DEEP BACKTRACKING COUNTERFACTUALS FOR CAUSALLY COMPLIANT EXPLANATIONS

Klaus-Rudolf Kladny[1] **Julius von Kügelgen**[1, 2] **Bernhard Schölkopf**[1] **Michael Muehlebach**[1]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Department of Engineering, University of Cambridge, United Kingdom
{kkladny, jvk, bs, michaelm}@tue.mpg.de

ABSTRACT

Counterfactuals can offer valuable insights by answering what would have been observed under altered circumstances, conditional on a factual observation. Whereas the classical interventional interpretation of counterfactuals has been studied extensively, *backtracking* constitutes a less studied alternative where all causal laws are kept intact. In the present work, we introduce a practical method for computing backtracking counterfactuals in structural causal models that consist of deep generative components. To this end, we impose conditions on the structural assignments that enable the generation of counterfactuals by solving a tractable constrained optimization problem in the structured latent space of a causal model. Our formulation also facilitates a comparison with methods in the field of counterfactual explanations. Compared to these, our method represents a versatile, modular and causally compliant alternative. We demonstrate these properties experimentally on a modified version of MNIST and CelebA.

1 INTRODUCTION

In recent years, there has been a surge in the use of deep generative models for causal modelling. The integration of deep learning in causal modelling combines the potential to effectively operate on high-dimensional distributions, a strength inherent to deep generative modeling, with the capability to answer inquiries of a causal nature, thus going beyond statistical associations. At the apex of such inquiries lies the ability to generate scenarios of a counterfactual nature—altered worlds where variables differ from their factual realizations, hence aptly termed *counter to fact* (Pearl, 2009; Bareinboim et al., 2022). Counterfactuals are deeply ingrained in human reasoning (Roese, 1997), as evident from phrases such as *"Had it rained, the grass would be greener now"* or *"Had I invested in bitcoin, I would have become rich"*.

Constructing counterfactuals necessitates two fundamental components: (i) a sufficiently accurate world model with mechanistic semantics, such as a structural causal model; and (ii) a sound procedure for deriving the distribution of all variables that are not subject to explicit alteration. The latter component has been a subject of debate: While the classical literature in causality constructs counterfactuals by actively manipulating causal relationships (*interventional counterfactuals*), this approach has been contested by some psychologists and philosophers (Rips, 2010; Gerstenberg et al., 2013; Lucas & Kemp, 2015). Instead, they have proposed an account of counterfactuals where alternate worlds are derived by tracing changes back to background conditions while leaving all causal mechanisms intact. This type of counterfactual is therefore termed *backtracking counterfactual* (Jackson, 1977) and has recently been formalized within the structural causal model framework by von Kügelgen et al. (2023). However, implementing this formalization for deep structural causal models is not straightforward due to multiple computationally intractable steps, such as marginalizations and the evaluation of distributions that are computationally intractable.

The present work addresses these challenges and offers a computationally tractable implementation by framing the generation of counterfactuals as a constrained optimization problem. The optimization is solved with an iterative algorithm, which linearizes the reduced form of the structural causal model. These measures provide effective remedies for generating counterfactual scenarios in multi-

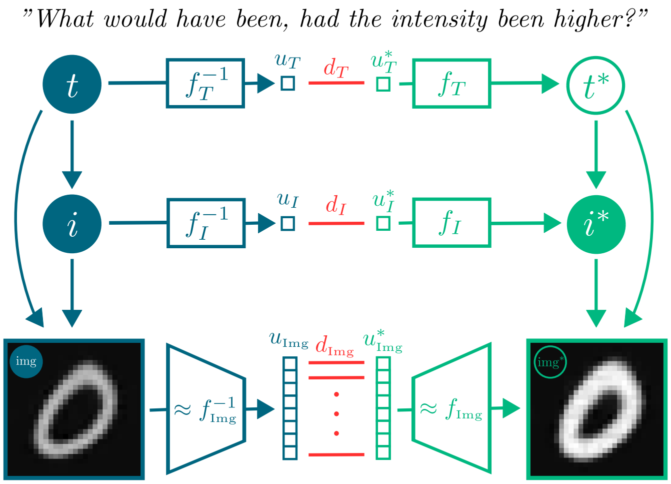

Figure 1: **Visualization of DeepBC for Morpho-MNIST.** We generate a counterfactual (green) image img* and thickness t^* with antecedent intensity i^* for the factual, observable realizations (blue) img, t, i. Our approach finds new latent variables $\mathbf{u}^*$ that minimize a distance d to the factual latents $\mathbf{u}$, subject to rendering the antecedent i^* true. The causal mechanisms in the factual world remain unaltered in the counterfactual world. In this specific distribution, thickness and intensity are positively related, thus rendering the image both more intense and thicker in the counterfactual. Dependence of f_i on graphical parents is omitted for simplifying visual appearance.

variable data with known causal relationships. Furthermore, the present work serves as a bridge between causal modelling and practical methods in the field of counterfactual explanations, which, despite its similar nomenclature, has evolved largely independently from the field of counterfactuals in causality.

