0 to 1 while keeping all other inputs/labels fixed. In Figure 19, to examine the degree of mode collapse and show the image diversity, we show 256 randomly sampled images.

10.8 Additional CausalBEGAN Simulations

In this section, we provide additional simulation results for CausalBEGAN. First we show that although our third margin term b_3 introduces complications, it can not be ignored. Figure 20 demonstrates that omitting the third margin on the image quality of rare labels.

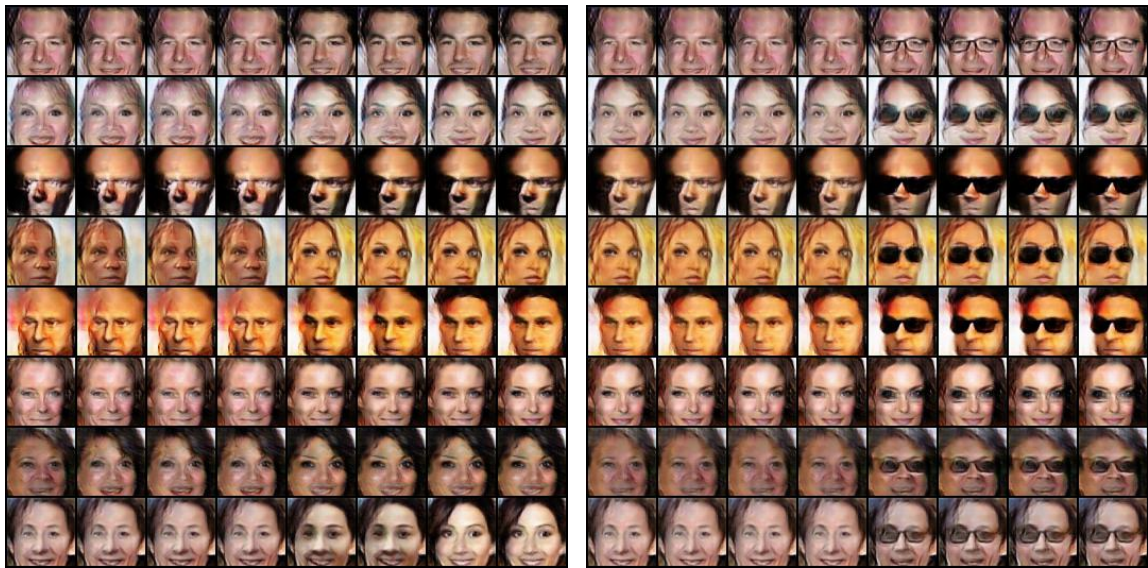
Furthermore just as the setup in BEGAN permitted the definition of a scalar " \mathcal{M} ", which was monotonically decreasing during training, our definition permits an obvious extension $\mathcal{M}_{complete}$ (defined in 10) that preserves these properties. See Figure 21 to observe $\mathcal{M}_{complete}$ decreasing monotonically during training.

We also show the conditional image generation properties of CausalBEGAN by using "label sweeps" that move a single label input from 0 to 1 while keeping all other inputs fixed (Figures 22a-22d). It is interesting to note that while generators are often implicitly thought of as continuous functions, the generator in this CausalBEGAN architecture learns a discrete function with respect to its label input parameters. (Initially there is label interpolation, and later in the optimization label interpolation becomes more step function like (not shown)). Finally, to examine the degree of mode collapse and show the image diversity, we show a random sampling of 256 images (Figure 23).



(a) Interpolating Bald label

(b) Interpolating Male label



(c) Interpolating Young label

(d) Interpolating Eyeglasses label

Figure 18: The effect of interpolating a single label for CausalGAN, while keeping the noise terms and other labels fixed.

