GENERATIVE CAUSAL MODELLING TECHNIQUES FOR VISUAL MODEL EXPLANATION AND COUNTERFACTUAL AUDIO GENERATION

by

William Keith Taylor-Melanson

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
October 2023

To my late grandmother, Lorna Joan Melanson (Gram).

Table of Contents

List of Tables

List of Figures

Abstract

This thesis evaluates the effectiveness of two recent works in the area of nonlinear causal modelling, DeepSCM and ImageCFGen, in their ability to explain image classifiers and model audio data. First, techniques are presented for generating local counterfactual explanations of classifiers using DeepSCM and ImageCFGen models, and quantitative comparisons are made with techniques from the OmnixAI explanation toolkit. The metrics used to evaluate these explanation techniques on the Morpho-MNIST dataset indicate that the proposed methods of model explanation are more interpretable than those in the OmnixAI toolkit. Second, a causal graph is constructed on top of the attributes of the Audio-MNIST speech dataset in order to train DeepSCM and ImageCFGen models. To evaluate the models on this speech dataset, classifiers are trained and used to measure the consistency of attributes in observational and counterfactual data generated by DeepSCM and ImageCFGen. DeepSCM outperforms the standard ImageCFGen model on this task, but after fine-tuning the ImageCFGen model shows similar levels of agreement with attribute classifiers when compared with DeepSCM. In addition to attribute classifiers, a speaker classifier is trained to measure the ability of the causal models to maintain a speaker's voice when computing speech counterfactuals. The counterfactual models are compared with interventional models which do not perform abduction in order to provide a baseline to the experiment. DeepSCM is the only model which significantly improves over the interventional baseline, suggesting this model may be preferred over ImageCFGen to establish a causal model with the ability to produce believable speech counterfactuals. Finally, a dataset of North American Right Whale (NARW) calls is investigated, and a similar evaluation using attribute classifiers is performed which demonstrates the ability of these models to manipulate audio data.

Acknowledgements

First, I would like to thank Dr. Stan Matwin for helping me through the first year and a half of my degree and sparking my interest in causality. Thanks also to Dr. Martha Dais Ferreira, who regularly met with me early in my degree and gave me advice on writing and research. Thanks also goes to Dr. Zahra Sadeghi, who met with me regularly during the second year of my degree, provided advice regarding the writing of this thesis, and worked alongside me in developing the method of counterfactual explanations described in this work.

Thanks also goes to Dr. Thomas Trappenberg, who was kind enough to allow me to join his lab on short notice when unforeseen circumstances caused me to require a new supervisor. He has helped guide the story of this thesis and provided a wonderful learning environment at his lab in Halifax.

Finally, I would like to thank my parents, Dr. Jennifer Taylor and Dr. Glen Melanson. I appreciate them more than words can say, and their encouragement has kept me going during my degree at Dalhousie. They are my biggest supporters, and I will always be thankful to have them in my life.

Chapter 1

Introduction

1.1 Counterfactual Modelling

Causality has been of growing interest to Machine Learning (ML) researchers due to the ability for causal systems to adapt to shifts in data distributions. Causal systems understand not just associations between variables but explicitly how variables influence each other. This allows them to generalize far better than traditional machine learning strategies, as they learn families of distributions rather than a single distribution as a statistical model does [1]. Additionally, causal models aware of the structure of a data generation process and are able to create and reason about data not present in observational reality, i.e., counterfactual data. Counterfactual statements have been proposed for use in machine learning domains, including reinforcement learning and transfer learning, and have been argued previously to be crucial to formulating "hypotheses that can be empirically verified in a process akin to the scientific method" [2]. A typical counterfactual statement will often take the form "if it had been the case that ϕ then it would have been the case that ψ.", where ϕ and ψ are statements about the data generation process, such as the values of variables [3]. For example, "if the patient had been to convinced to quit smoking ten years earlier, they would not have died of lung cancer" is one such counterfactual concerned with the state of a deceased patient.

Two recent works, DeepSCM [4] and ImageCFGen [5], have been developed using generative deep learning models to approximate counterfactual data generation. DeepSCM utilizes a method based on variational autoencoders [6], while ImageCFGen uses a GAN-style architecture [7]. Both methods employ generative deep learning models to create structural causal models (SCM) on high-dimensional data. Low dimensional variables, such as univariate attributes of an image, can be modelled via maximum likelihood using the method of normalizing flows [6, 8] to build a full SCM

on top of a known or assumed set of causal associations between variables when combined with these generative models. The authors of both DeepSCM and ImageCFGen study the Morpho-MNIST causal dataset [9], a dataset of handwritten digits modified to have attributes of thickness, intensity, and slant as "causes" of the digit image. Additionally, the authors of ImageCFGen proposed a counterfactual importance score metric to measure the importance of binary attributes on a classifier, suggesting that causal models are feasible to use as components of a classifier explanation system.

The ImageCFGen algorithm also comes with a recommended fine-tuning procedure from its authors in order to produce more realistic counterfactuals. This procedure uses two losses, a typical reconstruction loss and a latent loss involving an expected value over a latent prior. This thesis presents a closed form for the latent loss term to simplify the implementation of fine-tuning, a proof of which is found in Appendix A.

To facilitate the experiments in this thesis, the DeepSCM and ImageCFGen algorithms were implemented using the Pytorch software library [10]. This was done in part to adapt DeepSCM to the datasets used in this work, and also due to no public implementation of ImageCFGen being available. Normalizing flow models, which model low-dimensional random variables using learned transformations, were implemented as needed using transformations and distributions from the Pyro probabilistic programming language [11] and trained using Pytorch optimizers. A public implementation of all experiments in this work has been made available on GitHub.[1]

1.2 Model Explanation

Counterfactual examples, which aim to provide an interpretable change to a classifier's input in order to change a classifier's decision, have been used to explain classifiers in previous works [12], and open-source toolkits exist for the creation of such model explanations. One such open-source toolkit is OmnixAI [13]. The computer vision explanation library of OmnixAI provides both counterfactual and contrastive explainers of image classifiers, with the counterfactual explainer coming from Wachter et al. [12] and the contrastive explainer coming from Dhurandhar et al. [14]. Both methods

[1]The repository is available at `https://github.com/wtaylor17/ImageCFGen-Pytorch`, and holds all experiments for this thesis despite its name indicating that it is focused on the ImageCFGen algorithm.

provide visual explanations of classifiers that are meant to provide interpretable justifications of their decisions by showing minimal changes to an image which change a classifier's decision.

A central goal of this research is to determine if causal generative models can produce more interpretable visual classifier explanations than the perturbation-based methods available from OmnixAI. A main contribution of this research, therefore, is to show that DeepSCM and ImageCFGen can be used to produce counterfactual explanations specific to a given classifier. To generate explanations using DeepSCM and ImageCFGen models, minimal changes are made to the human-interpretable attributes of an image in order to change the decision of a classifier. When searching for changes, either a loss function can be used to perform gradient descent on the attributes in question or model-agnostic interpolation between attribute values can be performed. The hypothesis behind this experiment is that searching in the attribute space (the space of low-dimensional causes of an image) rather than the pixel space (the space of all possible images) allows for the style of an image to be preserved while also producing realistic counterfactual data to explain a classifier. To test this hypothesis, a case study was performed on Morpho-MNIST to generate explanations for a classifier using the methods considered. The IM1 and IM2 metrics from Van Looveren and Klaise [15] as well as the oracle score metric from Hvilshøj et al. [16] were used to quantitatively measure the interpretability of generated explanations from the proposed methods and those from OmnixAI (see subsection 4.5.1). The proposed methods from this work outperform the counterfactual explainer from OmnixAI on all metrics considered, suggesting that the proposed methods using causal generative models provide more interpretable (i.e., less adversarial) explanations of classifiers than the considered perturbation-based methods.

1.3 Counterfactuals for Audio Data

The main audio dataset considered in this thesis is Audio-MNIST [17], which consists of recordings of human speech (speakers uttering the digits "zero" through "nine") with metadata concerning speakers age, biological sex, and other attributes related to the speaker's voice, such as accent. This work presents an assumed causal graph over the attributes of the dataset, where causal influences exist between attributes of

the speakers voice in order to build a structural causal model.

The second central goal of this research is to evaluate the ability of the DeepSCM and ImageCFGen algorithms to generate observational and counterfactual spectrogram data from audio datasets containing metadata with causal influences on the audio. This is done in two ways. The first method of evaluation is similar to the evaluation of digit counterfactuals from Dash et al. [5], in that an image classifier is used to measure the validity of generated counterfactuals for attributes being classified, with the main difference being the application to audio data rather than Morpho-MNIST. For instance, if an utterance is changed from a "three" to an "eight" via a counterfactual being performed, the classifier is used to ensure that the generated counterfactual is a valid "eight". The second method of evaluating audio counterfactuals is unique to this work and is specific to the Audio-MNIST dataset. The creation of a second method of evaluation is motivated by the difficulty of evaluating counterfactual data in a way that is not identical to an evaluation of interventional data. Specifically, when considering the setting of creating a digit utterance counterfactual (e.g., changing "three" to "eight"), any model which produces a valid eight will achieve a high score on a metric that simply uses a classifier of the digit being spoken, regardless of whether or not it has produced a counterfactual in a causally valid manner. This fact highlights the difficulty of evaluating causal models which produce counterfactuals. To measure the ability of a model to compute this counterfactual, it must not only measure that the digit attribute was properly changed, but also that the speakers voice was preserved. With this in mind, this thesis proposes using a subject (speaker) classifier to evaluate this form of counterfactual. To provide a baseline, this thesis uses this "subject agreement" metric to compare DeepSCM and ImageCFGen models with simpler generative models which do not compute counterfactuals.

The results in this thesis found that in the experiment involving consistency of measured attributes, the DeepSCM model provides superior counterfactual generation to ImageCFGen after standard training completes in most settings. However, when combined with a fine-tuning approach, the ImageCFGen model is able to produce comparable performance to DeepSCM. This observation is not carried over to the speaker classifier-based evaluation, however, where DeepSCM is the only model able

to significantly improve over simpler generative models in terms of speaker preservation when approximating speech counterfactuals. This leads to the conclusion that the VAE-based model of DeepSCM can be more easily trained on spectrogram data than GAN-based models to produce accurate speech counterfactuals.

A second dataset consisting of North American Right Whale (NARW) calls is also considered in this thesis to test the ability of ImageCFGen and DeepSCM models to produce whale call spectrograms. The causal graph of this dataset is simple, with the metadata of the audio indicating which type of call was recorded (upcall, gunshot call, or no call). Both DeepSCM and ImageCFGen models are trained on these datasets in order to produce observational speech and whale call data. To evaluate DeepSCM and ImageCFGen on this dataset, a whale call classifier is trained to evaluate data generated by the two models. Because the data does not lend itself to the abduction evaluation strategy of speaker classification as with Audio-MNIST, this evaluation does not measure an ability unique to counterfactual models. The models achieve high agreement with the whale call classifier in most cases for both observational and counterfactual data. However, a repetition of this experiment resulted in a very low agreement for the ImageCFGen model without fine-tuning, highlighting the instability of the GAN-based model and suggesting that the VAE-based model of DeepSCM may be more desirable and simpler to train in some settings.

1.4 Summary

In summary, this thesis makes two main contributions. The first is a method of explaining image classifiers using pretrained causal models, which produces realistic handwritten digit examples and achieves promising results on quantitative evaluation metrics. The second is the evaluation of existing deep causal model architectures on a human speech dataset. This includes using a speaker classifier to measure the ability of causal models to accurately perform abduction and preserve a speaker's voice during counterfactual generation, where DeepSCM is the only causal model which achieves a significantly higher performance than models which do not compute counterfactuals. Evaluations using classifiers for the attributes of the human speech data also suggest that the causal models considered in this work are able to produce spectrograms for audio with desired values of attributes (e.g., accent). Additionally,

results on whale call data suggest that the causal generative models can produce whale call spectrograms with features believable to a classifier based on call type.

Chapter 2

Background on Causal Systems

2.1 The Ladder of Causation

A structural causal model is a mathematical object capable of answering queries at all three rungs of the ladder of causation as introduced by Pearl in *The Book of Why* [18]. These three rungs are:

1. Association: concerned with answering questions such as "Is smoking associated with lung cancer?" without considering causation, e.g., by using tests of correlation.

2. Intervention: concerned with answering questions such as "If I quit smoking today, how is my probability of contracting lung cancer influenced?". Answering such questions accurately requires knowledge of the causal relation between variables, and can be thought of as allowing exogenous influences to vary as usual, while modifying one or more endogenous variables.

3. Counterfactual: the highest and arguably most complicated rung of causality. Counterfactual questions include those such as "Would the patient have lived longer if they had been convinced to quit smoking ten years earlier?". Counterfactual questions must account for the exogenous influences on observed variables, and as such require more powerful inference techniques than interventional questions.

Because an SCM is able to answer questions at all three rungs of the ladder of causality, it must represent not only the presence of causal relationships between variables, but also the functional relationships between causes and effects. Further, it must account for the random factors present in data in order to build density functions for observed variables (when possible) and allow for sampling from its various distributions.

2.2 Structural Causal Models

The latter two rungs of the ladder of causation can be described in the language of SCMs as presented by Scholkopf and Kugelgen. Definitions provided in this section are heavily inspired by those presented in their previous work [1].

Definition 1 *A structural causal model (SCM) for a set of n observed variables* $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^n$ *is a tuple* $\mathcal{M} = (\mathbf{F}, p_{\mathbf{U}})$*, where* $p_{\mathbf{U}}$ *is a distribution over n independent noise variables* $\mathbf{U} = \{\mathbf{U}_i\}_{i=1}^n$ *and* $\mathbf{F}$ *is a set of n structural equations of the form:*

$$\mathbf{X}_i := f_i(\mathbf{PA}_i, \mathbf{U}_i).$$

The symbol $\mathbf{PA}_i$ *represents the causal parents, or the causes of* $\mathbf{X}_i$*, and* f_i *is a function representing the causal relationship between* $\mathbf{X}_i$ *and its causes which accounts for random factors* $\mathbf{U}_i$*.*

Because the noise variables $\mathbf{U}_i$ are independent, they admit a factorization over the n independent distributions $p_{\mathbf{U}_i}$:

$$p_{\mathbf{U}}(\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n) = \prod_{i=1}^{n} p_{\mathbf{U}_i}(\mathbf{U}_i). \tag{2.1}$$

Further, because each $\mathbf{X}_i$ is fully determined by $\mathbf{U}_i$ when conditioning on $\mathbf{PA}_i$, the overall distribution of $\mathbf{X}$ admits a so-called causal factorization:

$$p(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n) = \prod_{i=1}^{n} p(\mathbf{X}_i | \mathbf{PA}_i). \tag{2.2}$$

This causal factorization implicitly defines a directed acyclic graph (DAG) describing the causal relationships between variables, with the vertex set representing the set of variables in $\mathbf{X}$ and an edge $\mathbf{X}_j \to \mathbf{X}_i$ whenever $\mathbf{X}_j \in \mathbf{PA}_i$.

2.2.1 Interventions

Consider an SCM $\mathcal{M}$ over variables A, B, C, with the causal DAG having edges $\{A \to B, A \to C, B \to C\}$ (see Figure 2.1a). It is relatively straightforward to show that:

$$p(C|B = b) = \mathbb{E}_{a \sim p(A|B=b)}[p(C|A = a, B = b)]. \tag{2.3}$$

However, this distribution does not amount to an intervention on B in the SCM $\mathcal{M}$, which comes from the following definition.

Figure 2.1: The graph representing the causal structure of the SCM considered in section 2.2.1, with three variables A, B, and C with the causal relationships $A \to B, A \to C$, and $B \to C$. The graph is shown both (a) before an intervention is performed and (b) after an intervention $do(B = b)$ has removed the causal relationship $A \to B$.

Definition 2 *An intervention $do(\mathbf{X}_i = \mathbf{x}_i)$ in an SCM $\mathcal{M} = (\mathbf{F}, p_{\mathbf{U}})$ replaces the ith structural assignment in $\mathbf{F}$ with the assignment $\mathbf{X}_i := \mathbf{x}_i$, resulting in a new set of structural assignments $\mathbf{F}'$ and a new SCM $\mathcal{M}^{do(\mathbf{X}_i=\mathbf{x}_i)} = (\mathbf{F}', p_{\mathbf{U}})$.*

Using this definition, A and B become independent in our example SCM when we perform the intervention $do(B = b)$. Thus:

$$p(C|do(B = b)) = \mathbb{E}_{a\sim p(A)}[p(C|A = a, B = b)]. \tag{2.4}$$

This shows that, in general, intervened distributions are not equal to their standard conditional counterparts, due to the modification of the causal graph that takes place when an intervention is performed. A similar example can be found from Scholkopf and Kugelgen [1]. In reference to Equation 2.2, the term $p(\mathbf{X}_i|\mathbf{PA}_i)$ in Equation 2.2 is replaced by a delta function $\delta(\mathbf{X}_i - \mathbf{x}_i)$ when computing an intervention. Equivalently, some authors [1] may simply consider the distribution $p(\mathbf{X}_{-i}|do(\mathbf{X}_i = \mathbf{x}_i))$ where $\mathbf{X}_{-i}$ is the set of variables in $\mathbf{X}$ excluding $\mathbf{X}_i$, removing the intervened variable from consideration when computing Equation 2.2.

2.2.2 Counterfactuals

Interventions modify the causal structure of an SCM, but do not change the noise distribution $p_{\mathbf{U}}$ of the SCM. Counterfactuals, on the other hand, require updating $p_{\mathbf{U}}$ based on observed evidence in order to account for the conditions (unobserved

influences) under which the evidence was created. The process of updating $p_{\mathbf{U}}$ is commonly referred to as *abduction* [5, 4, 1], and can be thought of as the procedure by which the values of $\mathbf{U}$ which could have led to a given observation $\mathbf{X} = \mathbf{x}$ are recovered.

Definition 3 *A counterfactual SCM for a given observation* $\mathbf{X} = \mathbf{x}$ *is derived from an SCM* $\mathcal{M} = (\mathbf{F}, p_{\mathbf{U}})$ *by replacing* $p_{\mathbf{U}}$ *with the conditional distribution* $p_{U|\mathbf{X}=\mathbf{x}}$*, the* i*th component of which is the conditional* $p(\mathbf{U}_i|\mathbf{X} = \mathbf{x})$*. This counterfactual SCM is denoted* $\mathcal{M}^{\mathbf{X}=\mathbf{x}} = (\mathbf{F}, p_{\mathbf{U}|\mathbf{X}=\mathbf{x}})$*.*

Definition 4 *A counterfactual can be computed in an SCM* $\mathcal{M}$ *in two main steps:*

1. *An observation* $\mathbf{X} = \mathbf{x}$ *is used to construct a counterfactual SCM* $\mathcal{M}^{\mathbf{X}=\mathbf{x}}$ *according to Definition 3.*

2. *An intervention is performed in the counterfactual SCM according to Definition 2.*

Definition 4 is based on previously described steps of counterfactuals [1], though it is simplified due to building on Definitions 2 and 3. Abduction is the most crucial component of computing a counterfactual, and what separates SCMs from other types of models, as it requires knowledge of the inner workings of the data generation process. For example, if $\mathbf{X}_i := f_i(\mathbf{PA}_i, \mathbf{U}_i)$ is an assignment of $\mathcal{M}$, and unique f_i^{-1} exists (at least locally) satisfying the inverse function condition:

$$f_i(\mathbf{pa}_i, f_i^{-1}(\mathbf{x}_i, \mathbf{pa}_i)) = \mathbf{x}_i, \tag{2.5}$$

then the ith component of $p_{\mathbf{U}|\mathbf{X}=\mathbf{x}}$ satisfies:

$$p(\mathbf{U}_i|\mathbf{X}_i = \mathbf{x}_i, \mathbf{PA}_i = \mathbf{pa}_i) = \delta(\mathbf{U}_i - f_i^{-1}(\mathbf{x}_i, \mathbf{pa}_i)). \tag{2.6}$$

In the case of discrete distributions, where the logits of the distribution are defined as a function of $\mathbf{PA}_i$, the distribution $p_{\mathbf{U}|\mathbf{X}=\mathbf{x}}$ can be sampled from using a Gumbel-max parameterization [19, 20, 4] and the counterfactual can be approximated through Monte-Carlo style sampling.

2.3 Normalizing Flows

When approximating the distribution $p(\mathbf{X})$ of an SCM, it is common to learn the transformation $\mathbf{X}_i := f_i(\mathbf{PA}_i, \mathbf{U}_i)$ using a trainable model f_{θ_i} and a fixed prior $p(\mathbf{U}_i)$ [4]. This strategy allows for drawing samples from $p(\mathbf{X})$ by first drawing samples from $p_{\mathbf{U}}$ and transforming noise variables into observations through the learned functions f_{θ_i}. However, if the parameters θ_i are to be learned through maximum likelihood, the density $p_{\theta_i}(\mathbf{X}_i|\mathbf{PA}_i)$ of the transformed variable $\mathbf{X}_i$ is required. The method of normalizing flows allow the necessary densities to be computed exactly for observed data in a tractable manner [8]. The popularity of normalizing flow models is credited by Kobyzev et al. [8] to Rezende and Mohamed [6] and Dinh et al. [21], though they have been present in the literature for at least four years prior to these works [22]. More recently, these models have been extended to conditional distributions in high dimensions [23, 24]. Conditional normalizing flows have also found a home as the crux of the "invertible, explicit" strategy of deep structural causal models (DeepSCM) [4].

Normalizing flow models allow the modelling of a data generation process of a random variable $\mathbf{Y}$ as an invertible function f. A formal definition is provided below.