We summarize our main contributions as follows:

- We introduce a computationally tractable method called *deep backtracking counterfactuals* (DeepBC) for computing backtracking counterfactuals in deep structural causal models (§ 3). Our method exhibits multiple favorable properties such as versatility, causal compliance and modularity (§ 3.2).
- We show the relation between our method and the field of counterfactual explanations and elucidate how our method can be understood as a general form of the popular method proposed by Wachter et al. (2017) (§ 3.1).
- We demonstrate the applicability and distinct advantages of our method through experiments on two data sets, in comparison to existing methods. Specifically, we apply our method to Morpho-MNIST and the CelebA data set (§ 4).

Overview. Section § 2 introduces important concepts such as structural causal models (§ 2.1), the deep generative models that are employed subsequently (§ 2.2), interventional and backtracking counterfactuals (§ 2.3) and counterfactual explanations (§ 2.4). In Section § 3, we propose our method called *deep backtracking counterfactuals* (DeepBC) and discuss its relation to methods in the field of counterfactual explanations (§ 3.1) and its implementation (§ 3.3). In Section § 4, we perform experiments on Morpho-MNIST (§ 4.1) and CelebA (§ 4.2) that highlight the versatility, modularity and causal compliance of our method. We then discuss the limitations of our work in § 5 and conclude with a short summary in § 6. Related work is included in App. D.

2 Setting & Preliminaries

The following section introduces important concepts, such as structural causal models and backtracking counterfactuals, which sets the stage for introducing our method in § 3.

Notation. Upper case X denotes a scalar or multivariate continuous random variable, and lower case x a realization thereof. Bold $\mathbf{X}$ denotes a collection of such random variables with realizations $\mathbf{x}$. The components of $\mathbf{x}$ will be denoted by x_i. We denote the probability density of X by $p(x)$.

2.1 Structural Causal Models

Let $\mathbf{X} = (X_1, X_2, ..., X_n)$ be a collection of potentially high-dimensional observable "endogenous" random variables. For instance, X_1 could be a high-dimensional object such as an image and X_2 a scalar feature variable. The causal relationships among the X_i are specified by a directed acyclic graph G that is known. A structural causal model is characterized by a collection of structural equations $X_i \leftarrow f_i(\mathbf{X}_{\text{pa}(i)}, U_i)$, for $i = 1, 2, ..., n$, where $\mathbf{X}_{\text{pa}(i)}$ are the causal parents of X_i as specified by G and $\mathbf{U} = (U_1, U_2, ..., U_n)$ are "exogenous" latent variables. The acyclicity of G ensures that for all i, we can recursively solve for X_i to obtain a deterministic expression in terms of $\mathbf{U}$. Thus, there exists a unique function that maps $\mathbf{U}$ to $\mathbf{X}$, which we denote by $\mathbf{F}$:

$$\mathbf{X} = \mathbf{F}(\mathbf{U}), \tag{1}$$

and is known as the reduced-form expression. Hence, $\mathbf{F}$ induces a distribution over observables $\mathbf{X}$, for any given distribution over the latents $\mathbf{U}$. For the remainder of this work, we assume causal sufficiency (Spirtes, 2010) (no unobserved confounders), which implies joint independence of the components of $\mathbf{U}$.

2.2 Deep Invertible Structural Causal Models

In this work, we make the simplifying assumption that $f_i(\mathbf{x}_{\text{pa}(i)}, \cdot\,)$ is invertible for any fixed $\mathbf{x}_{\text{pa}(i)}$, such that we can write

$$U_i = f_i^{-1}(\mathbf{X}_{\text{pa}(i)}, X_i), \quad i = 1, 2, ..., n.$$

Under this assumption, the inverse $\mathbf{F}^{-1}$ of the mapping in (1) is guaranteed to exist, and we can write

$$\mathbf{U} = \mathbf{F}^{-1}(\mathbf{X}).$$

We assume that all f_i are given as (conditional) deep generative models, trained separately for each structural assignment (Pawlowski et al., 2020). We consider the following two classes of models, both of which operate on latent variables with a Gaussian prior.

Conditional normalizing flows (Rezende & Mohamed, 2015; Winkler et al., 2019) are constructed as a composition of invertible functions, hence rendering the entire function f_i invertible in u_i. In addition, they are chosen such that the determinant of the Jacobian can be compted efficiently. These two attributes facilitate efficient training of f_i via maximum likelihood.