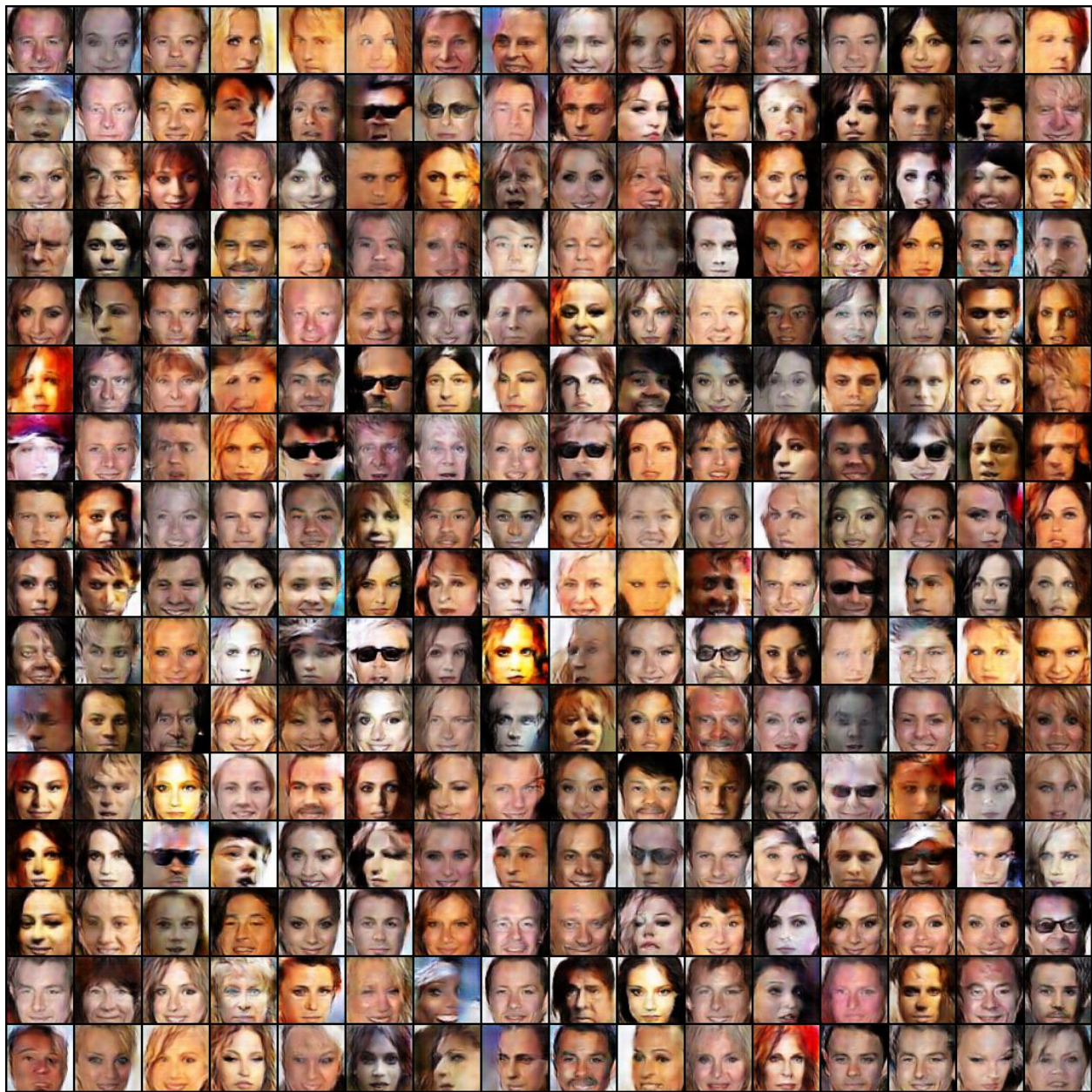


Figure 19: Diversity of the proposed CausalGAN showcased with 256 samples.



Figure 20: Omitting the nonobvious margin $b_3 = \gamma_3 * \text{relu}(b_1) - \text{relu}(b_2)$ results in poorer image quality particularly for rare labels such as mustache. We compare samples from two interventional distributions. Samples from $\mathbb{P}(.|\text{do}(\text{Mustache} = 1))$ (top) have much poorer image quality compared to those under $\mathbb{P}(.|\text{do}(\text{Mustache} = 0))$ (bottom).