Definition 5 *A normalizing flow model of a random variable* $\mathbf{Y}$ *consists of a prior* $p_{\mathbf{U}}$ *over a random variable* $\mathbf{U}$ *and an invertible function* f *with inverse* f^{-1}*. The data generation process of* $\mathbf{Y}$ *is modelled as:*

$$\mathbf{U} \sim p_{\mathbf{U}}, \tag{2.7}$$

$$\mathbf{Y} := f(\mathbf{U}). \tag{2.8}$$

Using the change of variables $\mathbf{U} = f^{-1}(\mathbf{Y})$*, the density of* $\mathbf{Y}$ *can be calculated as:*

$$p(\mathbf{Y}) = p_{\mathbf{U}}(f^{-1}(\mathbf{Y}))|\det \mathbf{D}f^{-1}(\mathbf{Y})|, \tag{2.9}$$

where $\mathbf{D}f^{-1}$ *denotes the Jacobian of* f^{-1}*.*

If f in Definition 5 is composed of multiple simple transformations $\phi_1, \phi_2, \ldots, \phi_N$, then the determinant of the Jacobian of f can be represented as a product of the determinants of the Jacobians of the individual transformations ϕ_i [21], allowing for the combination of multiple invertible transformations to be used. The strategy for

computing conditional normalizing flows is essentially the same as Definition 5, with the exception that f becomes a function with two arguments and f^{-1} returns only the value of one argument. In SCMs, these two arguments are $\mathbf{PA}_i$ and $\mathbf{U}_i$, and the function must be invertible in $\mathbf{U}_i$ to use normalizing flows.

2.4 Counterfactual Approximation

As seen in section 2.3, the distribution $p(\mathbf{X}_i|\mathbf{PA}_i)$ of a variable $\mathbf{X}_i$ of an SCM can be computed using normalizing flows, provided that f_i is invertible in $\mathbf{U}_i$ and $p_{\mathbf{U}_i}$ is known. However, this strategy can only be used to learn f_i through approximate maximum likelihood when f_i can be defined explicitly, which is typically only possible when $\mathbf{X}_i$ is a continuous variable. When $\mathbf{X}_i$ is categorical, the Gumbel-max parameterization trick [20] can be used when the logits $\boldsymbol{\lambda} = g(\mathbf{PA}_i)$ of the used softmax distribution are a learned (possibly non-invertible) function g of the causal parents $\mathbf{PA}_i$.

When high-dimensional unstructured data is used, such as images or audio, explicitly learning the densities $p(\mathbf{X}_i|\mathbf{PA}_i)$ of random variables in the SCM is often intractable using normalizing flows, motivating the use of other methods. To the best of my knowledge, there have been two main works on creating models for tractable counterfactual inference for image data: DeepSCM from Pawlowski et al. [4], and ImageCFGen from Dash et al. [5]. While the DeepSCM paper is more high-level, and outlines three strategies for learning SCM components using deep mechanisms (one of which is considered theoretically equivalent to ImageCFGen), the two studies are in practice different enough to be considered separately as they use different deep learning techniques (Variational Autoencoders vs. Bidirectional GANs). When discussing such models, it is helpful to recall that Pearl defines three abilities any model must have in order to perform counterfactual inference [25]:

1. Abduction: the process of computing $p_{\mathbf{U}|\mathbf{X}=\mathbf{x}}$.

2. Action: the process of computing an intervention in the resulting counterfactual SCM.

3. Prediction: the process of performing probabilistic inference in the SCM resulting from the previous two steps.

Because the action and prediction steps are straightforward when given the definition of all transformations f_i and the result of abduction, the abduction process is the focus of the discussion of DeepSCM and ImageCFGen in chapter 3.

Chapter 3

Related Work

3.1 DeepSCM

Work from Pawlowski et al. [4] is the first work to use deep mechanisms to learn high-dimensional flows for counterfactual inference. Referred to as DeepSCM, the work proposes three ways to model the data generation process of an SCM while allowing for counterfactual inference.

The first type of mechanism proposed by DeepSCM is referred to as "invertible, explicit" mechanisms. This is equivalent to a learned conditional normalizing flows strategy (see section 2.3). For each observed variable $\mathbf{X}_i$, a base distribution $p(\mathbf{U}_i)$ is chosen along with a (typically trainable) transformation $f_i(\mathbf{PA}_i, \mathbf{U}_i)$ invertible in its second argument. Due to the invertibility of f_i, the abduction step can be performed simply by computing the abduction noise distribution as a point mass:

$$p_{\mathbf{U}_i|\mathbf{X}=\mathbf{x}}(\mathbf{U}_i) = \delta(\mathbf{U}_i - f_i^{-1}(\mathbf{PA}_i, \mathbf{X}_i)). \tag{3.1}$$

The second mechanism proposed for counterfactual inference by Pawlowski et al. is referred to as the "amortized, explicit" strategy. In this case, the noise variables $\mathbf{U}_i$ are decomposed into two components as $\mathbf{U}_i = (\mathbf{T}_i, \mathbf{Z}_i)$, and two transformations H_i (invertible) and G_i (non-invertible) are chosen, to make the assignment of $\mathbf{X}_i$ represented as:

$$\mathbf{X}_i := H_i(\mathbf{PA}_i, G_i(\mathbf{Z}_i, \mathbf{PA}_i), \mathbf{T}_i). \tag{3.2}$$

In practice, this can be interpreted as the stochastic decoder of a variational autoencoder [26], such as those using the well-known reparameterization trick of the variational Bayes [27]. More specifically, G_i could output the mean of the decoder's output distribution, and H_i could transform the zero-mean Gaussian random variable $\mathbf{T}_i$ into a Gaussian random variable centered at $G_i(\mathbf{Z}_i, \mathbf{PA}_i)$ with fixed variance. If the latent distributions $Q(\mathbf{Z}_i|\mathbf{X}_i, \mathbf{PA}_i)$ and $p(\mathbf{Z}_i)$ are also known, a variational lower

bound can be constructed [4]:

$$\begin{aligned}\log p(\mathbf{X}_i|\mathbf{PA}_i) \geq\ & \mathbb{E}_{Q(\mathbf{Z}_i|\mathbf{X}_i,\mathbf{PA}_i)}\left[\log p(\mathbf{X}_i|\mathbf{PA}_i,\mathbf{Z}_i)\right] \\ & - D_{KL}\left[Q(\mathbf{Z}_i|\mathbf{X}_i,\mathbf{PA}_i)||p(\mathbf{Z}_i)\right] \end{aligned} \tag{3.3}$$

where D_{KL} is the Kullback–Leibler divergence, which can be computed analytically for independent Gaussians. As is usually done with variational autoencoders, the distribution $Q(\mathbf{Z}_i|\mathbf{X}_i,\mathbf{PA}_i)$ can be approximated by a stochastic encoder distribution $Q(\mathbf{Z}_i|E_i(\mathbf{X}_i,\mathbf{PA}_i))$ with parameters specified by a learned function E_i. The encoder can be jointly trained with the decoder to maximize the lower bound in Equation 3.3. While exact abduction cannot be performed to compute $p(\mathbf{X}_i|\mathbf{PA}_i)$ due to G_i not being invertible, samples from the encoder can be used to compute $p(\mathbf{X}_i|\mathbf{PA}_i,\mathbf{Z}_i)$ and hence the variational lower bound. This is because $\mathbf{T}_i$ can be uncovered from $\mathbf{X}_i,\mathbf{PA}_i$, and $\mathbf{Z}_i$ due to the invertibility of H_i. Because the distribution of $\mathbf{Z}_i$ is learned through the encoder, and $\mathbf{T}_i$ is found directly, the noise distribution can be approximated as:

$$p_{\mathbf{U}_i|\mathbf{X}=\mathbf{x}}(\mathbf{U}_i) \approx Q(\mathbf{Z}_i|E_i(\mathbf{X}_i,\mathbf{PA}_i))p(\mathbf{T}_i|\mathbf{X}_i,\mathbf{PA}_i,\mathbf{Z}_i). \tag{3.4}$$

That is, samples of $\mathbf{U}_i$ are found by sampling $\mathbf{Z}_i$ from the encoder and then by computing $\mathbf{T}_i$ directly using G_i and H_i. Because this distribution is itself not a point mass, counterfactual prediction is often performed by Monte Carlo sampling [4]. A diagram of the VAE architecture used to train a "amortized, explicit" DeepSCM model using a variational autoencoder is shown in Figure 3.1.

The third and final SCM mechanism proposed by Pawlowski et al. is the "amortized, implicit" mechanism, which does not rely on approximate maximum likelihood training. Because this method was first implemented as ImageCFGen, it is instead discussed in section 3.3. Specific implementations of the DeepSCM mechanisms, including model architectures, are discussed in detail in chapter 4.

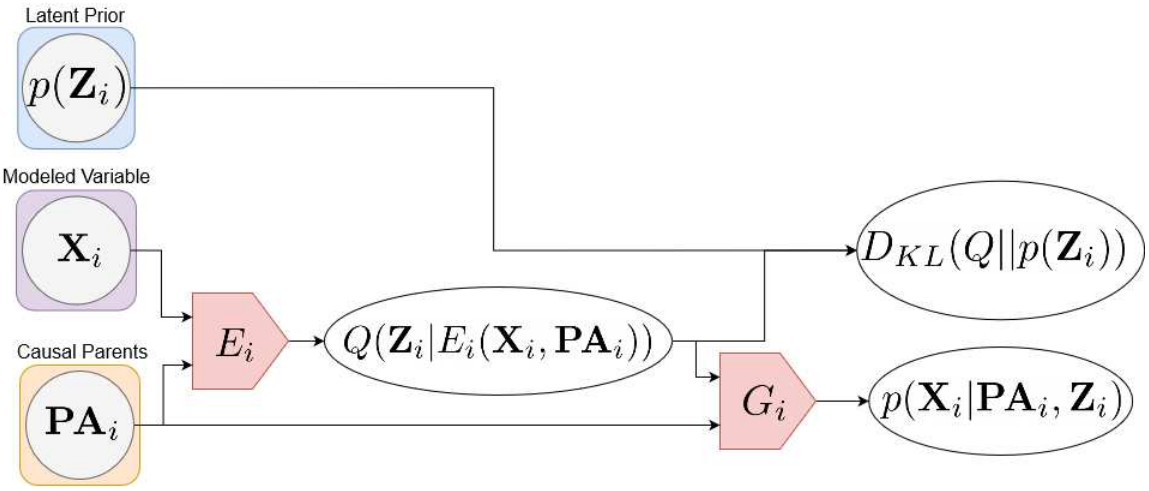


Figure 3.1: The variational autoencoder architecture used in the DeepSCM strategy of counterfactual approximation. The encoder E_i models a latent distribution Q conditioned on observed variables $\mathbf{X}_i$ and their causal parents $\mathbf{PA}_i$. A decoder/generator G_i is trained to reconstruct an image when provided with the causal parents $\mathbf{PA}_i$ and latent features $\mathbf{Z}_i$.

3.2 Generative Adversarial Networks

Before discussing the bidirectional generative models used in the ImageCFGen method of counterfactuals (see section 3.3), the general theory of generative adversarial networks (GANs) [28] is introduced. GANs aim to solve the generative modelling problem, in which a generator G attempts to produce samples similar to those observed from an observed distribution $q(\mathbf{x})$. Specifically, a prior distribution $p(\mathbf{z})$ is chosen, and images $G(\mathbf{z})$ are made to approximate those from $q(\mathbf{x})$. To solve this task, the generator engages in a zero-sum game with a discriminator function D, which aims to distinguish real samples from $q(\mathbf{x})$ from generated samples $G(\mathbf{z})$ (where $\mathbf{z} \sim p(\mathbf{z})$). This approach allows the model to learn the distribution $q(\mathbf{x})$ without explicitly representing the density function. While there are multiple possible objectives to jointly train G and D, one common approach is to use a binary cross-entropy style loss, treating D as a binary classifier. In this setting, the objective of the optimization process is to find:

$$\min_G \max_D V(G, D) = \mathbb{E}_{q(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{3.5}$$

Such an optimization strategy is referred to as the *minmax* strategy, as the same objective function is used for the generator and discriminator, with the generator

performing minimization and the discriminator performing maximization. This process is equivalent to training the generator using the negative of the objective used for the discriminator. Promising results have also been found when the generator instead maximizes $\log D(G(\mathbf{z}))$, as this objective function leads to less saturated gradients [28].

Once trained, new data samples $\tilde{\mathbf{x}}$ can be computed by sampling $\mathbf{z} \sim p(\mathbf{z})$ and setting $\tilde{\mathbf{x}} = G(\mathbf{z})$. Note that this takes the form of a zero-sum game: any increase in the objective being maximized by D corresponds to the same increase in the objective being minimized by G. Bidirectional and Wasserstein GANs, variants on this architecture, are discussed in section 3.3 and section 3.4 respectively.

3.3 ImageCFGen

Dash et al. [5] have implemented the amortized, implicit SCM mechanism proposed by Pawlowski et al. using Adversarially Learned Inference (ALI), also known as Bidirectional generative adversarial networks (BiGAN) [29, 7]. As with a typical generative adversarial network, in a BiGAN model a generator model G_i (which models the variable $\mathbf{X}_i$) and a discriminator model D engage in a zero-sum game in which the discriminator aims to tell generated data from observational data and the generator aims to trick the discriminator. In addition to these two models, an encoder model E_i aims to generate encodings of observational data which fool the discriminator into believing they are from a predefined prior $p(\mathbf{Z}_i)$. When the model is also conditioned on the attributes (causal parents) $\mathbf{PA}_i$ of a variable $\mathbf{X}_i$, the discriminator is meant to discriminate tuples $(\mathbf{X}_i, E_i(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i)$ of real images, real encodings, and attributes from tuples $(G_i(\mathbf{Z}_i, \mathbf{PA}_i), \mathbf{Z}_i, \mathbf{PA}_i)$ of generated images, samples from $p(\mathbf{Z}_i)$, and attributes. Denoting $q(\mathbf{X}_i, \mathbf{PA}_i)$ as the observational distribution of images and attributes, the solution of the conditional BiGAN zero-sum game can be described as:

$$\min_{G, E_i} \max_{D} \; \mathbb{E}_{q(\mathbf{X}_i, \mathbf{PA}_i)} \left[\log D(\mathbf{X}_i, E_i(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i)\right] + \mathbb{E}_{p(\mathbf{Z}_i)p(\mathbf{PA}_i)} \left[\log(1 - D(G_i(\mathbf{Z}_i, \mathbf{PA}_i), \mathbf{Z}_i, \mathbf{PA}_i))\right]. \tag{3.6}$$

Note that in this way, D acts as a binary classifier, typically with a single-neuron sigmoid output, determining whether the image and latent data comes from the empirical distribution and encoder or is generated from the latent prior. Contrast this with the original GAN discussed in section 3.2, which only discriminates based on image data and does not define E_i. Thus, only the BiGAN can create latent representations of observed images, a process crucial to counterfactual approximation.

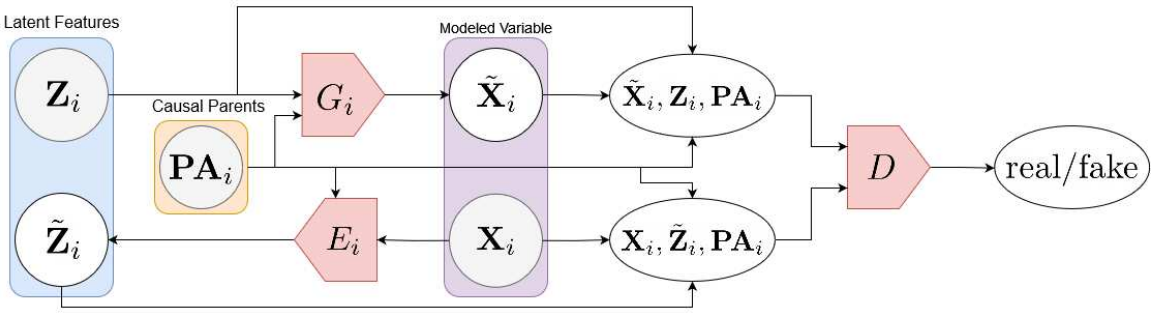


Figure 3.2: The conditional BiGAN architecture used in the ImageCFGen method of counterfactual approximation. The encoder E_i and generator G_i are jointly trained to fool a discriminator D from distinguishing generated images and real latent features from real images and encoded latent features.

Because the encoder E_i of a BiGAN is deterministic, approximate abduction can be performed by finding $\tilde{\mathbf{Z}}_i = E_i(\mathbf{X}_i, \mathbf{PA}_i)$. Hence, computing a counterfactual given evidence $\mathbf{X}_i, \mathbf{PA}_i$ and counterfactual attributes $\mathbf{PA}'_i$ recovered from other observed variables in the SCM is defined as:

$$\mathbf{X}'_i = G(E(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}'_i). \tag{3.7}$$

The appendices of the paper from Dash et al. [5] provide an analysis comparing this method of counterfactual generation to the standard form of an SCM [1].

In addition to training the ImageCFGen model to solve the optimization problem specified by Equation 3.6, Dash et al. recommends a finetuning process for the encoder in order to improve the quality of counterfactuals. The finetuning process jointly minimizes two losses, the image reconstruction error and the latent space loss. These errors are defined as:

$$\mathcal{L}_{\mathbf{x}} = \mathbb{E}_{q(\mathbf{X}_i, \mathbf{PA}_i)} ||\mathbf{X}_i - G_i(E_i(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i)||^2, \tag{3.8}$$

$$\mathcal{L}_{\mathbf{z}} = \mathbb{E}_{p(\mathbf{Z}_i)} ||\mathbf{Z}_i - E_i(\mathbf{X}_i, \mathbf{PA}_i)||^2. \tag{3.9}$$

The loss $\mathcal{L}_{\mathbf{x}}$ can be estimated directly from the observational distribution of images and attributes. This thesis shows that when the prior $p(\mathbf{Z}_i)$ has independent components, zero mean, and finite variance, $\mathcal{L}_{\mathbf{z}}$ has a closed form such that the fine-tuning process can be described as finding:

$$\min_{E_i} \mathbb{E}_{q(\mathbf{X}_i, \mathbf{PA}_i)} \left[||\mathbf{X}_i - G_i(E_i(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i)||^2 + ||E_i(\mathbf{X}_i, \mathbf{PA}_i)||^2 \right]. \tag{3.10}$$

A proof of this is provided in Appendix A.

3.3.1 ImageCFGen Feature Importance

Consider a standard image flow setting as before, where a set of attributes $\mathbf{PA}_i$ with a causal structure defined by an SCM $\mathcal{M}$ are causes of images $\mathbf{X}_i$. Further, suppose $f(\mathbf{X}_i)$ is a model providing classification scores for observations. Finally, suppose we observe an image-attribute pair $\mathbf{X}_i, \mathbf{PA}_i$, and we want to measure the (local) importance of a binary attribute $\mathbf{a}_j \in \mathbf{PA}_i$ with values v_0 and v_1. Denote $\mathbf{PA}_i^{\mathbf{a}_j \leftarrow v | \mathbf{PA}_i}$ as the counterfactual values of $\mathbf{PA}_i$ found by computing the intervention $do(\mathbf{a}_j = v)$ in the counterfactual SCM $\mathcal{M}^{\mathbf{PA}_i}$. Dash et al. [5] define an importance metric for binary attributes as follows:

$$\text{importance}(j; \mathbf{X}_i, \mathbf{PA}_i) = f(\mathbf{X}_i^{\mathbf{a}_j \leftarrow v_1 | \mathbf{PA}_i}) - f(\mathbf{X}_i^{\mathbf{a}_j \leftarrow v_0 | \mathbf{PA}_i}) \tag{3.11}$$

where $\mathbf{X}_i^{\mathbf{a}_j \leftarrow v | \mathbf{PA}_i}$ is computed by the BiGAN as:

$$\mathbf{X}_i^{\mathbf{a}_j \leftarrow v | \mathbf{PA}_i} = G(E(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i^{\mathbf{a}_j \leftarrow v | \mathbf{PA}_i}). \tag{3.12}$$

In the case of a stochastic encoder and decoder (as with deepSCM), $\mathbf{X}_i^{\mathbf{a}_j \leftarrow v | \mathbf{PA}_i}$ can be computed through Monte Carlo sampling. Dash et al. successfully used this importance metric to measure the impact of binary attributes on an attractiveness classifier trained on the CelebA dataset. For instance, when the attribute $\mathbf{a}_j$ corresponds to baldness, and $\text{importance}(j; \mathbf{X}_i, \mathbf{PA}_i) < 0$, this indicates that the presence of baldness causes a decrease in attractiveness.

3.4 SpecGAN

Generative modelling of audio using GANs has been previously proposed by Donahue et al. [30] in an algorithm referred to as WaveGAN. Standard GAN models employ

a generator G and a discriminator D in a zero-sum game similar to Equation 3.6. While the authors generate raw audio directly, they also propose a variant of their model which generates spectrograms, called SpecGAN. Generated spectrograms can be approximately inverted using Griffin-Lim algorithms [31] to recover audio. Further, WaveGAN and SpecGAN models are trained using a Wasserstein GAN objective with a gradient penalty [32]. The Wasserstein GAN objective is defined as:

$$\min_G \max_D \mathbb{E}_{q(\mathbf{x})}[D(\mathbf{x})] - \mathbb{E}_{p(\mathbf{z})}[D(G(\mathbf{z}))] \tag{3.13}$$

where G and D are the generator and discriminator functions of the GAN, $q(\mathbf{x})$ is the audio data distribution and $p(\mathbf{z})$ is a chosen prior. In this method, rather than classifying samples as real or fake, the discriminator aims to calculate the Wasserstein distance between the distribution $q(\mathbf{x})$ and that of $G(\mathbf{z})$. The gradient penalty proposed by Gulrajani et al. [32] aims to keep the norm of the gradient of D close to 1 by adding a regularization term to Equation 3.13, making a costly second-order derivative computation necessary for training the discriminator.