Conditional variational auto-encoders (Kingma & Welling, 2014; Sohn et al., 2015) consist of separate encoder e_i and decoder d_i networks. These modules parameterize the mean of their respective conditional distributions, i.e., $U_i|\mathbf{x}_{\text{pa}(i)}, x_i \sim \mathcal{N}(e_i(\mathbf{x}_{\text{pa}(i)}, x_i), \text{diag}(\boldsymbol{\sigma}_e^2))$ and $X_i|\mathbf{x}_{\text{pa}(i)}, u_i \sim \mathcal{N}(d_i(\mathbf{x}_{\text{pa}(i)}, u_i), \mathbf{I}\sigma_d^2)$. Through joint training of e_i, d_i and variance vector $\boldsymbol{\sigma}_e^2$ using variational inference, e_i and d_i become interconnected. Theoretical insights by Reizinger et al. (2022) support the use of an approximation, where the decoder effectively inverts the encoder, that is,

$$x_i = f_i(\mathbf{x}_{\text{pa}(i)}, f_i^{-1}(\mathbf{x}_{\text{pa}(i)}, x_i)) \approx d_i(\mathbf{x}_{\text{pa}(i)}, e_i(\mathbf{x}_{\text{pa}(i)}, x_i)).$$

2.3 Interventional and Backtracking Counterfactuals

Given a factual observation $\mathbf{x}$ and a so-called antecedent $\mathbf{x}_S^* = (x_i^* : i \in S)$ for a given subset $S \subset \{1, 2, ..., n\}$, we define a counterfactual as some $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)$ consistent with $\mathbf{x}_S^*$. We view $\mathbf{x}^*$ as an answer to the verbal query *"What values $\mathbf{x}^*$ had $\mathbf{X}$ taken instead of the given (observed) $\mathbf{x}$, had $\mathbf{X}_S$ taken the values $\mathbf{x}_S^*$ rather than $\mathbf{x}_S$?"*. In the present work, we consider interventional and backtracking counterfactuals. Both generate distributions over counterfactuals whose random variables we refer to as $\mathbf{X}^*$. We only provide a conceptual notion and refer the reader to App. A.1 for a more rigorous formalism for both types of counterfactuals.

Interventional counterfactuals render the antecedent true via modification of the structural assignments $(f_1, f_2, ..., f_n)$, which leads to a new collection of assignments $(f_1^*, f_2^*, ..., f_n^*)$. Specifically, these new structural assignments are constructed such that the causal dependence on the causal parents of all antecedent variables $\mathbf{X}_S^*$ is removed: $f_i^* = x_i^*$ for $i \in S$ and $f_i^* = f_i$ otherwise. Such a modification can be understood as a *hard intervention* on the underlying structural relations.

Backtracking counterfactuals leave all structural assignments unchanged. In order to set the antecedent $\mathbf{x}_S^* \neq \mathbf{x}_S$ true, they trace differences to the factual realization back to (ideally small)

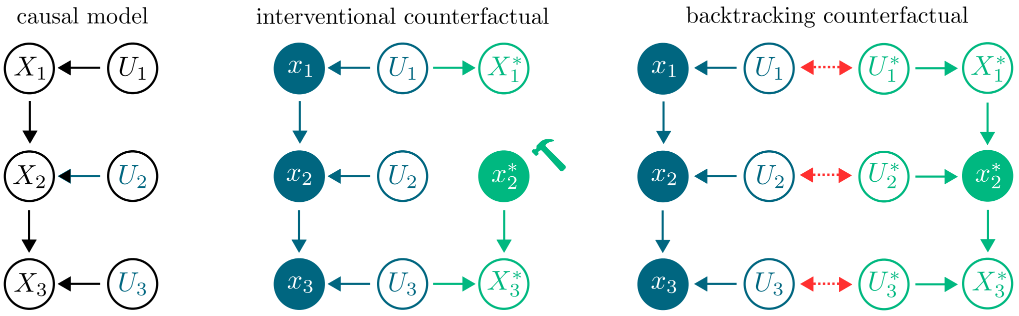
causal model
interventional counterfactual
backtracking counterfactual
X_1 U_1
X_2 U_2
X_3 U_3
x_1 U_1 X_1^*
x_2 U_2 x_2^*
x_3 U_3 X_3^*
x_1 U_1 U_1^* X_1^*
x_2 U_2 U_2^* x_2^*
x_3 U_3 U_3^* X_3^*

Figure 2: **Difference between interventional and backtracking counterfactuals on an example.** Variables that are conditioned on correspond to filled circles. Interventional counterfactuals perform a hard intervention (indicated by a hammer) $X_2^* \leftarrow x_2^*$ with antecedent x_2^* (i.e., $S = \{2\}$) in the counterfactual world (green). Backtracking counterfactuals, on the contrary, construct this counterfactual world via introducing a new set of latent variables $\mathbf{U}^*$ that depend on $\mathbf{U}$ via a backtracking conditional (red).

changes in the latent variables $\mathbf{U}$. These modified latent variables are represented by a new collection of variables $\mathbf{U}^*$ that depend on $\mathbf{U}$ via a backtracking conditional $p(\mathbf{u}^*|\mathbf{u})$ (von Kügelgen et al., 2023), which represents a probability density for computing similarity between $\mathbf{u}$ and $\mathbf{u}^*$ and which we assume to be decomposable, or factorized: $p(\mathbf{u}^* \mid \mathbf{u}) = \prod_{i=1}^{n} p(u_i^* \mid u_i)$. By marginalizing over $\mathbf{U}^*$, we obtain the distribution of $\mathbf{X}^* \mid \mathbf{x}_S^*, \mathbf{x}$.