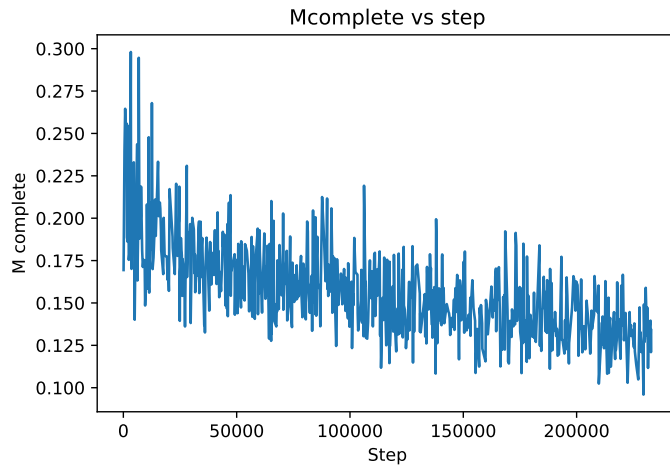
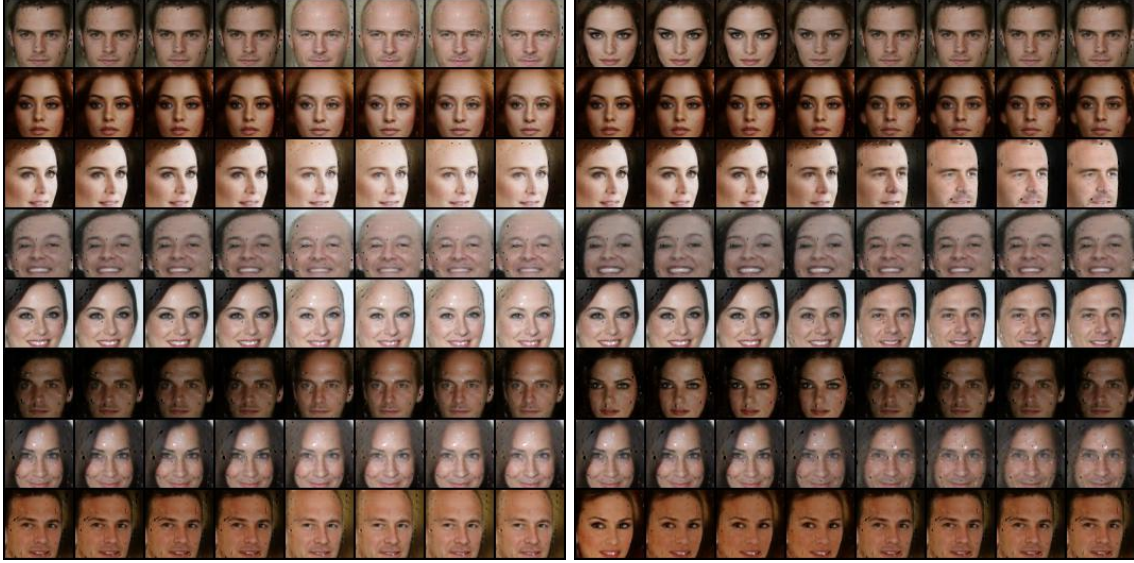
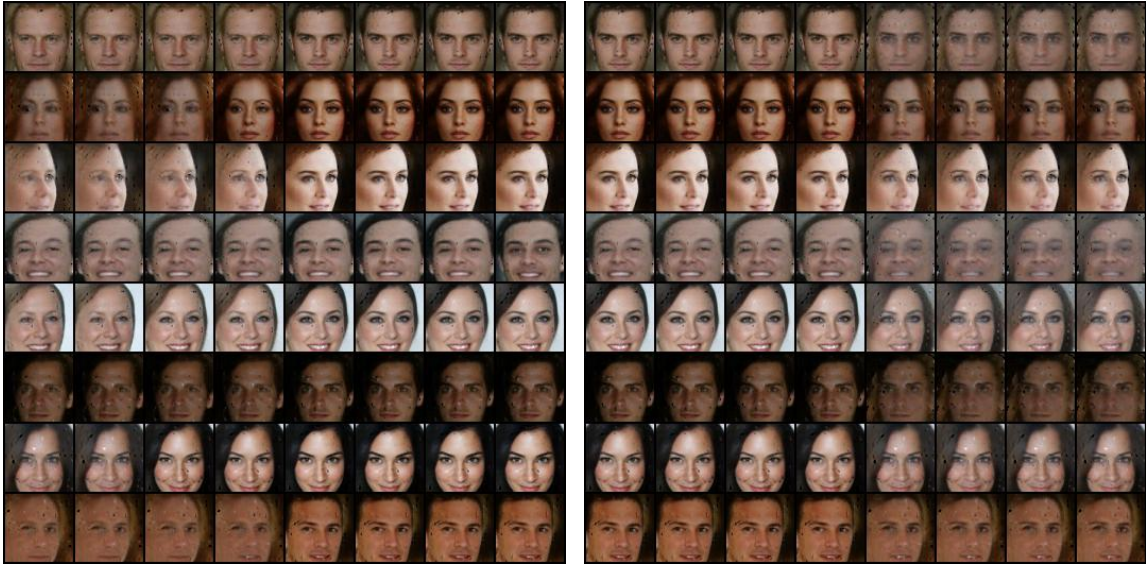