Chapter 4

Methods and Experimental Setup

4.1 Audio Processing

Audio data is often stored as a signal in formats such as the WAV file format [33], which represents the audio as a univariate time series. However, audio can be represented in ways other than this time-domain representation, such as in the frequency domain through the use of spectrograms. Previous studies have been conducted on performing audio synthesis using GAN models with both WAV and spectrograms [30, 34]. One recent study found that, among others, the short-time Fourier transform (STFT) provided strong generation results [35]. A benefit of using the STFT is that it can be quickly approximately inverted using a Griffin-Lim algorithm [31] to produce WAV representations of generated audio. Given a discrete signal $f(i)$ defined at integer points $1 \leq i \leq T$, the STFT of f is defined as a function of frequency ω as it changes over time:

$$\hat{f}(\omega, m) = \sum_{k=0}^{\ell_w} w(k) f(m\ell_h + m) \exp\left(-j\frac{2\pi\omega m}{\ell_w}\right) \tag{4.1}$$

where w (called the *window*) is a vector of length ℓ_w, ℓ_h (called the *hop length*) is a positive integer, and j is the imaginary unit ($j^2 = -1$). A common choice of w is the Hanning window, defined as:

$$w(k) = \sin^2\left(\frac{\pi k}{\ell_w - 1}\right). \tag{4.2}$$

Note that $\hat{f}$ is a complex-valued function and is defined for continuous ω. One can choose a discrete grid of ω to use, and convert the STFT to a spectrogram by using the square of its magnitude (called a "power" spectrogram):

$$\text{spectrogram}\{\hat{f}\}(\omega, m) = \text{Re}(\hat{f}(\omega, m))^2 + \text{Im}(\hat{f}(\omega, m))^2. \tag{4.3}$$

Alternatively, the magnitude itself can be used by taking the square-root of the above equation as in work by Nistal, Lattner, and Richard [35]. In the experiments performed in this work, Equation 4.3 is used to compute spectrograms, with ℓ_w and ℓ_h set based on the dataset in order to produce images of a desired size. Spectrograms are log-transformed in order to produce more similar scales among frequency bins. The same data scaling technique used by Donahue et al. [30] is used, where spectrogram features are clipped to three standard deviations to produce a bounded feature space and then are scaled linearly to $[-1, 1]$.

4.2 Passing Attributes To Models

There are two methods used in this work to pass a group of attributes from an attribute SCM to a deep CNN model, depending on whether a given attribute is continuous or categorical (discrete). For continuous attributes, it is assumed that the attribute values are bounded between -1 and 1, and the raw attribute may be passed to the model directly by adding a constant channel to the models input. For example, if a 28x28 image is expected and there are k continuous attributes available to the model, the input image conditioned on the attributes can take the form of a

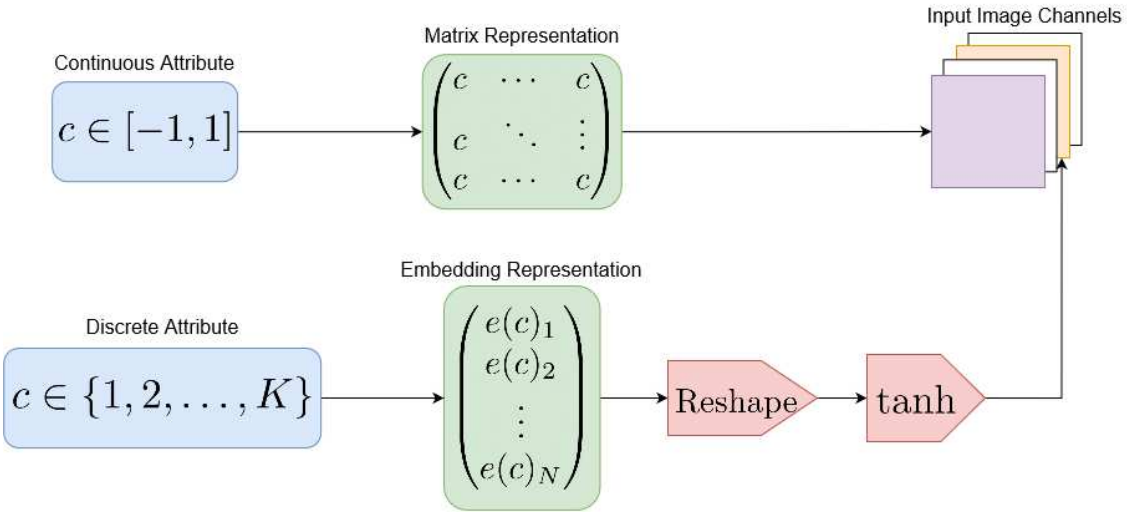


Figure 4.1: The strategy used in this work to pass attributes of an SCM to a CNN model. Continuous attributes form constant image channels in the input, while categorical attributes are transformed using a learned embedding $e(\cdot)$, reshaped, and upsampled. The learned embedding maps each value of the categorical attribute to a vector of fixed size.

$(k+1) \times 28 \times 28$ tensor, where k of the input channels correspond to single-valued 28x28 images having the same value as the continuous attributes. For categorical attributes taking integer values $1, 2, \ldots, K$, learned embedding vectors $e(c) \in \mathbb{R}^N$ can be used, where $c \in \{1, 2, \ldots, K\}$ is the value of the attribute and $e(\cdot)$ is the embedding lookup. The dimension N of the embedding is chosen to be a factor of the image size, and the image is upscaled by the result of dividing the image size by N in order to form a channel of the input image. For example, if a categorical attribute has K possible values and the input size of the original image is 28x28, we could take an embedding size of $N = 28$ and perform an upsampling of the embedding to create a new 28x28 channel of the input. A hyperbolic tangent function is applied to these upsampled embeddings to ensure features are constrained to $[-1, 1]$.

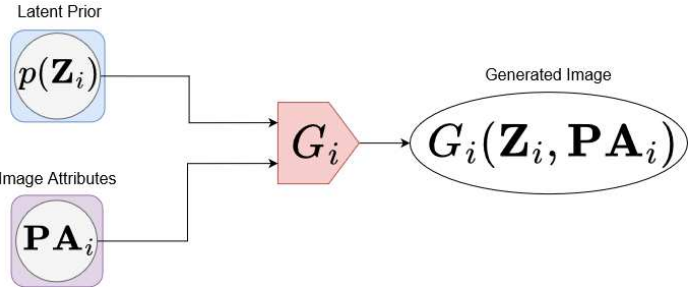
Note that the method of attribute passing shown in Figure 4.1 is only required for the encoder model of DeepSCM and the encoder and discriminator models of ImageCFGen (see section 3.1 and section 3.3). This is because generator/decoder models can simply concatenate attribute representations (i.e., embeddings and continuous attribute values) with latent image representations in order to produce image data.

4.3 Datasets

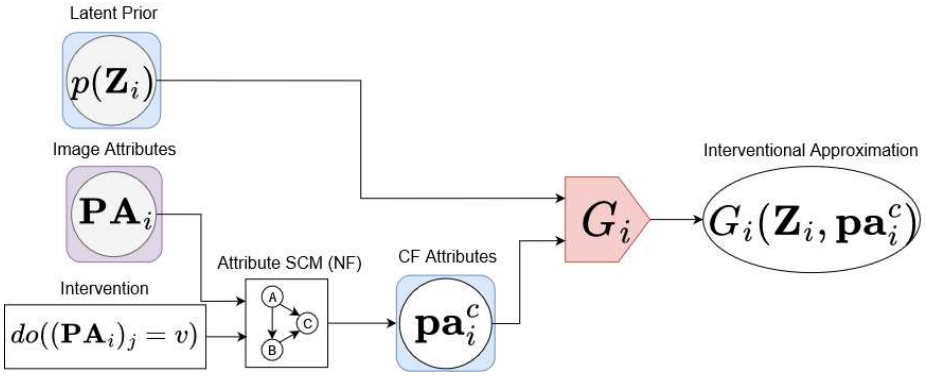
The following subsections describe the datasets used in this work, as well as the neural network architectures used to model the data.

In all datasets, transformations of random variables are required to model variables of the considered SCM other than the image or audio data. All transformations were implemented using the Pyro probabilistic programming framework [11], as they were in the work on DeepSCM by Pawlowski et al. [4]. Any learned parameters of these transformations were found through gradient ascent by maximizing the log likelihood of observed samples, which were computed using the method of normalizing flows (see section 2.3).

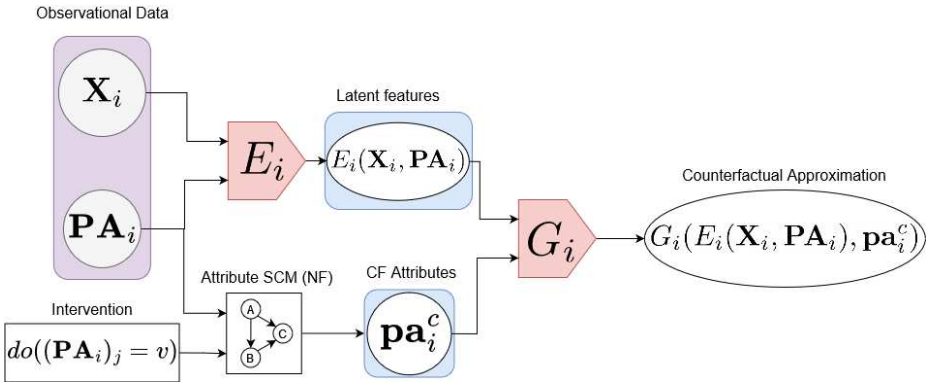
The encoder architectures of both BiGAN and VAE models trained in the following experiments are nearly identical, with the one key difference being that the encoder of a VAE is stochastic. Because the VAE encoder is stochastic, the features produced by the encoder architectures described in the following subsections are passed through



(a) An observational generative model, where image attributes and samples from a latent prior $p(\mathbf{Z}_i)$ are passed to a conditional generator G_i (as with a conditional GAN). Such models are common in modern machine learning.



(b) The type of generative model that is considered interventional in this work. Modifications to causes $\mathbf{PA}_i$ of a variable can be performed in a causally correct way by computing a counterfactual in the attribute SCM (a normalizing flows model), where the shown intervention changes the jth causal parent $(\mathbf{PA}_i)_j$. This is referred to as the interventional setting as no abduction is performed using the generative model itself, so sampling from the latent prior $p(\mathbf{Z}_i)$ is still necessary. This is the type of model we use to compare with counterfactual models using the Audio-MNIST subject classifier metric from subsection 4.4.2.



(c) A counterfactual model such as ImageCFGen, which follows the same method as the interventional model to uncover counterfactual attributes $\mathbf{pa}_i^c$, but also performs abduction using an encoder E_i to uncover latent features manually and compute a causally accurate image counterfactual.

Figure 4.2: Diagrams illustrating and contrasting the data generation process of standard generative models and causal models.

a LeakyReLU activation and two CNN layers, each with a filter size of 1, to produce the mean and log variance of a Gaussian distribution, as is often seeing when training VAEs [6].

The discriminator models used to train the BiGANs in this work always have three components: an image and attribute processing module $D_{\mathbf{x}}$, a latent vector processing module $D_{\mathbf{z}}$ producing feature vectors, and a downstream discriminator module $D_{\mathbf{x},\mathbf{z}}$ producing discriminator scores from the given features. Formally, we can write using the notation used in Figure 3.2:

$$D(\mathbf{X}_i, \mathbf{Z}_i, \mathbf{PA}_i) = D_{\mathbf{x},\mathbf{z}}(D_{\mathbf{x}}(\mathbf{X}_i, \mathbf{PA}_i), D_{\mathbf{z}}(\mathbf{Z}_i)) \tag{4.4}$$

where $\mathbf{X}_i$ is the observed variable to be modelled, $\mathbf{PA}_i$ are the causal parents of $\mathbf{X}_i$, and $\mathbf{Z}_i$ is a latent variable accounting for the unobserved influences of $\mathbf{X}_i$.

All neural network models in this work were implemented in Pytorch [10]. When describing the layers of neural network models (e.g. in Table 4.1), notation consistent with the Pytorch library is used when possible. Specifically, the following notation is used to describe neural network layers:

- Conv2D(f_i, f_o, k, s, p, a) refers to a 2D convolutional layer with the respective number of input and output filters (f_i, f_o), kernel size (k), stride (s), input padding (p), and activation function (a).
- Conv2DT(f_i, f_o, k, s, p, a) refers to a transposed convolutional layer with the respective number of input and output filters (f_i, f_o), kernel size (k), stride (s), padding (p), and activation function (a).
- Dropout(q) refers to a layer which sets input nodes to zero with probability q.
- Linear(n, m, a) denotes a fully-connected network layer with n input neurons, m output neurons, and activation function a.
- Reshape($\cdot$) represents an operation which reshapes the input tensor to the provided shape (batch dimensions ignored).

All models for each dataset are trained for 500 epochs using an Adam optimizer [36]. The hyperparameters β_1 and β_2 used for Adam, as well as learning rates

and batch size, are reported for each dataset in their respective subsections. When fine-tuning ImageCFGen models, 20 epochs are used to minimize Equation 3.10 with a learning rate of 10^{-5} in all cases. The default hyperparameters of the Adam optimizer from Pytorch are used during fine-tuning.

Curves showing the convergence of models trained while carrying out the experiments described in this chapter are provided in Appendix C.

4.3.1 Morpho-MNIST

The Morpho-MNIST dataset [9] presented by de Castro et al. adds additional causal aspects to the well-known MNIST handwritten digits dataset [37]. Specifically, attributes of line thickness and image intensity are added to the original handwritten digits of MNIST. Because these attributes are added to the digits of MNIST using image processing techniques, scientists have complete control over the data generation process of Morpho-MNIST data and hence have full knowledge of the SCM used to generate Morpho-MNIST data. Pawlowski et al. [4] were the first to create learned SCMs from the data generated using Morpho-MNIST techniques, and propose a causal graph involving the following attributes:

1. Thickness, denoted here as t_m.

2. Intensity, denoted here as i_m.

3. Digit image output, denoted here as $\mathbf{o}_m$.

In the SCM proposed by Pawlowski et al., and the one used in this work, all attributes cause the image output $\mathbf{o}_m$, and thickness t_m causes intensity i_m. However, in this work an additional attribute of image slant (denoted s_m, measured in radians) is applied to the handwritten digits, as was proposed by Dash et al. [5]. The causal graph for the Morpho-MNIST SCM is shown in Figure 4.3, and examples of instances from the dataset are shown in Figure 4.4.

The assignments of the variables in the Morpho-MNIST SCM are adapted from the equations provided by Pawlowski et al. [4]. The distribution of the slant s_m is taken to be a zero-centered Gaussian with variance of $\pi/10$, as the slant distribution was not provided by any prior works. Variable assignments are as follows in the

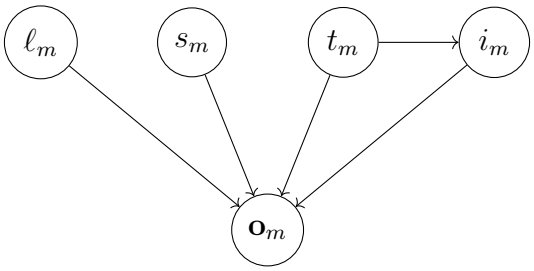


Figure 4.3: Causal graph for Morpho-MNIST, identical to the one used by Dash et al. [5]. The label, slant, thickness, and intensity are causes of the digit image, and thickness is a cause of intensity.

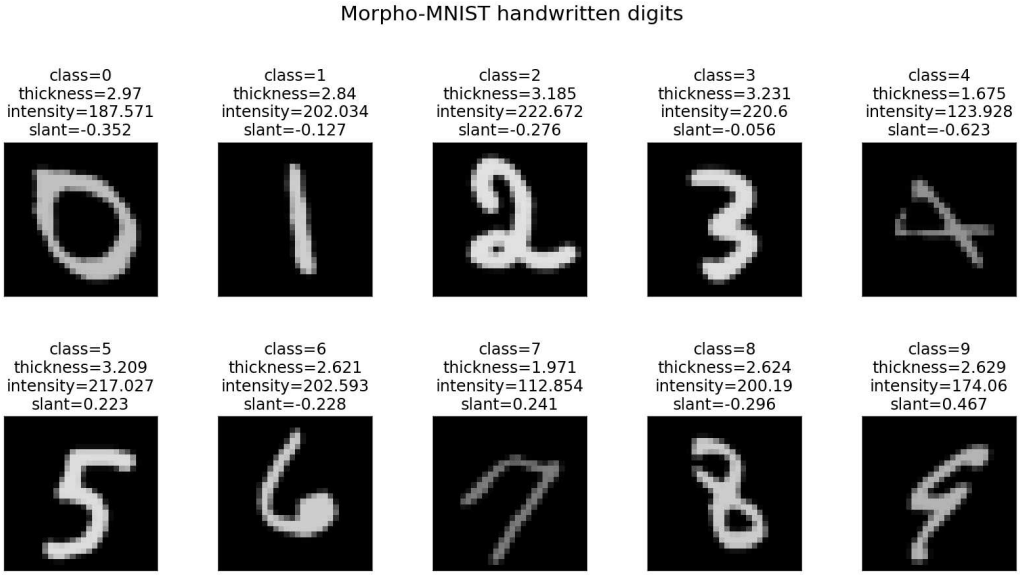


Figure 4.4: Instances from the Morpho-MNIST training set. Each digit has a varying class, thickness, intensity, and slant sampled from the ground-truth SCM.

ground-truth SCM:

$$t_m := 0.5 + \epsilon_t, \qquad \epsilon_t \sim \Gamma(10, 5) \tag{4.5}$$

$$i_m := 191\sigma(0.5\epsilon_i + 2t_m - 5) + 64, \qquad \epsilon_i \sim \mathcal{N}(0, 1) \tag{4.6}$$

$$s_m := \frac{\pi}{10}\epsilon_s, \qquad \epsilon_s \sim \mathcal{N}(0, 1) \tag{4.7}$$

$$\ell_m := \epsilon_\ell, \qquad \epsilon_\ell \sim \text{MNIST} \tag{4.8}$$

$$\mathbf{o}_m := \text{SetIntensity}(\text{SetSlant}(\text{SetThickness}(\text{SetLabel}(\epsilon_\mathbf{o}; \ell_m); t_m); i_m)), \quad \epsilon_\mathbf{o} \sim \text{MNIST} \tag{4.9}$$

In the above assignments, σ refers to the logistic sigmoid, SetIntensity and SetThickness refer to image processing operations, while SetLabel($\epsilon_{\mathbf{o}}; \ell_m$) refers to drawing an image with label ℓ_m randomly from the MNIST data (equivalent to setting the label and choosing a "random style" among those available).

To approximate the data generation process of this SCM, the following learned transformations are used for continuous attributes, which are denoted with a hat $\hat{}$ to indicate that they are approximations:

$$\hat{t}_m := \exp(\text{BatchNorm}_\theta(\hat{\epsilon}_t)), \qquad \hat{\epsilon}_t \sim \mathcal{N}(0,1) \tag{4.10}$$

$$\hat{i}_m := \text{Affine}\left(\sigma\left(\text{ConditionalAffine}_\theta(\hat{\epsilon}_i; \hat{t}_m)\right)\right), \qquad \hat{\epsilon}_i \sim \mathcal{N}(0,1) \tag{4.11}$$

$$\hat{s}_m := \text{Affine}\left(\text{Spline}_\theta(\hat{\epsilon}_s)\right), \qquad \hat{\epsilon}_s \sim \mathcal{N}(0,1). \tag{4.12}$$

The single independent categorical attribute $\hat{\ell}_m$ has its distribution taken from the MNIST training set. In Equation 4.10, $\text{BatchNorm}_\theta(\cdot)$ refers to a learned batch normalization transformation, while Affine$(\cdot)$ in Equation 4.11 and Equation 4.12 refers to a linear transformation scaling the output to the range of values found in a large sample of that attribute. The transformation $\text{ConditionalAffine}_\theta(\cdot;\cdot)$ computes the parameters of an affine transformation as a function of its second argument in order to transform its first argument.

The image observation $\mathbf{o}_m$ is modelled by both VAE and BiGAN neural networks, following from DeepSCM [4] and ImageCFGen [5] using a independent standard normal distribution with 512 components as a latent prior. For the encoder of both the BiGAN and VAE, an embedding of dimension 256 is used to compute features for the digit label, and all attributes are passed to the model according to the method described in section 4.2. Table 4.1 describes the model architecture generating encoder features for both the BiGAN and VAE models.

Layer number	Layer description
1	Conv2D(5, 64, 3, 2, 1, LeakyReLU(0.2))
2	Conv2D(64, 128, 4, 2, 1, LeakyReLU(0.2))
3	Conv2D(128, 256, 4, 2, 1, LeakyReLU(0.2))
4	Conv2D(256, 512, 4, 2, 1, LeakyReLU(0.2))
5	Conv2D(512, 512, 1, 2, 0, Identity)

Table 4.1: Description of layers in the encoder models for BiGAN and VAE used for Morpho-MNIST experiments.

Layer number	Layer Description
1	Conv2DT(772, 512, 3, 1, 0, LeakyReLU(0.2))
2	Conv2DT(512, 256, 2, 1, 0, LeakyReLU(0.2))
3	Conv2DT(256, 128, 3, 2, 1, LeakyReLU(0.2))
4	Conv2DT(128, 64, 3, 2, 1, LeakyReLU(0.2))
5	Conv2DT(64, 1, 4, 1, 1, tanh)

Table 4.2: Description of layers in the decoder models for BiGAN and VAE used for Morpho-MNIST experiments.

Layer number	Layer description
1	Dropout(0.2)
2	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.1))
3	Dropout(0.5)
4	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.1))

Table 4.3: Description of layers in the discriminator module $D_{\mathbf{z}}$ for the BiGAN used for Morpho-MNIST experiments.

Layer number	Layer description
1	Dropout(0.2)
2	Conv2D(5, 32, 5, 1, 0, LeakyReLU(0.1))
3	BatchNorm
4	Dropout(0.2)
5	Conv2D(32, 64, 4, 2, 0, LeakyReLU(0.1))
6	BatchNorm
7	Dropout(0.5)
8	Conv2D(64, 128, 4, 1, 0, LeakyReLU(0.1))
6	BatchNorm
7	Dropout(0.5)
8	Conv2D(128, 256, 4, 2, 0, LeakyReLU(0.1))
9	BatchNorm
10	Dropout(0.5)
11	Conv2D(256, 512, 3, 1, 0, LeakyReLU(0.1))

Table 4.4: Description of layers in the discriminator module $D_{\mathbf{x}}$ for the BiGAN used for Morpho-MNIST experiments.

The decoder models for BiGAN and VAE are identical for the Morpho-MNIST dataset. For both models, the latent vector is concatenated with the continuous attributes of the data and the embedding representation of the digit label. As with the encoder models, the digit label embedding for the decoder has dimension 256. These concatenated features are passed as a $772 \times 1 \times 1$ tensor to a sequence of transposed convolutional layers described in Table 4.2. The discriminator models $D_{\mathbf{z}}, D_{\mathbf{x}}$, and $D_{\mathbf{x},\mathbf{z}}$ used for Morpho-MNIST are shown in Table 4.3, Table 4.4, and Table 4.5 respectively. Both the VAE and BiGAN models are trained with a learning rate of 10^{-4} and a batch size of 64. The hyperparameters for the Adam optimizer used are $\beta_1 = 0.5, \beta_2 = 0.999$ for both models.