Section § 2 concludes by introducing so-called counterfactual explanations. This allows us to compare our method against this formulation in Section § 3.1.

2.4 Counterfactual Explanations

A wealth of prior work in machine learning is concerned about explaining the prediction $\hat{y}$ of a classifier $f_{\hat{Y}}$ with $\hat{y} \leftarrow f_{\hat{Y}}(x)$ through the generation of a new example x^* which is close to x, yet predicted as y^*, where y^* is a label that differs from the (factual) prediction $\hat{y}$ [1]. The intuitive idea is that contrasting x^* with x yields an interpretable answer as to why x is classified as $\hat{y}$ rather than y^*. Formally (see Wachter et al. (2017)), x^* can be obtained as the solution of

$$\arg\min_{x'} d_o\left(x',\ x\right) \quad \text{subject to} \quad f_{\hat{Y}}(x') \ = \ y^*, \tag{2}$$

where d_o represents a distance function between observed variables.

3 Deep Backtracking Counterfactuals (DeepBC)

In this work, we propose to generate a counterfactual example $\mathbf{x}^*$ for the factual realization $\mathbf{x}$ as a solution to the following constrained optimization problem:

$$\arg\min_{\mathbf{x}'} \sum_{i=1}^{n} d\left(\mathbf{F}_i^{-1}(\mathbf{x}'),\ \mathbf{F}_i^{-1}(\mathbf{x})\right) \qquad \text{subject to} \qquad \mathbf{x}'_S \ = \ \mathbf{x}^*_S, \tag{3}$$

where d denotes a differentiable distance function. Intuitively, we can understand this optimization as finding a solution $\mathbf{x}^*$ that is *close* to the factual realization $\mathbf{x}$ in terms of its latent components, while fulfilling the constraint that $\mathbf{x}^*$ is compliant both with the antecedent $\mathbf{x}_S^*$ and with the causal laws. This situation is visualized on the Morpho-MNIST example in Fig. 1. We further note that (3) is equivalent to an optimization problem within the latent space, i.e.:

$$\arg\min_{\mathbf{u}'} \sum_{i=1}^{n} d\left(u_i',\ u_i\right) \quad \text{subject to} \quad \mathbf{F}_S(\mathbf{u}') \ = \ \mathbf{x}_S^*,\ \mathbf{u} = \mathbf{F}(\mathbf{x}). \tag{4}$$

We obtain the solution of (3) by inserting the solution of (4) into $\mathbf{F}$. In App. A.2, we provide a derivation of DeepBC from the formalization given by von Kügelgen et al. (2023) .

[1] We stress that $\hat{Y}$ is the prediction of a model and thus an effect of X. In general, $\hat{Y}$ does not agree with Y, since Y might not be the cause of X or might be confounded with X.

3.1 Relation to Counterfactual Explanations

We can recover counterfactual explanations § 2.4 as a special form of DeepBC. To this end, we assume access to two variables with the following structural equations

$$X \leftarrow f_X(U_X) \quad \text{and} \quad \hat{Y} \leftarrow f_{\hat{Y}}(X), \tag{5}$$

where we note that $\hat{Y}$ is not subject to additional randomness $U_{\hat{Y}}$. In this specific case, we observe that the DeepBC optimization problem (3) reduces to

$$\arg\min_{x'} d\left(f_X^{-1}(x'),\ f_X^{-1}(x)\right) \quad \text{subject to} \quad f_{\hat{Y}}(x') \ = \ y^*, \tag{6}$$

which can be interpreted as an instance of (2), where distance is measured in an unstructured latent space (as implemented by, e.g., Jacob et al. (2022); Rodríguez et al. (2021)). From this viewpoint, we can interpret DeepBC as a general form of counterfactual explanations in two ways: Firstly, it accommodates non-deterministic relations among variables, taking into account the influence of noise on all variables. In the aforementioned instance (5), this can be modeled by $Y \leftarrow f_Y(X, U_Y)$. Secondly, DeepBC can account for multiple variables with complex causal relationships. For example, there could be a third variable Z related to X and Y in (6) that could be modeled as well.

3.2 Methodological Contributions

We highlight the main contributions of our work in the context of counterfactual explanations, which we demonstrate experimentally in § 4:

1. **Versatility.** DeepBC naturally supports complex causal relationship between multiple variables that are potentially high dimensional (e.g., images or scalar attributes), which goes beyond the instance-label setup (5) presented in § 3.1, and supports flexible choices of antecedent variables. Further, it allows for varying the distance functions d in (3) to obtain counterfactuals with different properties, such as sparsity. This property means that only few components of $\mathbf{x}^*$ differ from $\mathbf{x}$. The approach contrasts prior work (Wachter et al., 2017; Mothilal et al., 2020; Lang et al., 2022), which introduced a different sparsity measure that may violate causal relationships (see § 4.2).