Figure 21: Convergence of CausalBEGAN captured through the parameter $\mathcal{M}_{complete}$.



(a) Interpolating Bald label

(b) Interpolating Male label



(c) Interpolating Young label

(d) Interpolating Eyeglasses label

Figure 22: The effect of interpolating a single label for CausalBEGAN, while keeping the noise terms and other labels fixed. Although most labels are properly captured, we see that eyeglasses label is not.



Figure 23: Diversity of Causal BEGAN showcased with 256 samples.

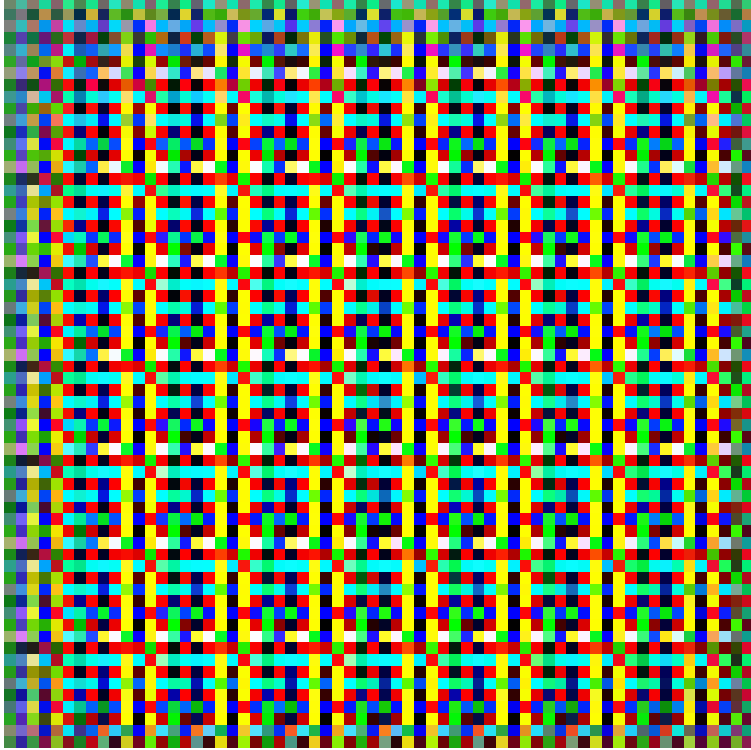


Figure 24: Failed Image generation for simultaneous label and image generation after 20k steps.

10.9 Directly Training Labels+Image Fails

In this section, we present the result of attempting to jointly train an implicit causal generative model for labels and the image. This approach treats the image as part of the causal graph. It is not clear how exactly to feed both labels and image to discriminator, but one way is to simply encode the label as a constant image in an additional channel. We tried this for Causal Graph 1 and observed that the image generation is not learned (Figure 24). One hypothesis is that the discriminator focuses on labels without providing useful gradients to the image generation.