The classifier used for evaluating these models according to the methods described in subsection 4.4.1 is present in Table 4.6. The model was trained for 20 epochs with a learning rate of 10^{-4}, $\beta_1 = 0.9, \beta_2 = 0.999$ for Adam hyperparameters, and a batch size of 512. Results for the ImageCFGen and DeepSCM models trained on this dataset have been reported in section 5.1.

Layer number	Layer description
1	Dropout(0.2)
2	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.1))
3	Dropout(0.2)
4	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.1))
5	Dropout(0.2)
6	Conv2D(1024, 1024, 1, 1, 0, Sigmoid)

Table 4.5: Description of layers in the discriminator module $D_{\mathbf{x},\mathbf{z}}$ for the BiGAN used for Morpho-MNIST experiments. Note that this model has 1024 input channels, as $D_{\mathbf{o}}$ and $D_{\mathbf{z}}$ both output 512 feature channels (see Table 4.4 and Table 4.3).

Layer number	Layer description
1	Conv2D(1, 32, 3, 1, 0, LeakyReLU(0.2))
2	Conv2D(32, 64, 3, 2, 0, LeakyReLU(0.2))
3	Conv2D(64, 128, 3, 2, 0, LeakyReLU(0.2))
4	Conv2D(128, 256, 3, 2, 0, LeakyReLU(0.2))
5	Linear(4096, 10, Identity)

Table 4.6: Description of layers in the image classifier used for Morpho-MNIST experiments.

4.3.2 Audio-MNIST

The Audio-MNIST dataset consists of 30,000 utterances of the digits 0-9 spoken by 60 different speakers. Speakers uttered each digit 50 times, and 9 of each these runs were randomly selected to form a validation set. Other than raw audio recordings, metadata associated with each speaker is present in the dataset repository.[1] While Becker et al. [17] originally studied this dataset for audio classification, there is an obvious causal association between the metadata of the speaker and the audio the speaker produces from an utterance, motivating experiments involving learned SCMs. When combined with the digit being spoken, this metadata forms the causal attributes of the dataset considered in this work. These attributes are:

1. Biological sex of the speaker, denoted here as s_a.
2. Age demographic of the speaker, denoted here as d_a.
3. Digit label spoken, denoted here as ℓ_a.
4. Country of origin of the speaker, denoted here as c_a.
5. A binary indicator of whether or not the speaker is a native speaker of English, denoted here as n_a.
6. The accent of the speaker, denoted here as a_a.

The audio variable of the dataset is denoted $\mathbf{o}_a$. Because the Audio-MNIST data is only observational, i.e., there is no known ground-truth SCM, assumptions must be made to simplify the process of learning an SCM from the data. In this work, the following associations are assumed between the variables:

1. The country of origin causally influences whether ot not the speaker is a native speaker of English. The justification for this is somewhat obvious, as someone born in a non-English speaking country is highly unlikely to be a native speaker of English simply because of where they were born.
2. Both the country of origin and whether ot not the speaker is a native speaker of English causally influences the accent of the speaker. The justification for

[1] https://github.com/soerenab/AudioMNIST

this is that while many accents are associated directly with a particular country (e.g. a German accent), someone born in an English-speaking country who is a non-native speaker may have developed an accent from members of their household early in life.

The full set of assumed associations between the the variables is displayed in Figure 4.5. Because all variables in this dataset are discrete (age is binned into demographics to discretize d_a), all independent variables can have their distributions taken directly from a training set. Dependent variables (n_a and a_a) are learned using the Gumbel-max trick described by Pawlowski et al. [4]. Specifically, feedforward neural networks with two hidden layers and 64 neurons per hidden layer are used to compute the logits of the conditional distributions of n_a and a_a. Denoting the networks with linear activations for n_a and a_a as g_n and g_a respectively, the distributions are modelled as:

$$p(n_a = k|c_a) = \frac{\exp g_n(c_a)_k}{\sum_j \exp g_n(c_a)_j}, \tag{4.13}$$

and

$$p(a_a = k|c_a, n_a) = \frac{\exp g_a(c_a, n_a)_k}{\sum_j \exp g_a(c_a, n_a)_j} \tag{4.14}$$

respectively. The conditional attributes (causes) c_a and n_a are passed to the logit networks as one-hot encoded vectors.

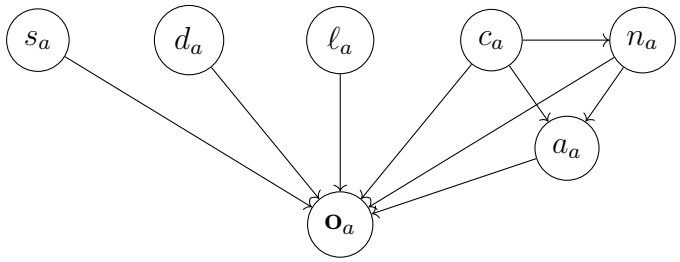


Figure 4.5: Proposed causal graph for the Audio-MNIST speech dataset. Causal relationships are assumed between the low-level variables of country of origin (c_a), native speaker (n_a), and accent (a_a).

To enable to use of CNN models when modelling the audio variable $\mathbf{o}_a$, raw waveforms are converted to log scaled power spectrograms according to the method described in section 4.1. The spectrograms have 255 frequency bins, and use a window length of 128 with a hop length of 64. Audio waveforms are zero-padded to produce

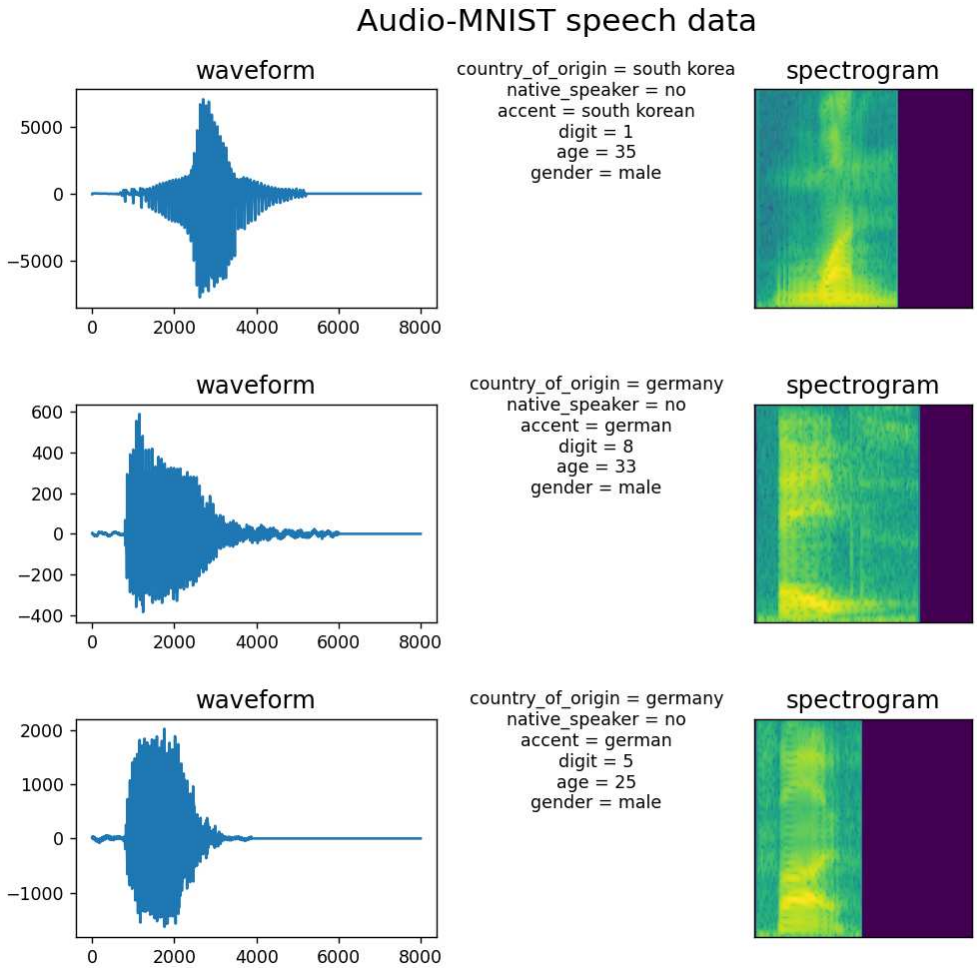


Figure 4.6: Audio waveforms (left), speaker attributes (middle), and audio spectrograms (right) from the Audio-MNIST training set. The 1-second waveforms were recorded at 8000Hz.

a 128×128 spectrogram matrix from the single-second utterances of the dataset. This allows the training process for ImageCFGen and DeepSCM to be similar to that seen in the MNIST experiment (subsection 4.3.1). Model architectures take inspiration from those used by Donahue et al. [30] in their audio synthesis paper and from previously used ImageCFGen models [5].

Encoder and decoder model architectures for Audio-MNIST are described in Table 4.7 and Table 4.8 respectively. The discriminator models $D_{\mathbf{z}}, D_{\mathbf{x}}$, and $D_{\mathbf{x},\mathbf{z}}$ are described in Table 4.9, Table 4.10, and Table 4.11 respectively. As with Morpho-MNIST, the latent prior takes the form of a 512-dimensional Gaussian and and embedding dimension of 256 is used for the discrete variables of the dataset. The causal generative models were trained with a batch size of 64, a learning rate of 10^{-4} and the default Adam hyperparameters from Pytorch [10].

Layer number	Layer description
1	Conv2D(7, 64, 5, 2, 1, LeakyReLU(0.2))
2	Conv2D(64, 128, 5, 2, 1, LeakyReLU(0.2))
3	Conv2D(128, 256, 5, 2, 1, LeakyReLU(0.2))
4	Conv2D(256, 512, 5, 2, 1, LeakyReLU(0.2))
5	Conv2D(512, 1024, 5, 2, 1, LeakyReLU(0.2))
6	Conv2D(1024, 512, 5, 2, 1, Identity)

Table 4.7: Description of layers in the encoder models for BiGAN and VAE used for Audio-MNIST experiments.

Layer number	Layer description
1	Linear(2048, 16384, LeakyReLU(0.2))
2	Reshape((1024, 4, 4))
3	Conv2DT(1024, 512, 5, 2, 2, LeakyReLU(0.2))
4	Conv2DT(512, 256, 5, 2, 2, LeakyReLU(0.2))
5	Conv2DT(256, 128, 5, 2, 2, LeakyReLU(0.2))
6	Conv2DT(128, 64, 5, 2, 2, LeakyReLU(0.2))
7	Conv2DT(64, 1, 5, 2, 2, tanh)

Table 4.8: Description of layers in the decoder models for BiGAN and VAE used for Audio-MNIST experiments.

Layer number	Layer description
1	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.2))
2	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.2))

Table 4.9: Description of layers in the discriminator module $D_{\mathbf{z}}$ for the BiGAN used for Audio-MNIST experiments.

Layer number	Layer description
1	Conv2D(7, 64, 5, 1, 0, LeakyReLU(0.2))
2	Conv2D(64, 128, 5, 1, 0, LeakyReLU(0.2))
3	Conv2D(128, 256, 5, 1, 0, LeakyReLU(0.2))
4	Conv2D(256, 512, 5, 1, 0, LeakyReLU(0.2))
5	Conv2D(512, 1024, 5, 1, 0, LeakyReLU(0.2))
6	Conv2D(1024, 512, 5, 1, 0, Identity)

Table 4.10: Description of layers in the discriminator module $D_{\mathbf{x}}$ for the BiGAN used for Audio-MNIST experiments.

Layer number	Layer description
1	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.2))
2	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.2))
3	Conv2D(1024, 1, 1, 1, 0, Sigmoid)

Table 4.11: Description of layers in the discriminator module $D_{\mathbf{x},\mathbf{z}}$ for the BiGAN used for Audio-MNIST experiments.

Layer number	Layer description
1	Conv2D(1, 32, 3, 1, 0, LeakyReLU(0.2))
2	Conv2D(32, 64, 3, 2, 0, LeakyReLU(0.2))
3	Conv2D(64, 128, 3, 2, 0, LeakyReLU(0.2))
4	Conv2D(128, 256, 3, 2, 0, LeakyReLU(0.2))
5	Conv2D(256, 512, 3, 2, 0, LeakyReLU(0.2))
6	Conv2D(512, 1024, 3, 2, 0, LeakyReLU(0.2))
7	Conv2D(1024, 3, 2, 0, LeakyReLU(0.2))
8	Linear(4096, 1024, LeakyReLU(0.2))
9	Linear(1024, 10, LeakyReLU(0.2))

Table 4.12: Description of layers in the image classifier used for Audio-MNIST experiments.

4.3.3 North American Right Whale Calls

The classification of North American Right Whale (NARW) calls is important to the monitoring of these endangered mammals. Further, passive aucoustic monitoring (PAM) produces large amounts of data that often cannot be feasibly annotated by human experts, motivating the use of automated systems. Due to the lack of large amounts of annotated data, Padovese et al. [38] investigated the use of data augmentation techniques for use while training whale call classifiers. The original data source for NARW calls, along with a detection method, was proposed and made public by Gillespie [39].[2] The dataset used in this work consists of NARW calls recorded off the coast of Massachusetts in the years 2000, 2008, and 2009. Three types of recordings are present in the data: NARW upcalls, NARW gunshot calls, and recordings that contain no NARW calls to the best of the annotator's knowledge. Because the distance of each animal to the recording devices varies greatly, the experiments in this work use a signal-to-noise ratio (SNR) calculation to eliminate upcalls consisting mostly of background noise. The SNR of a signal is calculated as:

$$SNR = \frac{\mathbb{E}[s_t]}{\text{Std}(s_t)} \tag{4.15}$$

where s_t is the time series representing the signal and $\text{Std}(s_t)$ is the standard deviation of the signal. A threshold of $SNR > -2$ was chosen empirically for upcall recordings. Information on the number of whale calls of each type (after dropping upcalls with too low SNR) for training and validation is displayed in Table 4.13.

Call Type	# Train	# Validation
No call	5370	1350
Gunshot	1796	427
Upcall	4882	982
Total	12048	2759

Table 4.13: The number of whale calls of each type for training and testing used in NARW call experiments. Counts are shown from the processed dataset, where upcalls with too low SNR have been removed.

The causal graph for this dataset only involves two variables: the categorical whale call type, t_w, and the whale call observation $\mathbf{o}_w$. The audio is transformed

[2]The NARW data is available online at `https://risweb.st-andrews.ac.uk/portal/en/datasets/dclde-2013-workshop-dataset(62c3eebc-5574-4ec0-bfef-367ad839fe1a).html`

Figure 4.7: Proposed causal graph for the NARW dataset. Only two observed variables are present, the whale call type t_w and the audio $\mathbf{o}_w$.

into a log-scaled spectrogram with 511 frequency bins, a window length of 128, and a hop length of 24. The audio signals are taken to be three seconds in length, with the whale calls centered within the time window and truncated when required. This produces a 256×256 spectrogram image for each whale call. The categorical variable t_w is processed as described in section 4.2, using a 256-dimensional embedding layer and a hyperbolic tangent activation. Table 4.14 and Table 4.15 describes the encoder and decoder model architectures used for both DeepSCM and ImageCFGen on the NARW dataset. Table 4.16, Table 4.17, and Table 4.18 describe the discriminator modules $D_{\mathbf{z}}, D_{\mathbf{x}}$, and $D_{\mathbf{x},\mathbf{z}}$, respectively. As with the previously described datasets, the latent prior for both DeepSCM and ImageCFGen is taken to be a 512-dimensional independent Gaussian. Both models are trained with a batch size of 32 and the default hyperparameters for Adam from Pytorch. The ImageCFGen BiGAN model is trained with a learning rate of 10^{-4}, while the DeepSCM VAE model is trained with a learning rate of 10^{-5} due to exploding gradients occurring when training with larger learning rates.

Layer number	Layer description
1	Conv2D(2, 64, 5, 2, 1, LeakyReLU(0.2))
2	Conv2D(64, 128, 5, 2, 1, LeakyReLU(0.2))
3	Conv2D(128, 256, 5, 2, 1, LeakyReLU(0.2))
4	Conv2D(256, 512, 5, 2, 1, LeakyReLU(0.2))
5	Conv2D(512, 1024, 5, 2, 1, LeakyReLU(0.2))
6	Conv2D(1024, 1024, 5, 2, 1, LeakyReLU(0.2))
7	Conv2D(1024, 512, 5, 2, 1, Identity)

Table 4.14: Description of layers in the encoder models for BiGAN and VAE used for NARW call experiments.

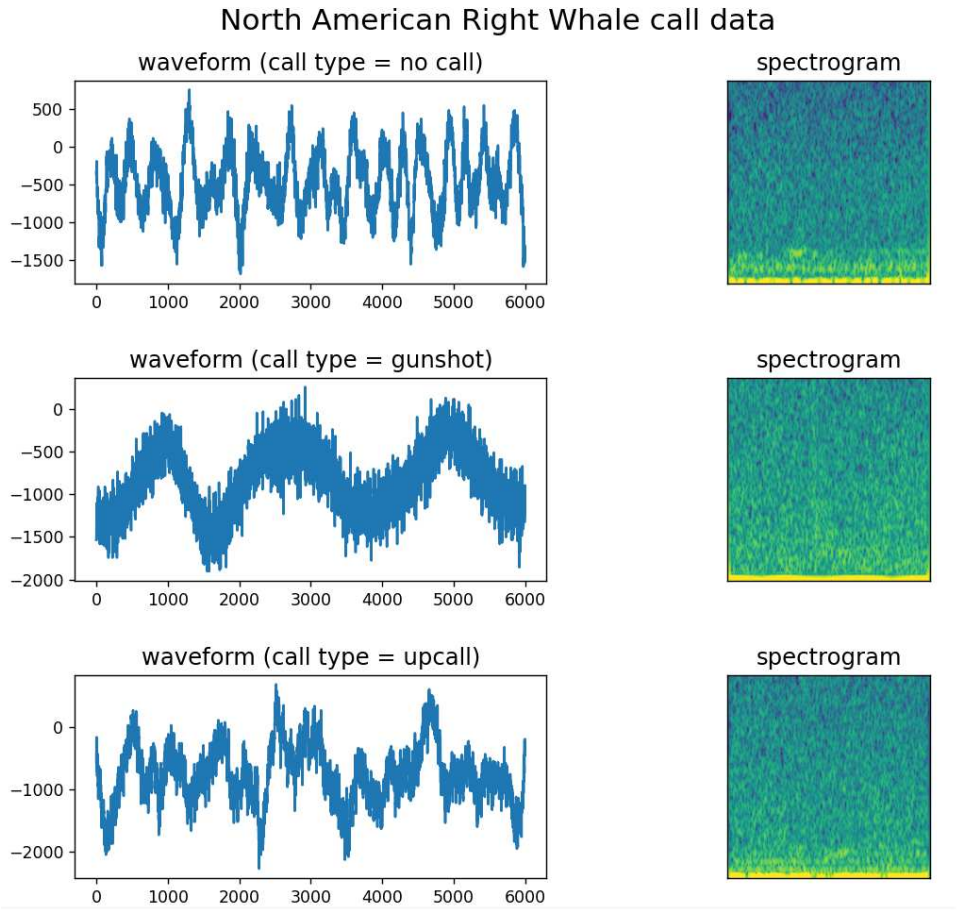


Figure 4.8: Waveforms (left) and spectrograms (right) from the training set of the NARW dataset. One sample is shown for each type of whale call.

Layer number	Layer description
1	Linear(768, 16384, LeakyReLU(0.2))
2	Reshape((1024, 4, 4))
3	Conv2DT(1024, 1024, 5, 2, 2, LeakyReLU(0.2))
4	Conv2DT(1024, 512, 5, 2, 2, LeakyReLU(0.2))
5	Conv2DT(512, 256, 5, 2, 2, LeakyReLU(0.2))
6	Conv2DT(256, 128, 5, 2, 2, LeakyReLU(0.2))
7	Conv2DT(128, 64, 5, 2, 2, LeakyReLU(0.2)
8	Conv2DT(64, 1, 5, 2, 2, tanh)

Table 4.15: Description of layers in the decoder models for BiGAN and VAE used for NARW call experiments.

Layer number	Layer description
1	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.2))
2	Conv2D(512, 512, 1, 1, 0, LeakyReLU(0.2))

Table 4.16: Description of layers in the discriminator module $D_{\mathbf{z}}$ for the BiGAN used for NARW call experiments.

Layer number	Layer description
1	Conv2D(7, 64, 5, 1, 0, LeakyReLU(0.2))
2	Conv2D(64, 128, 5, 1, 0, LeakyReLU(0.2))
3	Conv2D(128, 128, 5, 1, 0, LeakyReLU(0.2))
4	Conv2D(128, 256, 5, 1, 0, LeakyReLU(0.2))
5	Conv2D(256, 512, 5, 1, 0, LeakyReLU(0.2))
6	Conv2D(512, 1024, 5, 1, 0, LeakyReLU(0.2))
7	Conv2D(1024, 512, 5, 1, 0, Identity)

Table 4.17: Description of layers in the discriminator module $D_{\mathbf{x}}$ for the BiGAN used for NARW call experiments.

Layer number	Layer description
1	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.2))
2	Conv2D(1024, 1024, 1, 1, 0, LeakyReLU(0.2))
3	Conv2D(1024, 1, 1, 1, 0, Sigmoid)

Table 4.18: Description of layers in the discriminator module $D_{\mathbf{x},\mathbf{z}}$ for the BiGAN used for NARW call experiments.