2. **Causal Compliance.** A plethora of work has discussed the right choice of distance function between data points for generating counterfactual explanations (see, e.g., Guidotti, 2022). In this context, DeepBC offers a causally compliant solution: Rather than defining similarity directly between observable variables that can lead to violations of causal laws, DeepBC delineates similarity in terms of latent variables, embedded into a causal model. This implies that generated counterfactual explanations are guaranteed to preserve causal relationships since the counterfactual variables are always subject to the causal laws of the factual world.

3. **Modularity.** Structural relations between variables $(f_1, f_2, ..., f_n)$ exhibit disparities across distinct domains. It has been postulated that these disparities tend to manifest sparsely, signifying that many modules f_i demonstrate analogous behavior across different domains (Schölkopf et al., 2021; Perry et al., 2022). Leveraging the explicit incorporation of structural equations, DeepBC offers adaptability to new domains through the straightforward substitution of individual components f_i, without the need for relearning the remaining modules. This contrasts with counterfactual explanation methods, which do not incorporate such replaceable modules and thus require relearning of the entire model to handle a domain shift.

3.3 Algorithms

We rely on a penalty formulation to approximate (4), leading to an unconstrained optimization problem. Specifically, we aim at minimizing the following objective function with respect to $\mathbf{u}'$:

$$\mathcal{L}(\mathbf{u}';\, \mathbf{u}, \mathbf{x}_S^*) \ \coloneqq \ \sum_{i=1}^{n} d(u_i',\ u_i) \ + \ \lambda \left\|\mathbf{F}_S(\mathbf{u}') - \mathbf{x}_S^*\right\|_2^2, \tag{7}$$

where $\lambda > 0$ is a sufficiently large penalty parameter and $\mathbf{u} = \mathbf{F}(\mathbf{x})$.

DeepBC via Constraint Linearization. Rather than minimizing (7) via gradient descent, we empirically observe that employing the first-order Taylor approximation of $\mathbf{F}_S$ at $\bar{\mathbf{u}}$ is beneficial, when minimizing the distance $d(u'_i, \bar{u}_i) = \|u'_i - \bar{u}_i\|_2^2$, i.e.,

$$\mathbf{F}_S(\mathbf{u}') \approx \mathbf{F}_S(\bar{\mathbf{u}}) + \mathbf{J}_S(\bar{\mathbf{u}})(\mathbf{u}' - \bar{\mathbf{u}}),$$

where $\mathbf{J}_S(\bar{\mathbf{u}}) := \nabla_{\mathbf{u}} \mathbf{F}_S(\bar{\mathbf{u}})$ denotes the Jacobian matrix. As a result of this approximation, (7) is a convex quadratic function in $\mathbf{u}'$ and can therefore be solved for its minimum $\hat{\mathbf{u}}^*$ in closed form:

$$\hat{\mathbf{u}}^* \;=\; (\mathbf{I} + \lambda \mathbf{J}_S^\top(\bar{\mathbf{u}}) \mathbf{J}_S(\bar{\mathbf{u}}))^{-1} (\mathbf{u} + \lambda \mathbf{J}_S^\top(\bar{\mathbf{u}}) \tilde{\mathbf{x}}_S^*), \tag{8}$$

where $\tilde{\mathbf{x}}_S^* = \mathbf{x}_S^* + \mathbf{J}_S(\bar{\mathbf{u}})\bar{\mathbf{u}} - \mathbf{F}_S(\bar{\mathbf{u}})$. A detailed derivation of (8) is provided in App. A.3.

Solving (8) once, starting from the initial condition $\mathbf{u}'_0 = \mathbf{u}$, does not accurately fulfill the constraint due to the constraint linearization, except for special cases. We thus apply an iterative algorithm similar to Newton's method, based on (8) that is specified in Alg. 1. Empirically, we observe Alg. 1 to converge much faster than gradient descent and to be more robust with respect to non-linear $\mathbf{F}$.

Algorithm 1 DeepBC via Constraint Linearization

```
$\mathbf{u}'_0 \leftarrow \mathbf{u}$
for $t = 1, 2, ..., \#\text{it}$ do
    $\bar{\mathbf{J}}_S \leftarrow \mathbf{J}_S(\mathbf{u}'_{t-1})$
    $\tilde{\mathbf{x}}_S^* \leftarrow \mathbf{x}_S^* + \bar{\mathbf{J}}_S \mathbf{u}'_{t-1} - \mathbf{F}_S(\mathbf{u}'_{t-1})$
    $\mathbf{u}'_t \leftarrow (\mathbf{I} + \lambda \bar{\mathbf{J}}_S^\top \bar{\mathbf{J}}_S)^{-1} (\mathbf{u} + \lambda \bar{\mathbf{J}}_S^\top \tilde{\mathbf{x}}_S^*)$
end for
```

Sparse DeepBC. We further employ a variant of DeepBC that encourages sparse solutions, where sparsity is measured in $\mathbf{u}$ rather than $\mathbf{x}$. Specifically, we use sparse DeepBC to obtain solutions where only few elements in $\mathbf{u}^*$ differ from $\mathbf{u}$, i.e., $d(u'_i, u_i) = \|u'_i - u_i\|_0$, where $\|\cdot\|_0$ denotes the number of nonzero elements. We apply a greedy approach similar to Mothilal et al. (2020), where we start by fixing an integer $M > 0$ for which we desire that $\|\mathbf{u}' - \mathbf{u}\|_0 \leq M$. We then apply an optimization twice: In a first step, we solve for $\mathbf{u}^*$ using DeepBC. Then, we use the M elements of the solution vector with largest $\|u_i - u_i^*\|_2$ and apply DeepBC again only on these elements, while fixing the others to u_i.[2]

4 Experiments

We run experiments as to contrast DeepBC to existing ideas and showcase its properties and abilities as outlined in § 3.1. For all experiments, we use `PyTorch` (Paszke et al., 2019), `PyTorch Lightning` (Falcon, William and The PyTorch Lightning team, 2019) and `normflows` (Stimper et al., 2023). We use DeepBC via constraint linearization (Alg. 1) with $\lambda = 10^4$ and $\#\text{it} = 30$.

4.1 Morpho-MNIST

Experimental Setup. We use Morpho-MNIST, a modified version of MNIST proposed by Castro et al. (2019), to showcase how deep backtracking contrasts with its interventional counterpart (Pawlowski et al., 2020). The data set consists of three variables, two scalars and an MNIST image. The first scalar variable T describes thickness, whereas the second variable I describes intensity. They have a non-linear relationship and are positively correlated, as can be seen in Fig. 4 **(b)** and **(c)**, where the observational density of thickness and intensity is shown in blue. The known causal relationship between thickness and intensity is depicted in Fig. 4 and we show the true structural equations in App. B. We first train a normalizing flow for thickness and one for intensity (conditionally on thickness) and model the image via a conditional β-VAE (Higgins et al., 2017). We provide more details about the employed architectures in App. A.5.1. We use $d(u'_i, u_i) = \|u'_i - u_i\|_2^2$ as the distance function for DeepBC.

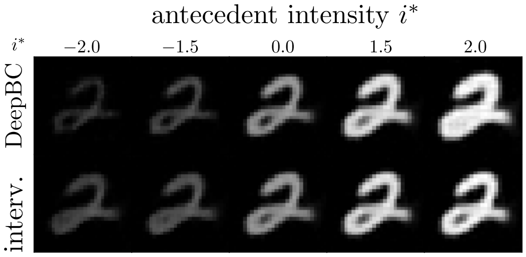

Figure 3: **Counterfactual Images**. DeepBC (top row) changes intensity alongside thickness, since their causal relation is preserved. Interventional counterfactuals (bottom row), on the contrary, solely change the intensity value.

[2] We do not need to weigh $\|u_i - u_i^*\|_2$ by standard deviation/mean absolute distance, since all u_i have the same distribution, see § 2.2.

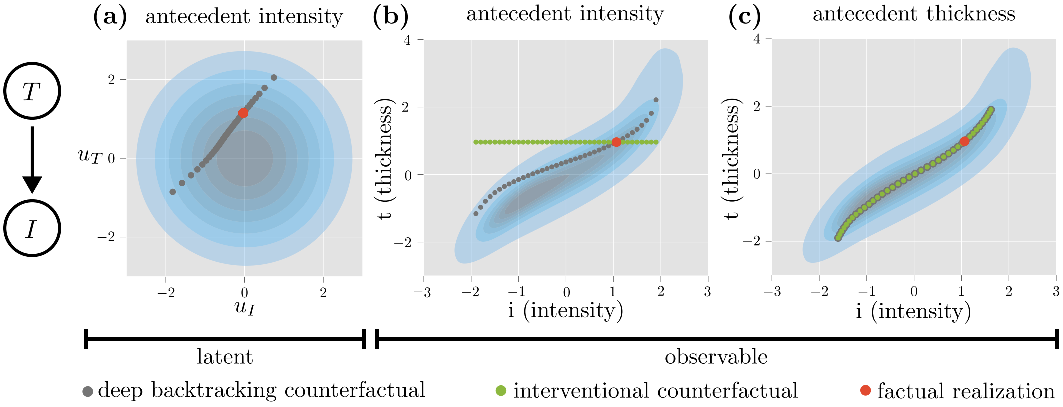

Figure 4: The blue shaded areas indicate probability density on the observed data set and we note that both thickness and intensity variables are causes of the image pixels. **(a)** For varying values of the antecedent i^*, both u_I^* and upstream variable u_T^* change (latent variables). Only the deep backtracking solution is shown. **(b)** Interventional counterfactuals, in contrast to backtracking counterfactuals, leave t^* unchanged for antecendent intensity. **(c)** For antecedent thickness, counterfactual and backtracking counterfactuals are identical.