Layer number	Layer description
1	Conv2D(1, 32, 3, 1, 0, LeakyReLU(0.2))
2	Conv2D(32, 64, 3, 2, 0, LeakyReLU(0.2))
3	Conv2D(64, 128, 3, 1, 0, LeakyReLU(0.2))
4	Conv2D(128, 256, 3, 2, 0, LeakyReLU(0.2))
5	Conv2D(256, 512, 3, 2, 0, LeakyReLU(0.2))
6	Conv2D(512, 1024, 3, 2, 0, LeakyReLU(0.2))
7	Conv2D(1024, 1024, 3, 2, 0, LeakyReLU(0.2))
8	Conv2D(1024, 1024, 3, 2, 0, LeakyReLU(0.2))
9	Linear(4096, 1024, LeakyReLU(0.2))
10	Linear(1024, 3, Identity)

Table 4.19: Description of layers in the image classifier used for NARW call experiments.

4.4 Evaluation of Counterfactuals

4.4.1 Evaluation by Attribute Classifiers

Given an encoder-decoder model with encoder E_i, decoder G_i, and target prior $p(\mathbf{Z}_i)$, one question of the validity of the model is how well G_i can condition itself on attributes; that is, how well setting the value of the causal parents $\mathbf{PA}_i$ produces data that shares characteristics of known data with similar values of $\mathbf{PA}_i$. For example, in Morpho-MNIST or Audio-MNIST, one may want to know how well setting the digit label ℓ_m to '9' causes a reasonable handwritten digit or recording of speech to be generated. To this end, consider an SCM $\mathcal{M}$ on n variables, in which some observed variable Y takes the form of a categorical random variable and $\mathbf{X}_i$ takes the form of an image representation of the data, such that Y is a causal parent of $\mathbf{X}_i$. If Y is of particular interest, we can train a classifier $f(\mathbf{X}_i) = Y$ to evaluate the validity of counterfactuals in an automated fashion. Naturally, to test the ability of G_i alone, we can generate random instances of the variable $\mathbf{X}_i$ using randomly generated causal parents $\mathbf{PA}_i$ and latent vectors $\mathbf{Z}_i$, checking if the classification of the generated instances is consistent with the desired label $Y \in \mathbf{PA}_i$. Letting $\mathrm{Acc}(a, b)$ equal 1 when $a = b$ and 0 otherwise, the following formula for a generator evaluation is proposed:

$$\mathrm{score}_G = \mathbb{E}_{p(\mathbf{PA}_i)p(\mathbf{Z}_i)}\left[\mathrm{Acc}(Y, f(G_i(\mathbf{Z}_i, \mathbf{PA}_i)))\right]. \tag{4.16}$$

Alternatively, the expected value over $\mathbf{Z}_i$ used to compute score_G can be replaced by computing latent codes of observational data using the encoder E_i. Given an observation $(\mathbf{X}_i, \mathbf{PA}_i)$ from e.g. a validation set, and a distribution of alternative classes $q(Y')$, a score accounting for the ability of E_i to compute effective latent representations is:

$$\mathrm{score}_{EG} = \mathbb{E}_{q(Y')}\left[\mathrm{Acc}\left(Y', f\left(G(E(\mathbf{X}_i, \mathbf{PA}_i), \mathbf{PA}_i^{Y \leftarrow Y' | \mathbf{PA}_i})\right)\right)\right]. \tag{4.17}$$

Note that this score actually involves computing a counterfactual, and as such measures the ability for the SCM to produce believable counterfactuals of the form "what would $\mathbf{X}_i$ have been if the variable Y had instead had value Y'?". Here, $\mathbf{PA}_i^{Y \leftarrow Y' | \mathbf{PA}_i}$ are the counterfactual values of $\mathbf{PA}_i$ computed by performing the intervention $do(Y = Y')$ in the counterfactual attribute SCM conditioned on the observed

values of $\mathbf{PA}_i$. When approximating the above value, samples from the distribution over Y' which have the same value as that of the observed set of attributes are discarded to ensure counterfactuals are not simply reconstructions of the original data. In chapter 5, the values of the two metrics proposed above are presented for each dataset and model considered.

4.4.2 Evaluation by an Audio-MNIST Subject Classifier

In certain cases, the nature of a dataset (specifically, the abundance of data of a certain type) can facilitate comparisons of counterfactual data with observational data from a test set via a classifier trained on observational data. The setting of Audio-MNIST (subsection 4.3.2) allows such a comparison due to the fact that each speaker in the dataset utters each of the digits "zero" through "nine" multiple times. Thus, when we ask the counterfactual "what would the utterance sound like if it had contained a different digit", we can compare this to the subset of data in which the same speaker uttered the digit in question by querying a classifier trained to detect speakers. For example, let $\mathbf{X}, \mathbf{PA}$ denote the audio and attribute observations of Audio-MNIST, and denote $\mathbf{X}_k^c$ be the counterfactual obtained from a deep generative model by performing abduction on the given observation and the intervention $do(\text{digit} = k)$. Further, denote $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ as the subset of observational validation data such that each $\mathbf{X}_k^{(i)}$ is an utterance of the digit k from the same subject which produced the original utterance $\mathbf{X}$. To evaluate the abduction process, a subject classifier is trained on the Audio-MNIST training set. Then, beyond merely checking that the digit k is classified correctly, one can check the agreement between the classifier and the subject associated with the original observation. The mean agreement is used as a form of accuracy on the abduction process in section 5.2. In the case of the Audio-MNIST validation set, there are 60 unique subjects speaking 10 unique digit utterance types 9 times each. Hence, the number of counterfactual utterances for a given counterfactual digit k and a given subject is $9 \times 9 = 81$, giving $81 \times 60 \times 10 = 48600$ counterfactuals to check the agreement between causal models and the subject classifier.

4.5 Counterfactual Explanations with Causal Generative Models

This section proposes a method of explaining a classifier f using the encoder and decoder models of either a BiGAN or VAE, which are denoted E and G respectively. Dash et al. [5] define a counterfactual importance score using the ImageCFGen architecture, which measures how a classifier's prediction changes when a given attribute (cause) of the classifier's input changes. For instance, measuring how an "attractive" classifier's output changes when the attribute "bald" changes. While this method proved useful in the work by Dash et al., it is not without limitations. Namely, that it is only defined for binary attributes and binary classifiers. These limitations motivate the method described in this section, which aims to provide image-based counterfactual explanations on datasets with continuous and categorical variables of interest. This method uses the BiGAN and VAE architectures of ImageCFGen and DeepSCM respectively to generate explanations which lie in the original space of images. Specifically, we focus on counterfactual explanations, which provide alternative values of variables which lead to a different classification from a certain classifier.

Definition 6 ([40]) *Given a classifier f that outputs the decision y for an instance $\mathbf{x}$, a counterfactual explanation consists of an instance $\mathbf{x}'$ such that the decision for f on $\mathbf{x}'$ is different from y, i.e., $f(\mathbf{x}') \neq y$, and such that the difference between $\mathbf{x}$ and $\mathbf{x}'$ is minimal. As such, the instance $\mathbf{x}'$ should be a realistic instance from the same space as $\mathbf{x}$.*

Ideally, counterfactual explanations come from a "closest possible world" [12], i.e., minimal changes to variables are performed while still producing realistic data. This concept of minimality is left without a rigorous definition in Definition 6 due to this concept being specific to the method of explanation used [40]. The OmnixAI library [13] is a publicly available software toolkit which provides a method of computing counterfactual explanations. For a given image $\mathbf{x}$ and classifier $f(\cdot)$, the counterfactual is found by solving the following optimization problem:

$$\min_{\mathbf{x}'} \max_{\lambda} \lambda \mathcal{H} \left(f_y(\mathbf{x}') - \max_{y' \neq y} f_{y'}(\mathbf{x}') \right) + ||\mathbf{x}' - \mathbf{x}||_1 \tag{4.18}$$

where f_j denotes the score for class j, y is the prediction of f for the original input $\mathbf{x}$, and $\mathcal{H}$ is the hinge loss function. This optimization problem aims to find $\mathbf{x}'$ which flips

the prediction of f while also keeping $\mathbf{x}'$ close to $\mathbf{x}$ via the L_1 regularization term. One potential downfall of this approach is that while $\mathbf{x}'$ is close to $\mathbf{x}$, it may not represent a realistic image, i.e., it may not fall within a "possible world" as described by Wachter et al. [12]. Such examples may be considered *adversarial*, as they are not easily interpreted by humans and are very specific to the classifier being explained.

To generate counterfactual examples which remain in the space of realistic images, this work proposes using the encoder E and decoder G of a causal generative model to search in the space of possible counterfactuals for an image $\mathbf{x}$. This way, the search for $\mathbf{x}'$ becomes a search over possible counterfactual attributes $\mathbf{pa}'$. As in Equation 3.7, the counterfactual is defined as:

$$\mathbf{x}' = G(E(\mathbf{x}, \mathbf{pa}), \mathbf{pa}') \tag{4.19}$$

where $\mathbf{pa}$ are the original causes of $\mathbf{x}$. Accounting for a desired target class y_t of $\mathbf{x}'$, the optimization problem for the gradient-based method of counterfactual explanation proposed here takes the following form:

$$\min_{\mathbf{pa}'} \max_{\lambda} \lambda \left(\max_{y' \neq y_t} f_{y'}(\mathbf{x}') - f_{y_t}(\mathbf{x}') \right) + ||\mathbf{x}' - \mathbf{x}||_1 \tag{4.20}$$

where $\mathbf{x}'$ is defined in terms of $\mathbf{x}, \mathbf{pa}$, and $\mathbf{pa}'$ as in Equation 4.19. In practice λ was set to 10 rather than performing an additional search over the parameter for each counterfactual explanation.

While a search over continuous attributes is straightforward, a search over categorical attributes can be more difficult to achieve. As discussed in section 4.2, categorical attributes are passed to the generator G via an embedding lookup function $e(\cdot)$ learned during training. If the search space over a categorical attribute $a \in \mathbf{PA}'$ is converted to a search over discrete probability distributions $p(a)$, we can represent an element of the search space as a softmax vector $\mathbf{p}$ with $\mathbf{p}_k = p(a = k)$. Then, instead of passing the embedding vector $e(a)$ of a single categorical attribute, we pass the expected vector for the given element of the search space:

$$\mathbb{E}_{p(a)}[e(a)] = \sum_k \mathbf{p}_k e(k). \tag{4.21}$$

This allows the counterfactual explanations to be "inbetween" values of a given categorical attribute (e.g. the digit 3 and 8) while still producing realistic image styles via the generator G.

The previously described method of embedding interpolation also allows for a model-agnostic search for counterfactuals to be performed. Letting Y be the categorical variable classified by f, we can interpolate between y (the original classification) and y' (the desired classification) to find the minimum change required in order to flip the label. Let $\mathbf{PA}_t$ denote the original attribute values with the embedding for Y replaced by the linear combination $te(y') + (1-t)e(y)$, such that $\mathbf{PA}_0 = \mathbf{PA}$. A model-agnostic search can then be performed by finding the smallest value of t such that:

$$\arg\max_j f_j\left(G(E(\mathbf{x}, \mathbf{PA}), \mathbf{PA}_t)\right) = y'. \tag{4.22}$$

This can be approximated via a grid search over several values of $t \in [0, 1]$. Because this method only requires querying the classifier rather than backpropagating through it, it is naturally faster than the gradient-based optimization problem described in Equation 4.20. However, it also only changes a single attribute (along a single axis) and does not perform as sophisticated of a search.

4.5.1 Evaluating Counterfactual Explanations

The above counterfactual explanation methods, which are referred to as gradient-based and model agnostic, are compared with methods implemented in the OmnixAI toolkit [13] on the Morpho-MNIST dataset. Specifically, they are compared with the counterfactual image explainer and the contrastive image explainer from OmnixAI, which were originally proposed by Wachter et al. [12] and Dhurandhar et al. [14], respectively. The comparison is made based on the hypothesis that the methods from OmnixAI may produce small, adversarial-like changes rather than meaningful, interpretable changes in the data to explain a classifier. To test this hypothesis, quantitative metrics are employed which measure the interpretability of generated counterfactuals.

Definition 7 ([15]) *A counterfactual explanation $\mathbf{x}'$ of a model f is considered interpretable if it lies close to the model's training distribution. This applies not just to the overall distribution, but also to training instances which belong to the counterfactual class.*

The first metrics come from Van Looveren and Klaise [15], who proposed metrics

of IM1 and IM2 and who also provide the definition interpretability from Definition 7. For an image $\mathbf{x}$ from class p and corresponding counterfactual $\mathbf{x}'$ supposedly from class q, IM1 measures whether $\mathbf{x}'$ is closer to class p or q using autoencoders AE_p and AE_q trained on data coming from only the respective classes:

$$\text{IM1} = \frac{||\mathbf{x}' - \text{AE}_q(\mathbf{x}')||_2^2}{||\mathbf{x}' - \text{AE}_p(\mathbf{x}')||_2^2 + \varepsilon}. \tag{4.23}$$

If IM1 falls below 1, the counterfactual is better reconstructed by a model trained on data from the target class, suggesting it can be easily interpreted as an instance from the target class. IM2, also proposed by Van Looveren and Klaise, aims to measure how well the counterfactual follows the overall distribution of data by utilizing an autoencoder AE trained on the entire training set:

$$\text{IM2} = \frac{||\text{AE}_q(\mathbf{x}') - \text{AE}(\mathbf{x}')||_2^2}{||\mathbf{x}'||_1 + \varepsilon}. \tag{4.24}$$

Similarly to IM1, lower values of IM2 are considered better by the authors who proposed the metrics. These metrics have been met with some criticism, including from Hvilshøj et al. [16] who found that as dataset complexity increases, differences in the values of these metrics between explanation methods can become insignificant. Despite this, the metrics are included in this work due to the relatively low complexity of Morpho-MNIST digits.

Additional quantitative evaluation metrics have been proposed by Hvilshøj et al. [16], who aim to determine whether the changes found in a counterfactual explanation are too specific to the classifier being explained to be interpretable (i.e., adversarial). The metrics proposed by Hvilshøj et al. are based on the presence of an *oracle* o, which is an additional classifier trained with a different initial random state than the one used to train the explained classifier f. While a label variation score (LVS) is proposed by Hvilshøj et al., the metric is stated to be suitable only for datasets where more than one label is present. Instead, we use the *oracle score*, a measure of agreement between the classifier f and oracle o on the counterfactual data. Letting $\mathcal{X}'$ denote a set of counterfactuals explanations from a given method, the oracle score is defined as:

$$\text{Oracle} = \frac{1}{|\mathcal{X}'|} \sum_{\mathbf{x}' \in \mathcal{X}'} \text{Acc}(f(\mathbf{x}'), o(\mathbf{x}')) \tag{4.25}$$

where Acc is a binary indicator of equality as in Equation 4.17. If this score is very low, it indicates that the counterfactual explanations in $\mathcal{X}'$ may be very specific to the weights of f, hence it is used to avoid giving good scores to adversarial-like changes.

Results for the chosen quantitative metrics above on the four proposed explanation methods (BiGAN and VAE, gradient-based and agnostic) and the two explanation methods from OmnixAI are presented in section 5.1. When reporting values of IM1 and IM2, a normal approximation is used to compute 95% confidence intervals. At the 95% confidence level, this interval is of the form:

$$\bar{x} \pm 1.96 \frac{s}{\sqrt{n}} \tag{4.26}$$

where $\bar{x}$ is the observed sample mean of the metric, s is the sample standard deviation, and n is the sample size (10,000 for Morpho-MNIST). The 1.96 value in Equation 4.26 comes from the 97.5th percentile of the standard normal distribution. For the oracle score, we measure the change with which the score changes when a different oracle is used, training 10 oracles with different random initializations and forming a 95% confidence interval using Equation 4.26.

Chapter 5

Results

This chapter displays the results from experiments on the three datasets considered in this work. Both the evaluation of the DeepSCM and ImageCFGen models on the datasets, as well as the results for counterfactual explanation on the Morpho-MNIST dataset, are reported. When reporting agreement between the generative models and classifiers in the observational setting (Equation 4.16), no results are reported for the fine-tuned ImageCFGen model as the generator is not modified during fine-tuning.

When reporting metrics for causal models' generation capabilities (e.g., Equation 4.16 and Equation 4.17) in tables, two values are presented for each metric. This is because limited resources led to each causal model only being retrained once.

5.1 Morpho-MNIST

When evaluating the effectiveness of DeepSCM and ImageCFGen to model the Morpho-MNIST dataset, the direct measurement of thickness, intensity, and slant was performed using morphological operations, similarly to work from Pawlowski et al. [4]. Specifically, a random intervention of each type (thickness, intensity, and slant) was performed on each of the images in the Morpho-MNIST test set, and then target attributes were measured to check for consistency with causal models. This experiment was reproduced for ImageCFGen, DeepSCM, as well as the fine-tuned ImageCFGen model to ensure the training process of the models is consistent with previously published work. The measured values extracted from counterfactuals are shown in Figures 5.1, 5.2, and 5.3.

In addition to evaluation of counterfactuals concerning continuous attributes of Morpho-MNIST using direct measurement, counterfactuals for the digit label were evaluated using a trained classifier using the methods from subsection 4.4.1. Table 5.1 shows the evaluation of digit counterfactuals, as well as the evaluation of digits generated using random latent vectors and attributes from the Morpho-MNIST test set.

	ImageCFGen		ImageCFGen (ft)		DeepSCM	
Digit (obs)	0.9810	0.9843	-		0.9916	0.9900
Digit (cf)	0.9533	0.9678	0.9979	0.9987	0.9854	0.9851

Table 5.1: Classification-based scoring of the Morpho-MNIST handwritten digit causal models. The top row displays scores in the observational setting, where attributes are taken from the test set and latent features are randomized. The bottom row displays scores in the counterfactual setting, where a random intervention is performed on each digit in the test set.

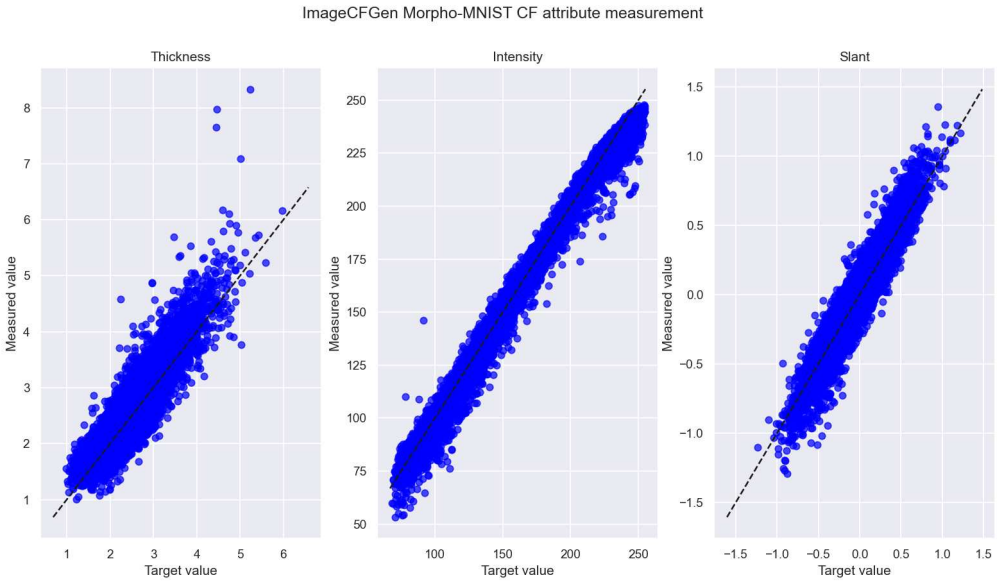


Figure 5.1: Measured thickness, intensity, and slant values from Morpho-MNIST counterfactuals computed by the trained ImageCFGen model described in section subsection 4.3.1.

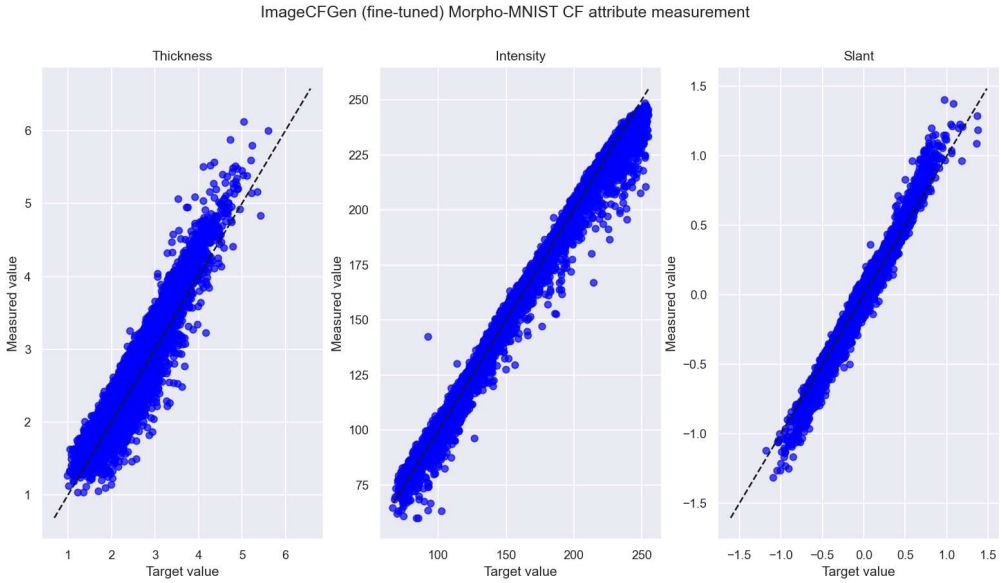


Figure 5.2: Measured thickness, intensity, and slant values from Morpho-MNIST counterfactuals computed by the trained ImageCFGen model after fine-tuning using the method described in section 3.3.