Results. Our experiments illustrate distinctive properties of the backtracking approach. For choosing intensity as the antecedent, backtracking preserves causal laws and thus changes thickness in accordance with the change in intensity, creating counterfactuals that resemble the images in the data set (see Dominguez-Olmedo et al. (2023)), where thickness and intensity change simultaneously. This is in contrast to the interventional approach, which always leaves thickness unchanged by breaking the causal relationship between both variables. Thus, it generates examples in low density regions, which can be considered a weakness in regard of generating realistic counterfactuals. We show the generated images in Fig. 3. DeepBC arrives at these counterfactuals, since $i^* \neq i$ can either be achieved by choosing a different $u_I^* \neq u_I$ or by changing the upstream $u_T^* \neq u_T$. This is true because i^* also depends on the realization t^*, which, in turn, depends on u_T^*. As to minimize the sum of squares $d(u_T, u_T^*) + d(u_I, u_I^*)$, DeepBC dissociates both latent variables from their factual realizations, as can be seen in Fig. 4 **(a)**. This entails that the upstream t^* diverges from t, which lies in stark contrast to the interventional approach that always keeps upstream variables unmodified (see the bottom row in Fig. 3 and the green dots in Fig. 4 **(b)**).

However, interventional and deep backtracking counterfactuals can also be identical, as visible in Fig. 4 (**c**), where the thickness variable T is used as antecedent. If the antecedent is a root node of the causal graph G, which is the case for T, the change in $t^* \neq t$ cannot be traced back to any latent variable other than u_T, which is why both $u_I^* = u_I$ and $u^*_{\text{Img}} = u_{\text{Img}}$, analogously to interventional counterfactuals. The change in the value i^* as a function of t^* then solely corresponds to the causal effect of t^*, for both counterfactuals (Fig. 4 (**c**)).

4.2 CelebA

Experimental Setup. We generate counterfactual celebrity images on the CelebA data set (Liu et al., 2015) with a resolution of 128 × 128 using binary attributes with the causal graph as assumed by Yang et al. (2021). The causal graph is shown in Fig. 5 (**a**). Our optimization algorithms assume differentiability of $\mathbf{F}$ in $\mathbf{u}$ (§ 3.3), which is why we preprocess the data to use the standardized logits of classifiers that were trained to predict each attribute from the given image. Then, analogously to § 4.1, we train a conditional normalizing flow for each attribute and a conditional β-VAE for the image. A detailed description of the architectures is included in App. A.5.2.

Baselines & Ablations. 1) Measuring distance in $\mathbf{x}$: Prior work has measured distance directly in terms of the observable $\mathbf{x}$ rather than latent variables $\mathbf{u}$ that are embedded into a causal model. For the sake of demonstration, we use a method that encourages sparse solutions in terms of $\mathbf{x}$, akin to Mothilal et al. (2020); Lang et al. (2022) that we refer to as *endogenous sparsity* method. In the style of tabular counterfactual explanations, we train a new regressor, which predicts an attribute from all