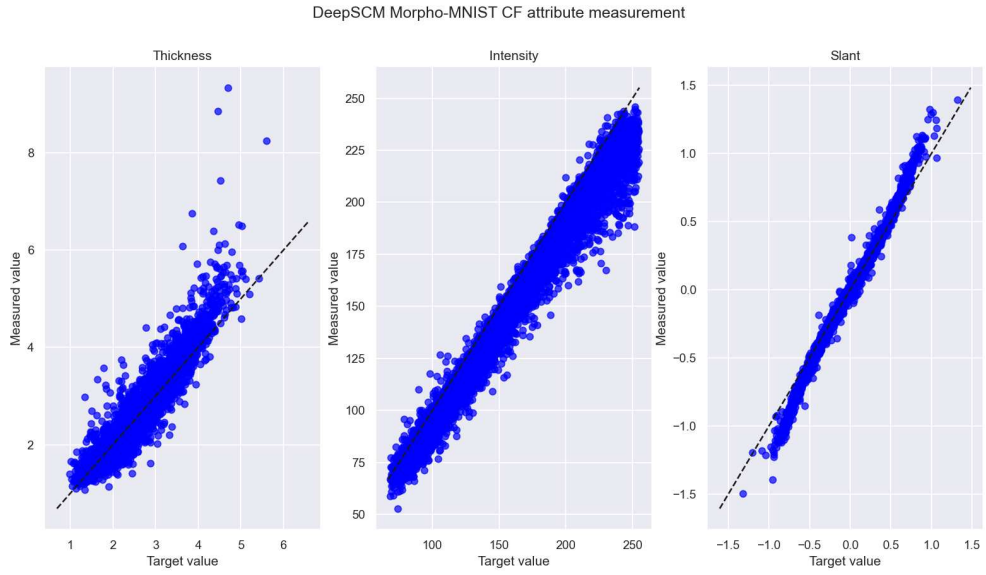


Figure 5.3: Measured thickness, intensity, and slant values from Morpho-MNIST counterfactuals computed by the trained DeepSCM model described in subsection 4.3.1.

All three models accurately model counterfactuals of the three continuous attributes, with thickness being the most difficult of the three attributes to model. Further, all models are able to produce counterfactuals concerning digit labels with very high accuracy as measured by a classifier. In the case of both continuous attributes and the digit label, finetuning the ImageCFGen model increases the model's ability to produce accurate counterfactuals. Figure 5.4 shows examples of thickness counterfactuals, and Figure 5.5 shows reconstructions of Morpho-MNIST digits computed by each of the trained generative causal models.

In addition to the evaluation of counterfactual inference discussed above, the Morpho-MNIST dataset is also used as a case study for the counterfactual explanation methods described in section 4.5. For each image in the test set, a counterfactual explanation is produced to explain a classifier trained on the Morpho-MNIST training set. As the contrastive explainer from OmnixAI does not use a target class, the class of the example produced by the contrastive explainer is used as the target class of the VAE and BiGAN-based explainers in order to make a fair comparison. Classifiers, including oracles, were trained with the same procedure as the classifer used to evaluate the validity of digit counterfactuals.

An example of visual classifier explanation for the methods proposed here as well as the counterfactual and contrastive explainer from OmnixAI are shown in Figure 5.6. The gradient-based methods seem to fully transform the provided zero into a six with the same style, while the model-agnostic methods peform a smaller change to

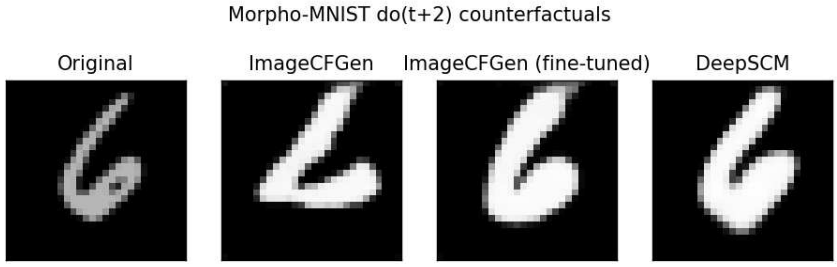


Figure 5.4: Counterfactuals computed by the three causal generative models on the Morpho-MNIST dataset. The counterfactual shown is produced increasing the thickness attribute t_m by two units. Note that intensity is increased along with thickness due to the causal structure of Morpho-MNIST (see Figure 4.3).

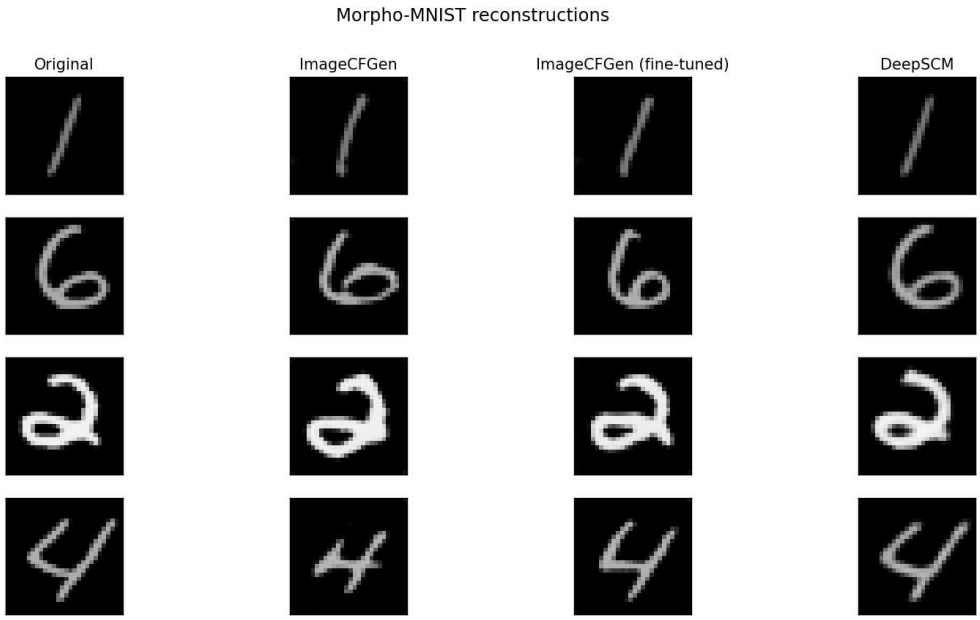


Figure 5.5: Reconstructions (left-center, right-center, right) of original images (left) from the Morpho-MNIST test set. All models produce accurate reconstructions, though the DeepSCM autoencoder seems to produce images visually closest to the originals.

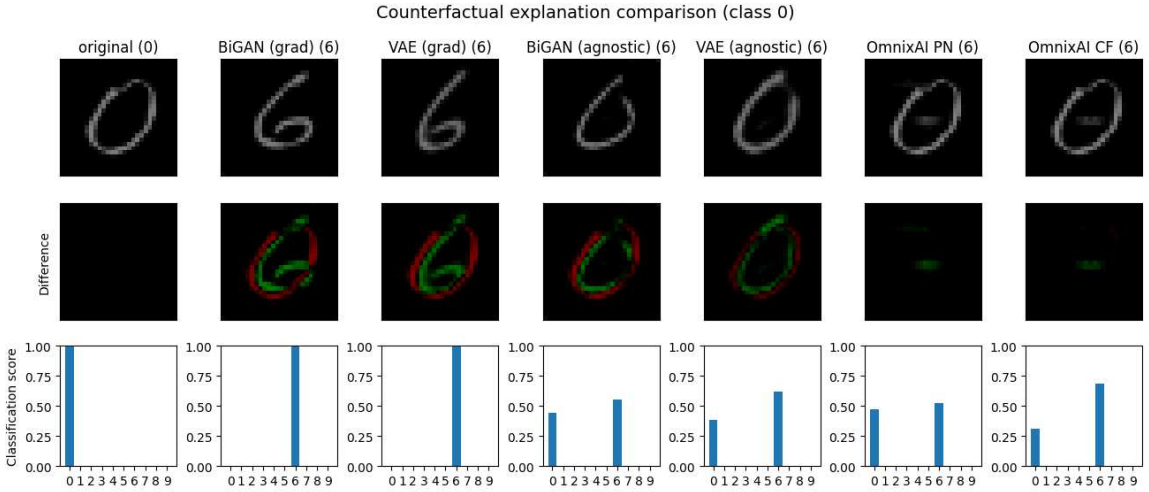


Figure 5.6: Comparison of visual explanation methods proposed in this work with the counterfactual and contrastive explainers from OmnixAI. The class of each explanation is shown in the title of each subfigure with the name of the explanation method, and the class score distribution from the classifier is shown underneath each image. The label "OmnixAI PN" refers to the contrastive explainer, which produces what are called pertinent negatives.

the image by removing part of the zero to make it closer to a six. The OmnixAI methods, however, appear to add small pixels to the center of the zero. The agnostic and OmnixAI methods also appear to leave the classifier with a nonzero score for both the original class (0) and target class (6), while the gradient-based methods fully flips the score distribution. Additional examples of classifier explanations are provided in Appendix B.

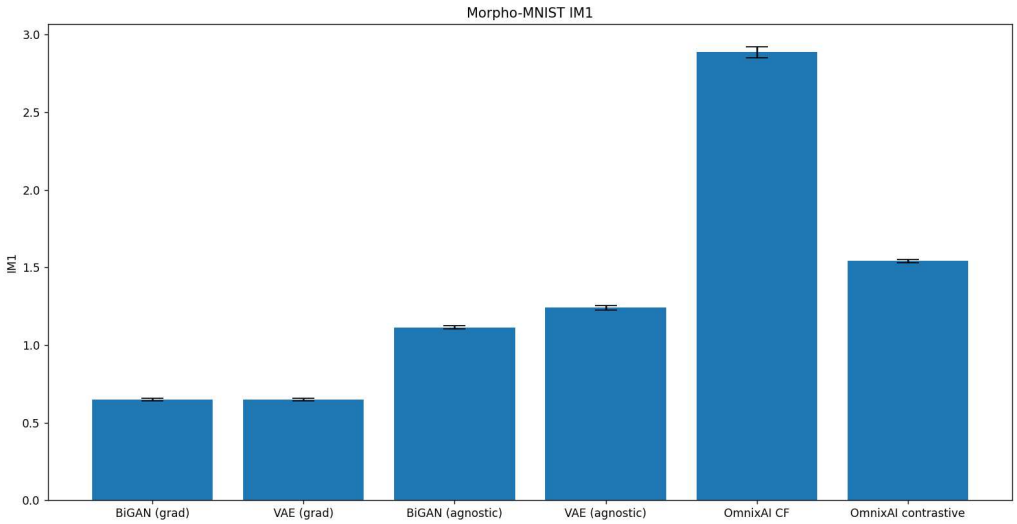


Figure 5.7: Mean IM1 scores computed on the Morpho-MNIST test set for the visual explanation methods considered in this work. Confidence intervals at the 95% level are shown as error bars on each value. All methods proposed in this work (those using VAE or BiGAN) have significantly better performance than those from OmnixAI at the 95% confidence level, with gradient-based methods performing the best.

Results for counterfactual explanation evaluation metrics are shown in Figure 5.7, Figure 5.8, and Figure 5.9. The gradient-based explainers clearly outperform the rest in terms of IM1 (see Figure 5.7). Further, the OmnixAI CF explainer has comparatively high IM1 scores compared to the rest of the methods. The agnostic methods have similar performance on IM1 to the OmnixAI contrastive explainer, though they still outperform the contrastive explainer at the 95% confidence level. For the IM2 metric, all methods except the gradient-based VAE explainer outperforms the methods from OmnixAI at the 95% confidence level (see Figure 5.8). Examining the oracle score values in Figure 5.9, the gradient-based explainers proposed in this work obtain the highest scores. However, both the gradient-based and model-agnostic explainers have higher scores than the methods from OmnixAI. The low score from the OmnixAI

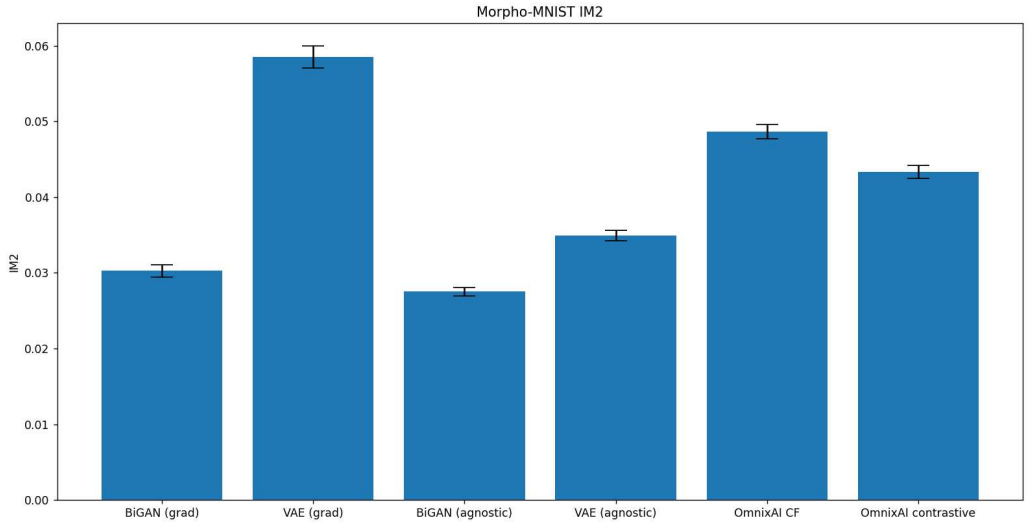


Figure 5.8: Mean IM2 scores computed on the Morpho-MNIST test set for the visual explanation methods considered in this work. Confidence intervals at the 95% level are shown as error bars on each value. All methods have significantly different performance at this confidence level, with the agnostic BiGAN method performing the best.

counterfactual explainer in particular suggests the changes provided by this method may be adversarial. This observation is reinforced by the unrealistic changes to the image made by the OmnixAI methods in Figure 5.6.

5.2 Audio-MNIST

Although the Audio-MNIST dataset contains several different attributes, not all can be easily measured by a classifier (e.g. country of origin). Because of this, three attributes were chosen to train spectrogram classifiers for the evaluation of the trained generative models in this work. The three attributes chosen are the speaker's biological sex, the speaker's accent, and the digit spoken by the speaker. The accuracy of the trained classifiers is shown for each attribute in Table 5.2. The classifiers achieve over 95% accuracy on the validation set in all three settings, with the speaker's accent being the most challenging attribute to classify.

This dataset was used as a case study in the use of the subject classifier agreement metric from subsection 4.4.2. The metric produces a single binary value for each

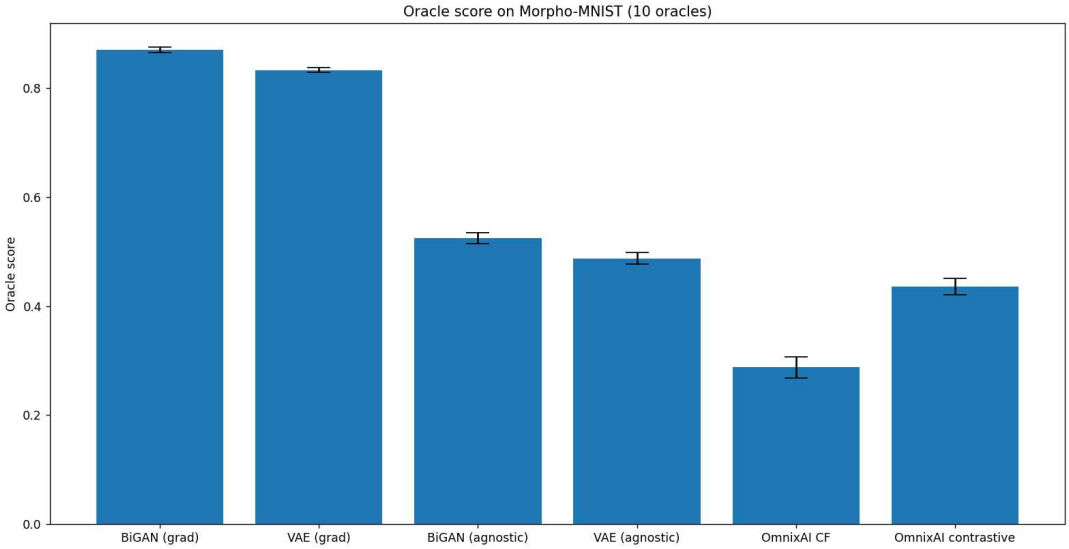


Figure 5.9: Oracle scores for each of the explanation methods considered in this work, computed across the Morpho-MNIST test set. Confidence intervals at the 95% confidence level are shown by error bars at the top of each score bar, which were computed using samples collected from 10 oracles with different random initializations to remove bias that would be introduced from the weights of a single oracle.

Attribute	Classifier Validation Accuracy
Speaker's biological sex	0.9985
Digit spoken	0.9946
Speaker's accent	0.9522

Table 5.2: Validation accuracy for different attribute classifiers trained on the Audio-MNIST dataset.

possible digit counterfactual that can be produced from the validation set. The classifier trained has the same architecture as the other Audio-MNIST classifiers used, and was trained for 10 epochs. The subject classifier obtained 96.2% accuracy on the validation set. The agreement metric was recorded for the ImageCFGen and DeepSCM models as well as the fine-tuned ImageCFGen model. To test whether the counterfactuals produced matched the desired subject better than a simpler interventional distribution, the generators of the two models were also used to produce alternative images with the same target label without being counterfactuals (no abduction). These models are referred to as "GAN" and "VAE Decoder" in Table 5.3. Examining the classifier agreement results, even the interventional models achieve a nontrivial level of agreement on subject with the classifier (ranging from approximately 25% to 30%). This could be explained in part by some subjects in the dataset having attributes such as accent unique to only them. Using this range as a baseline, both ImageCFGen and its fine-tuned version fail to beat the baseline. However, the DeepSCM model achieves almost double the level of the agreement of the other models, suggesting it most accurately preserves the speaker's voice when computing digit counterfactuals.

ImageCFGen		ImageCFGen (ft)		DeepSCM		GAN^{nc}		$\text{VAE Decoder}^{\text{nc}}$	
0.2177	0.2391	0.2532	0.2543	0.5802	0.5722	0.2489	0.2416	0.3031	0.3047

Table 5.3: Mean agreement on subject for digit counterfactuals as described in subsection 4.4.2. The causal models ImageCFGen, fine-tuned ImageCFGen, and DeepSCM are compared with non-causal generative models GAN^{nc} and $\text{VAE Decoder}^{\text{nc}}$ formed by removing the encoder from an ImageCFGen or DeepSCM model (see Figure 4.2b). The $^{\text{nc}}$ superscript in the names of these models stands for non-causal. The comparison between these models is made to determine their relative abilities to preserve a subject's voice when changing the utterance being said.

Attribute	ImageCFGen		DeepSCM	
Speaker's biological sex	0.9733	0.9846	1.0	1.0
Digit spoken	0.9811	0.9659	0.9981	0.9980
Speaker's accent	0.8002	0.8028	0.8754	0.8793

Table 5.4: Classification-based scoring of the Audio-MNIST generator (Equation 4.16) models for prominent categorical attributes. Each score is computed over the Audio-MNIST validation set with 4 Monte-Carlo samples from $p(\mathbf{z})$ for each set of attributes.

As with all datasets, the ability for DeepSCM and ImageCFGen to model categorical attributes is measured using classifiers according to the methods from subsection 4.4.1. The models are evaluated on their ability to reproduce data with attributes from the distribution of the validation set (Table 5.4) and on their ability to produce believable counterfactuals on given attributes (Table 5.5). Original images for counterfactuals are taken from the validation set, and new attributes are sampled from the trained attribute SCM described in subsection 4.3.2.

In the observational setting (Table 5.4), high agreement between the classifier and the generative causal models is achieved. However, the DeepSCM model seems to have an advantage over ImageCFGen, especially in the case of the speaker's accent where a discrepancy of over 7% is observed and duplicated on a second run of training. Results on counterfactuals seen in Table 5.5 are consistent with results on different datasets from Dash et al. [5] in that fine-tuning the ImageCFGen architecture improves the ability of the model to produce believable counterfactuals. In fact, the fine-tuned ImageCFGen appears to give comparable performance to DeepSCM in the counterfactual setting. Interestingly, the model is able to produce believable digit and sex counterfactuals, but not believable accent counterfactuals. This shortcoming is discussed further in chapter 6.

Attribute	ImageCFGen		ImageCFGen (ft)		DeepSCM	
Speaker's biological sex	0.8950	0.9174	0.9233	0.9072	0.9463	0.9193
Digit spoken	0.9252	0.9598	0.9736	0.9856	0.9832	0.9829
Speaker's accent	0.0940	0.1011	0.0780	0.1094	0.0796	0.1096

Table 5.5: Classification-based scoring of the Audio-MNIST model for prominent categorical attributes. Each score is computed by performing a counterfactual on an instance of the Audio-MNIST validation set.

5.3 North American Right Whale Calls

When evaluating the models trained on the North American Right Whale call data, the method of measuring the models' abduction abilities used on Audio-MNIST is not applicable due to the lack of metadata on individual whales in the dataset. Because of this, the only quantitative evaluation used on the generative models for this dataset is the agreement with a trained whale call classifier on generated observational and

counterfactual data. The resulting agreement scores are shown in Table 5.6. In the observational setting, both ImageCFGen and DeepSCM models achieve high agreement with the whale call classifier, suggesting the association between whale call types and spectrogram features was learned effectively by both models. However, in both the observational and counterfactual setting, the ImageCFGen model achieved wildly different agreement scores across the two training runs completed, dropping from full agreement in the counterfactual setting to less than 40% agreement. This discrepancy in performance is aided by the fine-tuning process, which brings the model to above 90% agreement with the classifier once again.

Generated whale calls from the causal models are shown in Figure 5.10. The calls generated by DeepSCM are blurrier than those generated by ImageCFGen, but also seem to more accurately capture features such as the half-parabola shape of a whale's upcall (bottom right of Figure 4.8). This could be due to the VAE of the DeepSCM requiring a lower learning rate to avoid exploding gradients.

	ImageCFGen		ImageCFGen (ft)		DeepSCM	
Call Type (obs)	0.9736	0.8774	-		1.0	1.0
Call Type (cf)	1.0	0.3898	0.9815	0.9326	0.9868	0.9825

Table 5.6: Classifier agreement with causal generative models on the NARW dataset. Results from both observational (top) and counterfactual (bottom) settings are presented.

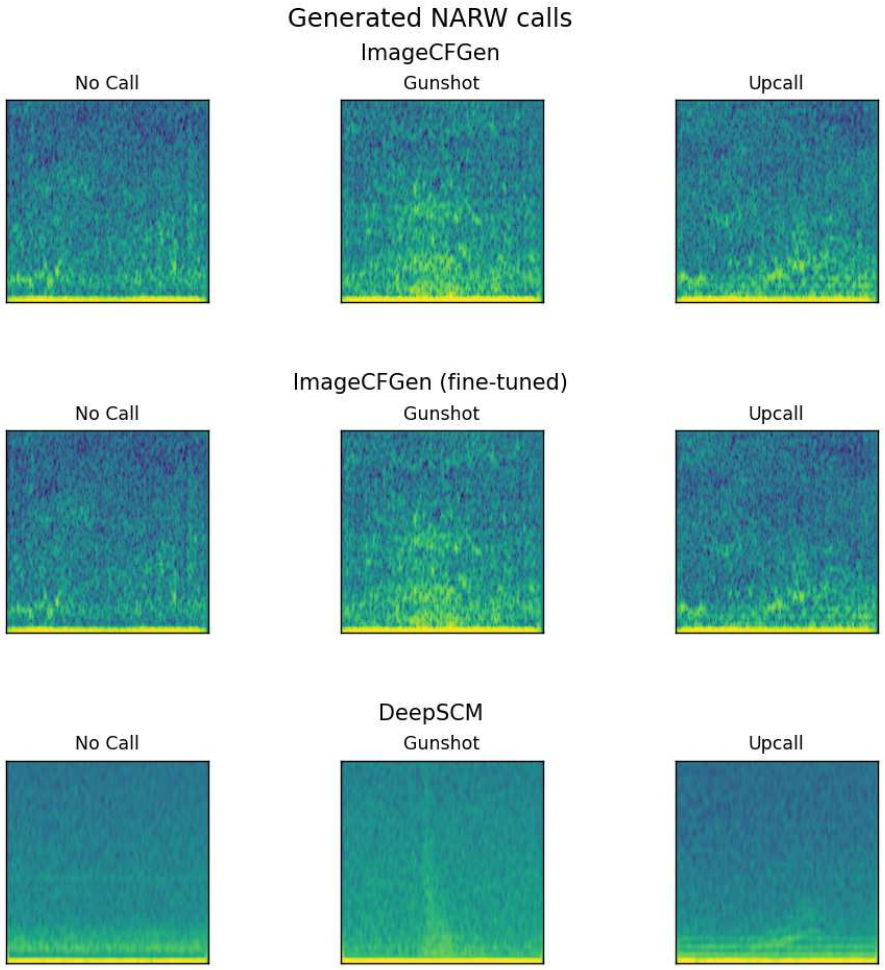


Figure 5.10: North American Right Whale calls generated by ImageCFGen (top), fine-tuned ImageCFGen (middle), and DeepSCM (bottom) models. The three classes of audio are represented in the columns of the figure. Spectrograms were generated by averaging over 4 monte carlo samples using the latent prior.

Chapter 6

Discussion

This thesis has presented two main contributions in the area of counterfactual data generation. First, methods were proposed for the generation of counterfactual explanations of classifiers using both gradient descent and model-agnostic interpolation by utilizing the generative structure of DeepSCM [4] and ImageCFGen [5] models (see section 4.5 and section 5.1). These explanations were compared with the counterfactual [12] and contrastive [14] explainers implemented in the OmnixAI model explanation toolkit [13] using quantitative metrics for evaluating counterfactual explanations [15, 16] on the Morpho-MNIST dataset. Second, two more complex datasets were chosen to train and evaluate the DeepSCM and ImageCFGen models on spectrogram-representation audio data, a speech dataset (Audio-MNIST) and a dataset containing North American Right Whale calls. Two methods of evaluating these models using spectrogram classifiers were proposed. The first method was used to test consistency between the attributes of generated observational and counterfactual data by using agreement with the classifier(s) as a sort of accuracy (see Equation 4.16 and Equation 4.17). The second method of evaluation was unique to Audio-MNIST, and used a speaker classifier to measure the ability of DeepSCM and ImageCFGen to accurately maintain a speakers voice when computing counterfactuals concerning the utterance being performed (see Table 5.3).

A limitation of the speaker classifier metric from subsection 4.4.2 is that it does not directly measure the ability of the generative models to produce data which is not present in the training set. That is, because every subject speaks each digit multiple times, counterfactuals involving changing the digit spoken by any subject will produce data similar to that in the training set. This could be potentially improved by training a model with ablations, removing some digit utterances for given subjects and evaluating the model's ability to produce utterances for those unseen digits for a given subject (while still allowing the subject classifier to be trained on all data).

Limitations also exist when using any classifier-based metric, i.e., either the speaker classifier metric from subsection 4.4.2 or the metrics from subsection 4.4.1 for attribute agreement. Namely, that the metrics measure a change in a discrete outcome (the classification of a generated image). This could potentially lead to overoptimistic metric values in cases where generated images lie close to the classifier's decision boundary. A metric accounting for the uncertainty in the outcome of the classifier using the classifier's class distribution could potentially alleviate this limitation in future work.

When evaluating counterfactual explanations of classifiers, two classes of metrics were chosen. The first, IM1 and IM2 [15], uses class-specific autoencoders to measure the relative distance from a counterfactual example to a target manifold. The second, the oracle score from Hvilshøj et al. [16], uses a second classifier independent of the explanation method to measure how adversarial the changes performed to generate a counterfactual are. In both cases, the metrics used for evaluation were chosen in order to compare the interpretability of the model explanations generated by the methods proposed in this thesis with those from OmnixAI.

In terms of IM1, a clear difference is seen when comparing the gradient-based methods of BiGAN and VAE to the methods from OmnixAI. Specifically, the mean IM1 of the proposed gradient-based methods is less than half of that of either method from OmnixAI (see Figure 5.7). Further, values of IM1 from the model-agnostic method proposed are significantly smaller than the the IM1 recorded from the OmnixAI explainers at the 95% confidence level. All methods proposed in this thesis other than the gradient-based VAE explainer also were significantly better in terms of IM2 than those from OmnixAI at the 95% confidence level (see Figure 5.8), however, this metric in particular has been met with criticism in the literature. Specifically, Schut et al. [41] choose not to take IM2 into account due to its inability to distinguish in-distribution data from out-of-distribution (junk) data.

The results recorded using the oracle score from Hvilshøj et al. favour the explanation methods presented in this work over those from OmnixAI, with all methods proposed in this work significantly outperforming both methods from OmnixAI at the 95% confidence level (see Figure 5.9). As with the IM1 metric, the gradient-based BiGAN and VAE methods achieve the most desirable values of the metric among

all methods considered, while the model-agnostic BiGAN and VAE methods have comparable (but superior) performance to the contrastive explainer from OmnixAI, and the counterfactual explainer from OmnixAI has the worst value of the metric. The results of both IM1 and oracle score suggest that the methods presented in this thesis for the generation of counterfactual explanations of classifiers produce more interpretable explanations than those from OmnixAI.

On the Audio-MNIST dataset, three attributes were considered for the generation of observational data and counterfactuals: digit spoken, sex of speaker, and accent of speaker. Using a spectrogram classifier, the generated data was evaluated for consistency of these attribute values. When generating observational data, consistency of all three attributes is very high, suggesting a high capacity in both the DeepSCM and ImageCFGen models to produce believable spectrograms. In the counterfactual setting, sex and digit counterfactuals again produce high consistency metrics, but the models are unable to produce believable accent counterfactuals. This is likely due to the lack of diversity of accents in the dataset, with several accents having only one speaker. Overall however, the models appear to produce believable counterfactuals and are suitable to this audio dataset. In both the observational and counterfactual setting, the DeepSCM model has an advantage over the ImageCFGen model, potentially due to the VAE of DeepSCM having an explicit reconstruction term in its loss function in the case of a Gaussian prior. However, when the ImageCFGen model is fine-tuned, this discrepancy in performance is no longer observed, suggesting the fine-tuning process proposed by Dash et al. [5] is a valid way of improving counterfactual model performance.

Perhaps the most unique model evaluation method used in this work is the one proposed in subsection 4.4.2. Because measuring consistency of attributes does not measure the ability to perform abduction (the process which separates interventions from counterfactuals), this work proposed a metric for the Audio-MNIST dataset which uses a speaker classifier to measure how well a speaker's voice is maintained when a digit (utterance) counterfactual is computed. If abduction was performed correctly, the speaker's voice should be maintained through a combination of noise variables and attributes, whereas attributes alone (an intervention) may lead to a

speaker with the same sex, accent, etc. but a different voice overall. The experiments in this thesis used this metric to compare ImageCFGen and DeepSCM with interventional models formed by using the generator/decoder of ImageCFGen and DeepSCM models along with noise variables sampled from a latent prior. The results in Table 5.3 clearly indicate that DeepSCM is the only model which can significantly improve over the interventional baseline, increasing the agreement with the speaker classifier from approximately 30% to approximately 60%. This result suggests that on the Audio-MNIST dataset, the DeepSCM model is best suited for counterfactuals which preserve the speaker's voice, and does not require fine-tuning to achieve high levels of agreement with the attribute classifiers.

On the North American Right Whale call dataset, the evaluation offered by a whale call classifier does not actually measure the ability of models to perform abduction, i.e., compute counterfactuals. This means that a model producing interventional data may achieve similar scores to the causal models trained on this dataset. Therefore, the results shown in Table 5.6 are more concerned with the feasibility of training DeepSCM or ImageCFGen models to generate believable whale call data.

The goal of this thesis was to evaluate and compare the abilities of the DeepSCM and ImageCFGen causal generative models to produce accurate counterfactual audio data, as well as to produce counterfactual image explanations of classification models. A metric for measuring the ability of these models to perform abduction was proposed on a human speech dataset, which recorded a significant improvement of DeepSCM over standard generative models such as conditional GANs. While this metric is dataset-specific, the development of such metrics, which can distinguish the performance of causal models from standard generative models, will become more important if the performance of emerging deep causal model architectures are to be trusted. Thus, future work in the area of causal modelling should include the development of new metrics to distinguish the performance of counterfactual-capable models from other similar architectures. The counterfactual explanation methods proposed in this work also demonstrate the ability of causal generative models such as DeepSCM and ImageCFGen to explain classifiers, providing an additional use for such models and further demonstrating the link between causality and explainable AI. Future work in the area of explainable AI could explore new methods of classifier

explanation using causal models, or potentially refine the explainers proposed in this work.

Bibliography

[1] Bernhard Schölkopf and Julius von Kügelgen. From statistical to causal learning, 2022.

[2] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning, 2021.

[3] W. Starr. Counterfactuals. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edition, 2022.

[4] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

[5] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 915–924, 2022.

[6] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[8] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[9] Daniel Coelho de Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-mnist: Quantitative assessment and diagnostics for representation learning. *CoRR*, abs/1809.10780, 2018.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[11] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *CoRR*, abs/1810.09538, 2018.

[12] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018.

[13] Wenzhuo Yang, Hung Le, Tanmay Laud, Silvio Savarese, and Steven C. H. Hoi. Omnixai: A library for explainable ai, 2022.

[14] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives, 2018.

[15] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer, 2021.

[16] Frederik Hvilshøj, Alexandros Iosifidis, and Ira Assent. On quantitative evaluations of counterfactuals. *arXiv preprint arXiv:2111.00177*, 2021.

[17] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[18] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[19] Chris J. Maddison and Danny Tarlow. Gumbel machinery, Jan 2017.

[20] Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

[21] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[22] Juan P Agnelli, M Cadeiras, Esteban G Tabak, Cristina Vilma Turner, and Eric Vanden-Eijnden. Clustering and classification through normalizing flows in feature space. *Multiscale Modeling & Simulation*, 8(5):1784–1802, 2010.

[23] Brian L Trippe and Richard E Turner. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*, 2018.

[24] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

[25] Judea Pearl. *Causality*. Cambridge university press, 2009.

[26] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[29] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[30] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[31] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. A fast griffin-lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

[32] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[33] WAVE File Format — web.archive.org. `https://web.archive.org/web/20080113195252/http://www.borg.com/~jglatt/tech/wave.htm`. [Accessed 03-Mar-2023].

[34] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[35] Javier Nistal, Stefan Lattner, and Gael Richard. Comparing representations for audio synthesis using generative adversarial networks. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 161–165. IEEE, 2021.

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

[37] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[38] Bruno Padovese, Fabio Frazao, Oliver S Kirsebom, and Stan Matwin. Data augmentation for the classification of north atlantic right whales upcalls. *The Journal of the Acoustical Society of America*, 149(4):2520–2530, 2021.

[39] Douglas Michael Gillespie. Detection and classification of right whale calls using an 'edge' detector operating on a smoothed spectrogram. *Canadian Acoustics*, 32(2):39–47, June 2004.

[40] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

[41] Lisa Schut, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *International Conference on Artificial Intelligence and Statistics*, pages 1756–1764. PMLR, 2021.

Appendices

Appendix A

Proof of ImageCFGen Finetuning Objective

For brevity and clarity of notation when referencing vector components, the subscript i on the variables $\mathbf{X}_i, \mathbf{PA}_i$, and $\mathbf{Z}_i$ of the SCM being discussed are dropped in the following proof. To begin the proof, we start from the definition of the latent loss $\mathcal{L}_\mathbf{z}$:

$$\mathcal{L}_\mathbf{z} = \mathbb{E}_{p(\mathbf{Z})}||\mathbf{Z} - E(\mathbf{X}, \mathbf{PA})||^2. \tag{A.1}$$

Assume the components $\mathbf{Z}_j$ of the latent vector $\mathbf{Z} \in \mathbb{R}^d$ are independent random variables with zero mean and finite variances σ_j^2. Taking $||\cdot||^2$ to be the squared L_2 vector norm, we expand the expected value in the definition of $\mathcal{L}_\mathbf{z}$ as follows:

$$\begin{aligned}
\mathcal{L}_\mathbf{z} &= \mathbb{E}_{p(\mathbf{Z})}||\mathbf{Z} - E(\mathbf{X}, \mathbf{PA})||^2 && \text{(A.2)}\\
&= \mathbb{E}_{p(\mathbf{Z})}\left[\sum_{j=1}^{d}(\mathbf{Z}_j - E(\mathbf{X}, \mathbf{PA})_j)^2\right] && \text{(A.3)}\\
&= \sum_{j=1}^{d}\mathbb{E}_{p(\mathbf{Z}_j)}[(\mathbf{Z}_j - E(\mathbf{X}, \mathbf{PA})_j)^2] && \text{(A.4)}\\
&= \sum_{j=1}^{d}\mathbb{E}_{p(\mathbf{Z}_j)}[\mathbf{Z}_j^2 + E(\mathbf{X}, \mathbf{PA})_j^2 - 2\mathbf{Z}_j E(\mathbf{X}, \mathbf{PA})_j]. && \text{(A.5)}
\end{aligned}$$

Next, note that:

$$\mathbb{E}_{p(\mathbf{Z}_j)}[\mathbf{Z}_j^2] = \sigma_j^2 + \mathbb{E}_{p(\mathbf{Z}_j)}[\mathbf{Z}_j]^2 = \sigma_j^2, \tag{A.6}$$

and

$$\mathbb{E}_{p(\mathbf{Z}_j)}[-2\mathbf{Z}_j E(\mathbf{X}, \mathbf{PA})_j] = -2E(\mathbf{X}, \mathbf{PA})_j \mathbb{E}_{p(\mathbf{Z}_j)}[\mathbf{Z}_j] = 0. \tag{A.7}$$

The loss $\mathcal{L}_{\mathbf{z}}$ then becomes:

$$\mathcal{L}_{\mathbf{z}} = \sum_{j=1}^{d} \left(\mathbb{E}_{p(\mathbf{z}_j)}[\mathbf{Z}_j^2] + E(\mathbf{X}, \mathbf{PA})_j^2 + \mathbb{E}_{p(\mathbf{z}_j)}[-2\mathbf{Z}_j E(\mathbf{X}, \mathbf{PA})_j] \right) \tag{A.8}$$

$$= \sum_{j=1}^{d} (\sigma_j^2 + E(\mathbf{X}, \mathbf{PA})_j^2) \tag{A.9}$$

$$= ||E(\mathbf{X}, \mathbf{PA})||^2 + \sum_{j=1}^{d} \sigma_j^2. \tag{A.10}$$

Hence, as all the σ_j^2's are constant, minimizing $\mathcal{L}_{\mathbf{z}}$ under the given assumptions on $p(\mathbf{Z})$ is equivalent to minimizing the norm of latent vectors produced by E.

Appendix B

Morpho-MNIST Visual Classifier Explanations

This appendix contains Morpho-MNIST counterfactual examples for each of the ten classes in the dataset. All figures are displayed in the same way as Figure 5.6, with explanations for each method considered shown alongside the original image. Also as in Figure 5.6, the target class for the causal-model based explainers is taken to be the resulting class of the OmnixAI contrastive explanation. In all cases but one the methods are successful in flipping the classifier's prediction to the target class, with the gradient-based explainers failing in the case of flipping a '5' to a '3' (see Figure B.6). However, even in this case, the gradient explainer still flipped the classifier's prediction to another class.

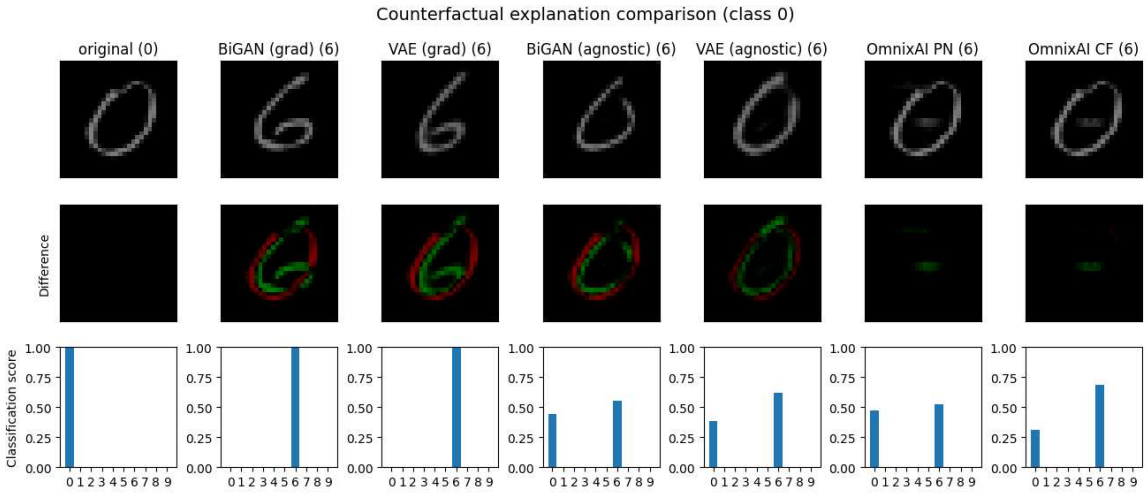


Figure B.1: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 0.

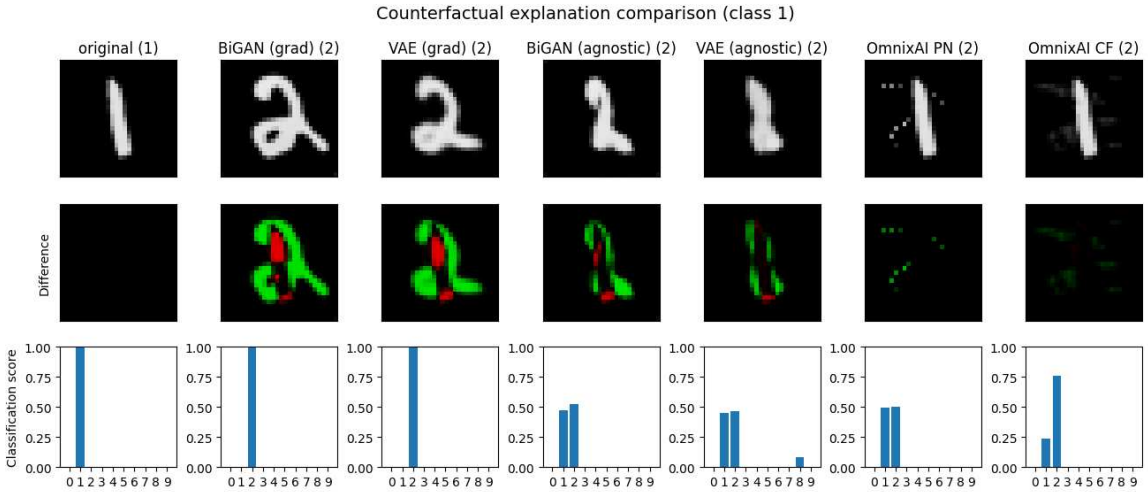


Figure B.2: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 1.

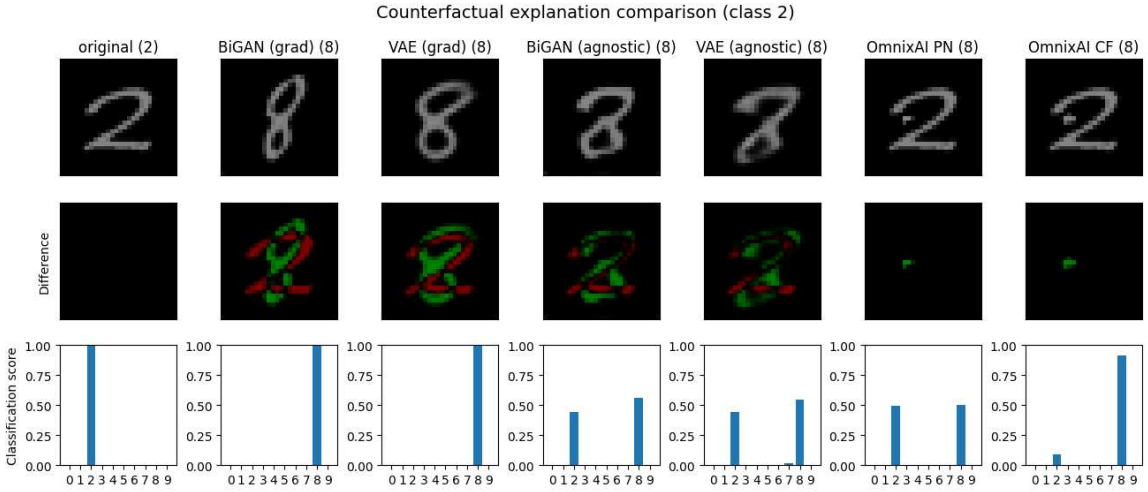


Figure B.3: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 2.

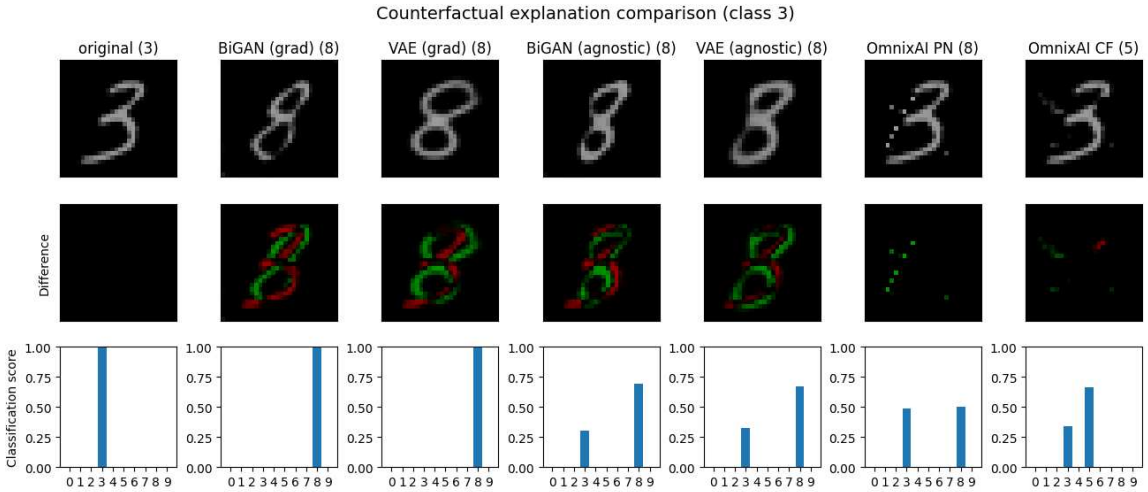


Figure B.4: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 3.

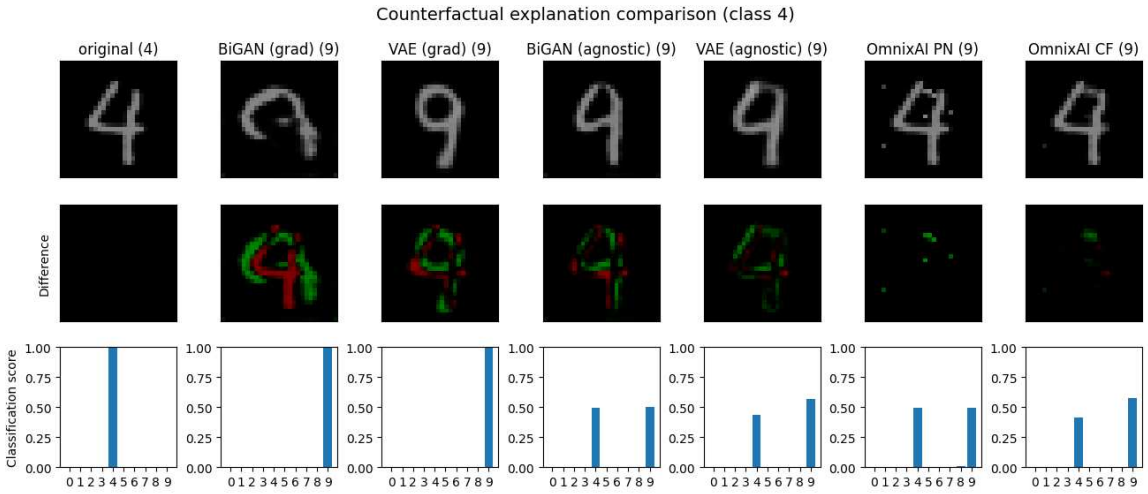


Figure B.5: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 4.

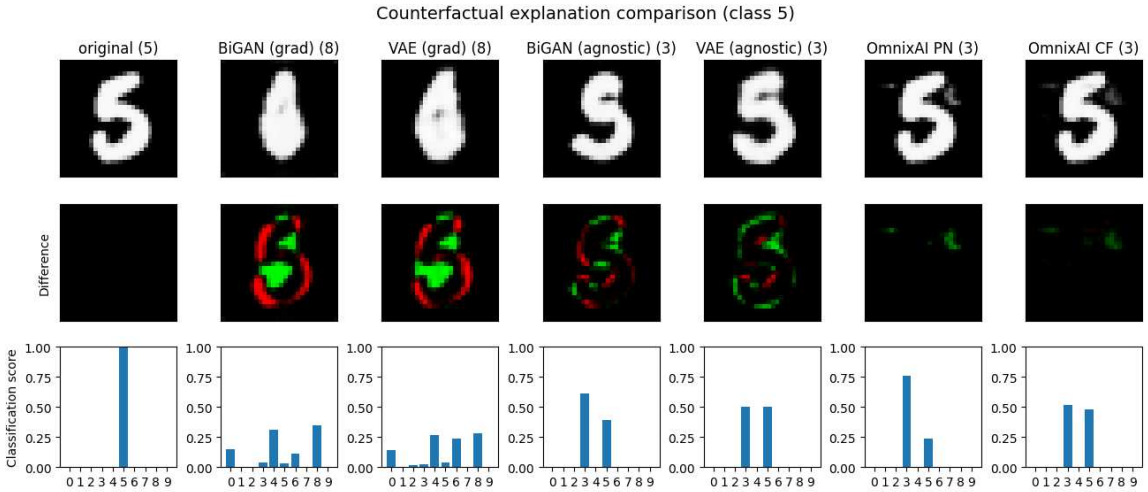


Figure B.6: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 5.

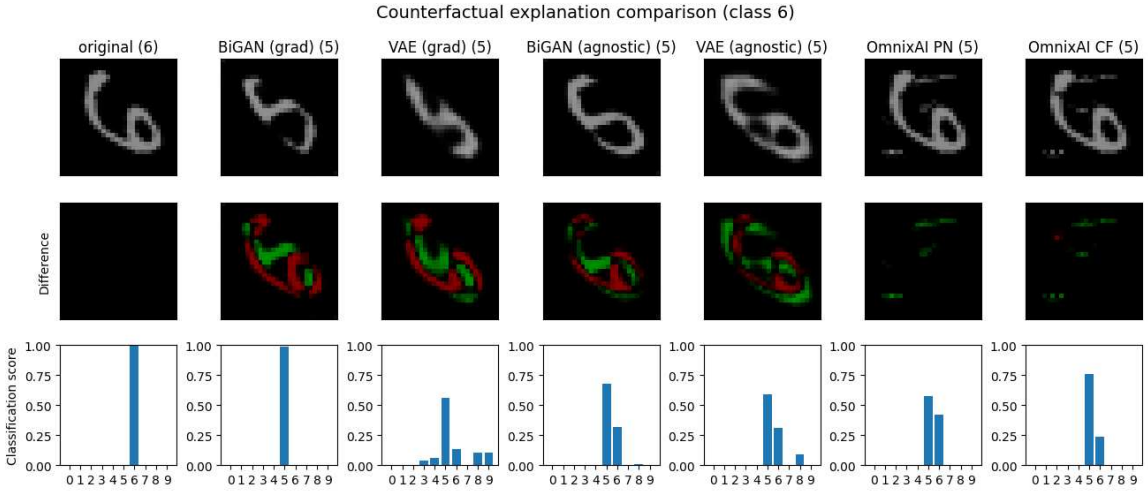


Figure B.7: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 6.

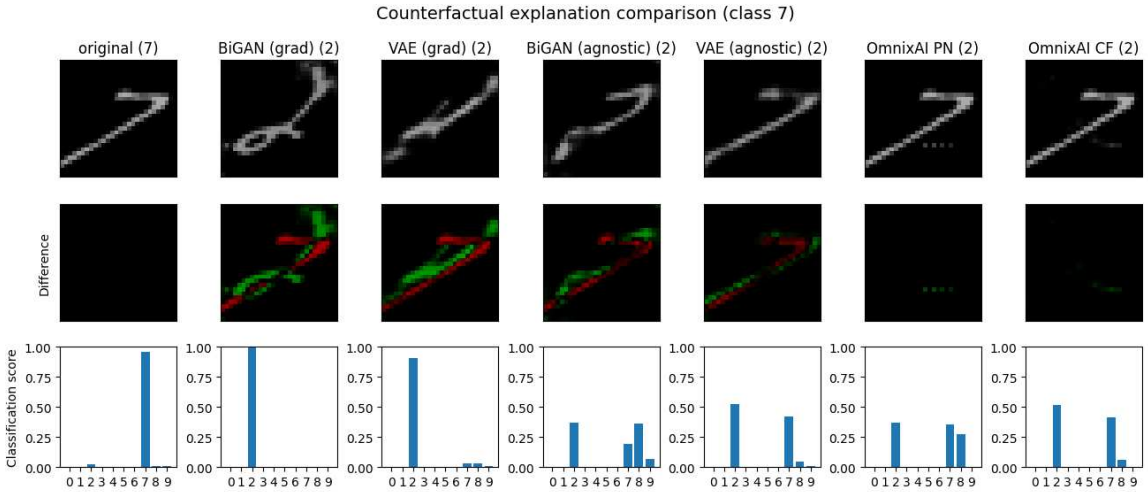


Figure B.8: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 7.

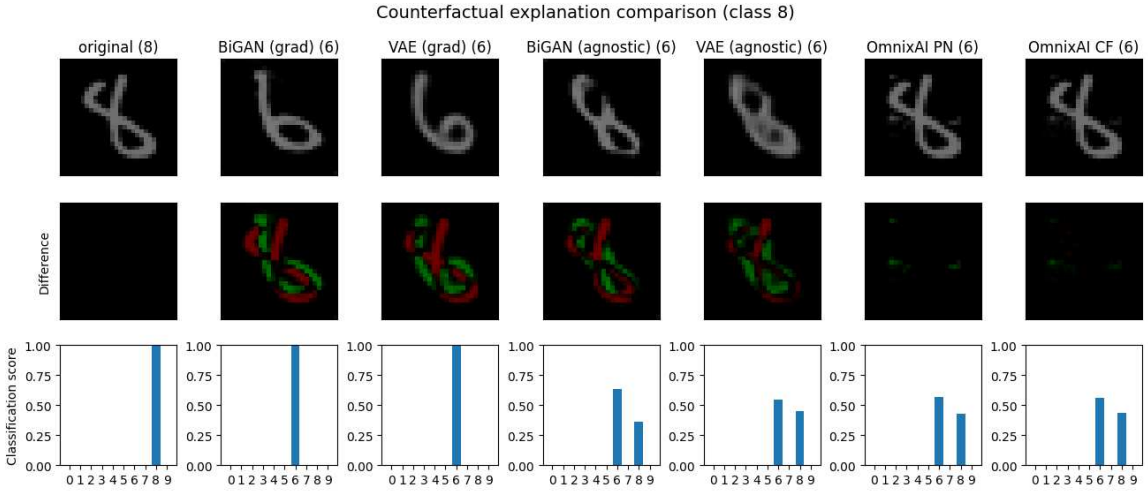


Figure B.9: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 8.

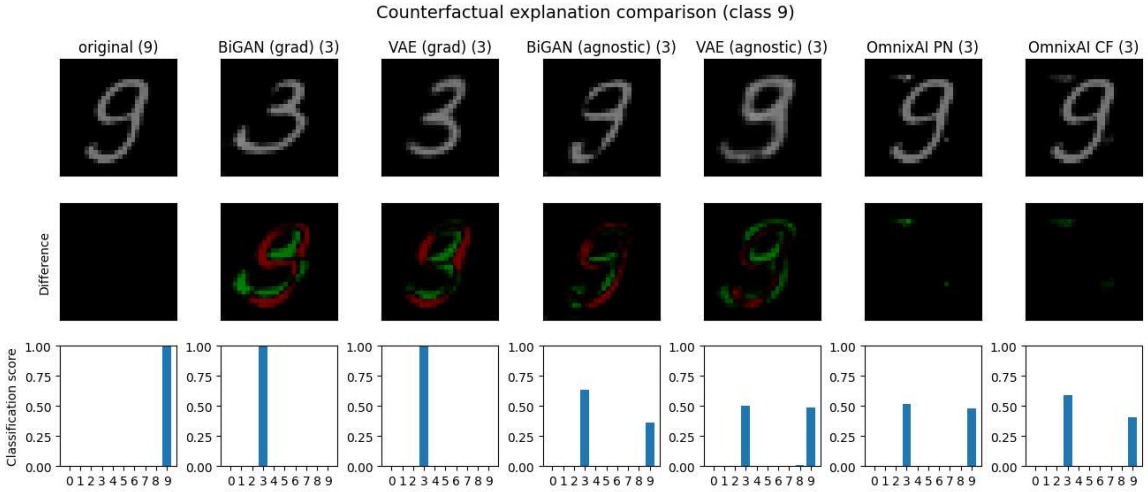


Figure B.10: Classifier explanations and score distributions from the explanation methods considered in this work. The pictured example is a Morpho-MNIST digit from class 9.

Appendix C

Model Convergence Curves

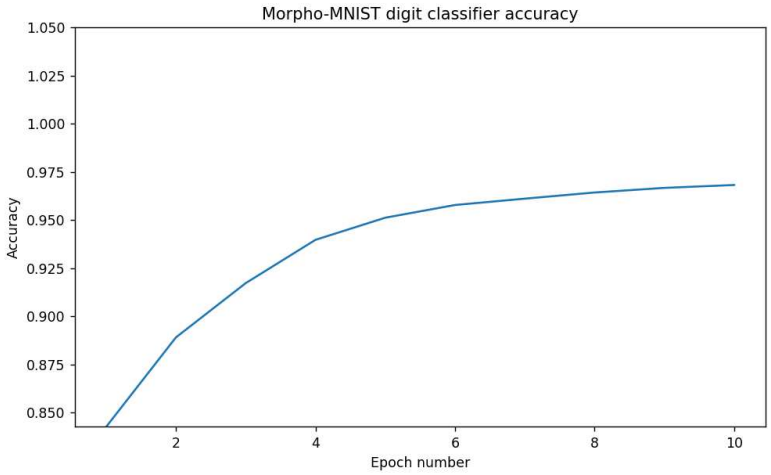


Figure C.1: Validation accuracy during training of classifiers for the Morpho-MNIST dataset used to produce results in section 5.1.

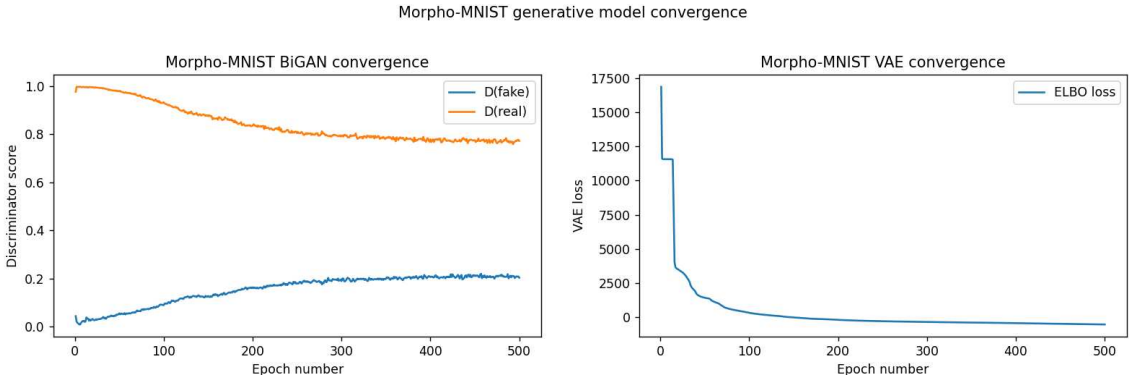


Figure C.2: Convergence of the BiGAN (left) and VAE (right) models trained to compute counterfactuals on the Morpho-MNIST dataset.

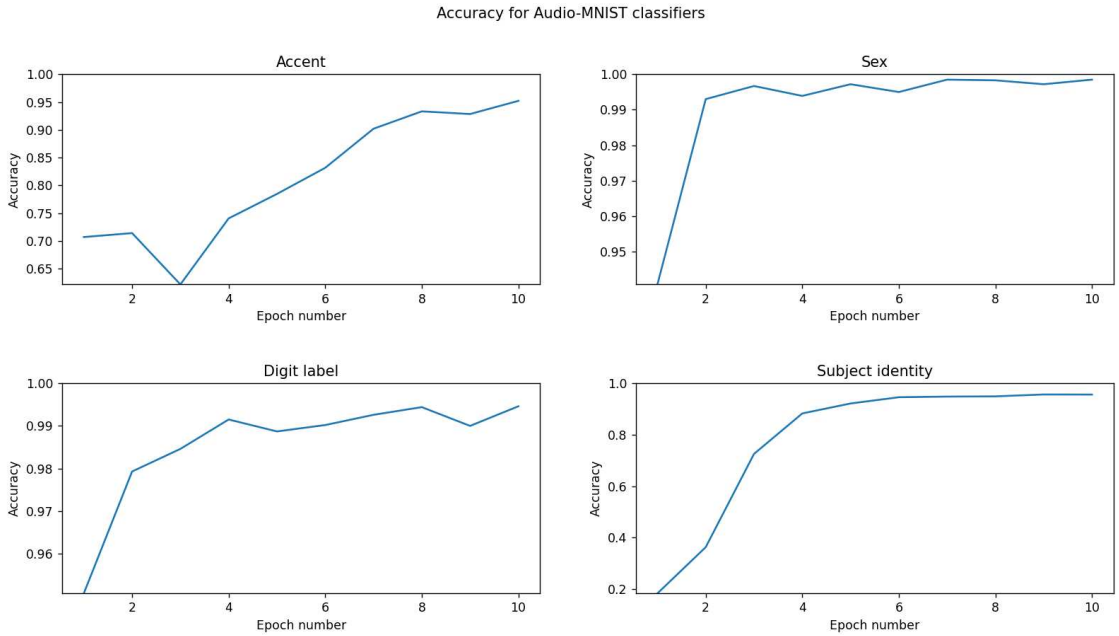


Figure C.3: Validation accuracy during training of classifiers for the Audio-MNIST dataset used to produce results in section 5.2.

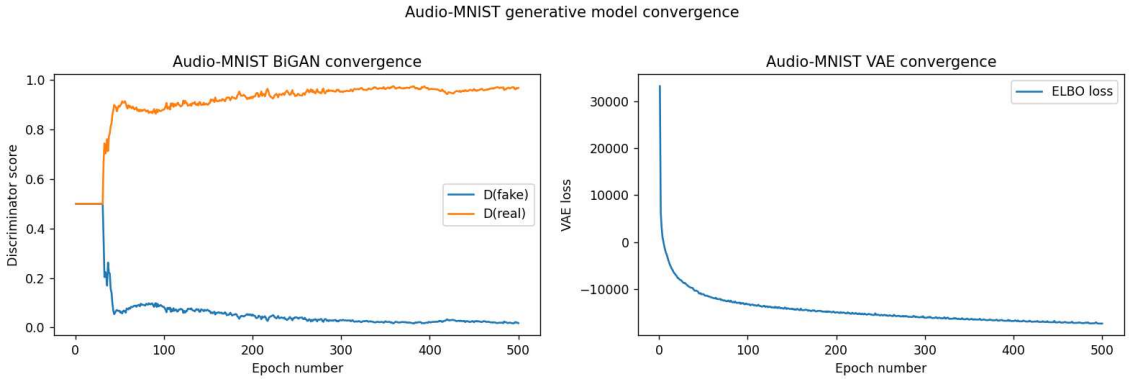


Figure C.4: Convergence of the BiGAN (left) and VAE (right) models trained to compute counterfactuals on the Audio-MNIST dataset.

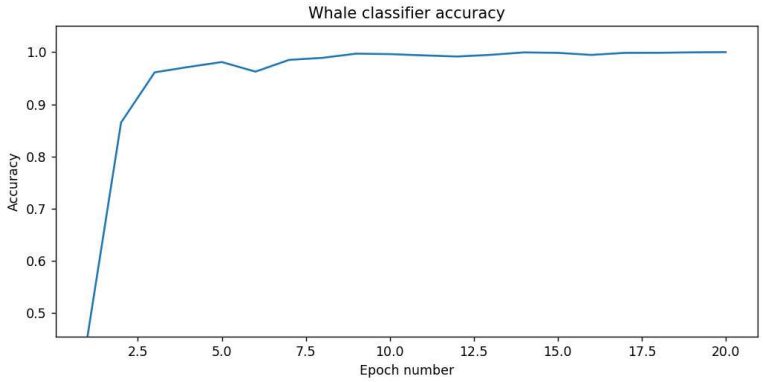


Figure C.5: Validation accuracy during training of the whale call type classifier used to produce results in section 5.3.

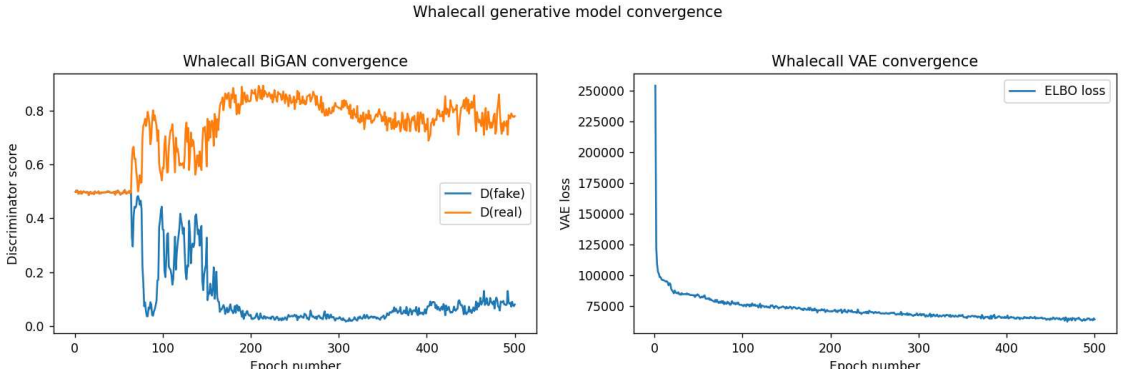


Figure C.6: Convergence of the BiGAN (left) and VAE (right) models trained to compute counterfactuals on the whale call dataset.