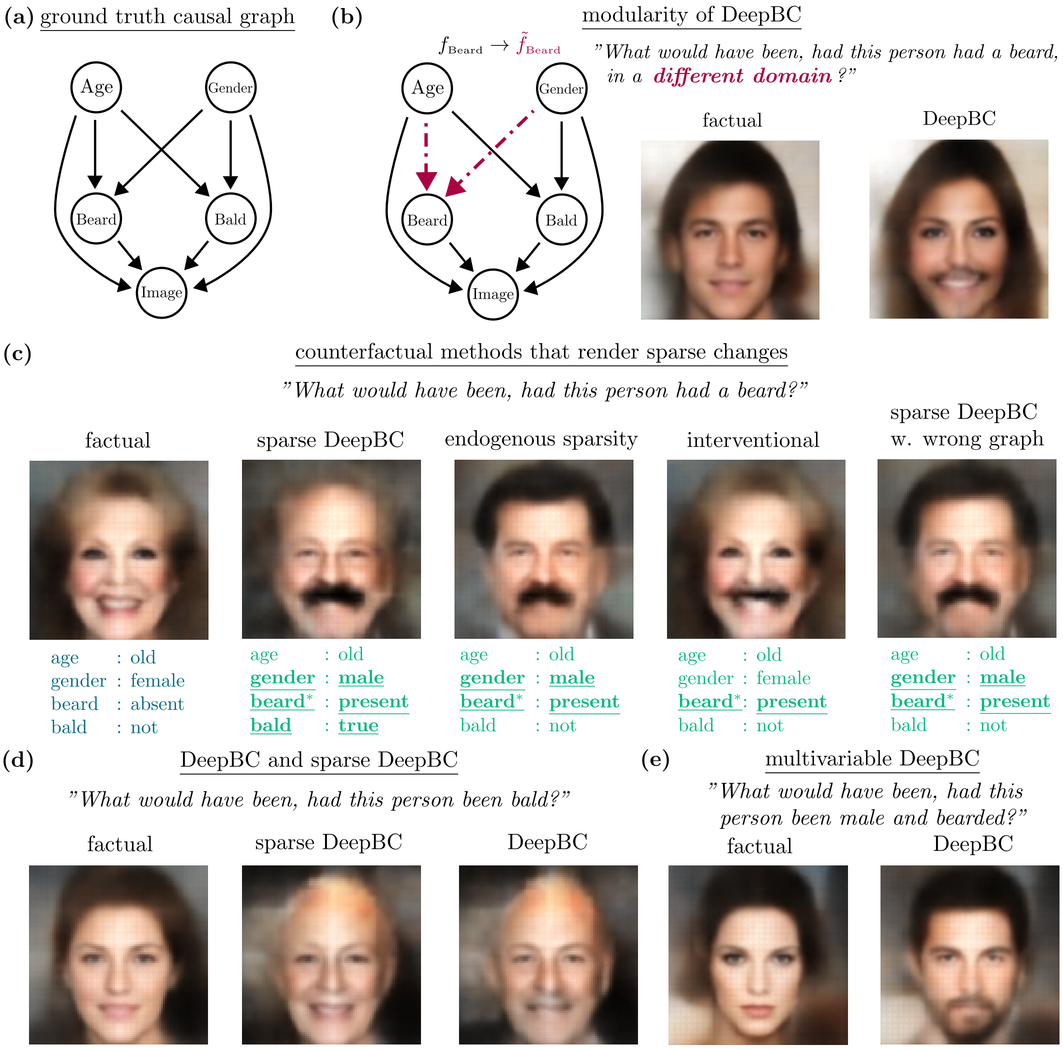

Figure 5: **DeepBC for CelebA**. **(a)** The ground truth causal graph for the considered variables as used by Yang et al. (2021). **(b)** A male, beardless person develops female traits as an upstream of antecedent beard, where the learned structural equation f_{Beard} is replaced by $\tilde{f}_{\text{Beard}}$ to mimic an out-of-domain setting that can be handled by DeepBC, but not by prior methods. **(c)** Variables that differ from the factual are underlined. Both sparse DeepBC and the endogenous sparsity method alter gender to add a beard while keeping age unchanged. However, only sparse DeepBC respects the causal downstream: baldness increases as gender is changed. In contrast, the endogenous sparsity method leaves the variable bald unchanged, though it would be very likely for an old man to be bald. The interventional counterfactual does not trace back the antecedent and simply adds a beard to the factual, keeping all other variables (except for the downstream image) unchanged. The last image shows that the result of sparse DeepBC is highly dependent on using the correct graph structure. **(d)** DeepBC generally modifies all variables, in contrast to sparse DeepBC (which only modifies age in this instance). **(e)** DeepBC allows for choosing antecedents in a versatile manner.

other attributes (not including the image). We then employ sparse DeepBC on this regressor, but measure distance in $\mathbf{x}$ rather than $\mathbf{u}$. 2) Wrong causal graph: We assess how choosing a different causal graph changes the result of the counterfactual. For this, we use the causal graph as shown in App. C **(a)**. 3) Non-causal counterfactual explanation: We use an image regressor, together with an *unconditional* auto-encoder to generate counterfactual explanations, according to (6). This corresponds to the core idea of how Jacob et al. (2022); Rodríguez et al. (2021) obtain counterfactual explanations for image data. We show a comparison to this baseline in App. C.

Results. Unlike previous approaches, sparse DeepBC measures sparsity in terms of $\mathbf{u}$ (that are subject to the causal laws) rather than $\mathbf{x}$ directly. We refer to the latter approach as *endogenous sparsity*. Fig. 5 ($\mathbf{c}$) shows sparse DeepBC ($M = 2$, see § 3.3) and other approaches that are able to generate counterfactuals that render sparse changes with respect to the considered (observed) attributes. As can be seen from the causal graph, the woman from the factual image could develop a beard by changing gender and age. Both the endogenous sparsity method and sparse DeepBC choose only gender, leaving the value of age fixed (standard DeepBC generally changes all variables). For DeepBC, despite the latent variable u_{Bald} not being updated, the realization of bald is automatically decreased as a downstream effect as encoded by the structural causal model. This lies in contrast to measuring sparsity in terms of $\mathbf{x}$ directly, where downstream effects are not taken into account in general. As a result, the image generated by the endogenous sparsity method does not render the man bald, thus keeping the value from the factual realization unchanged.

Another distinctive property of DeepBC is its modularity in terms of causal mechanisms. We present an illustrative example in Fig. 5 ($\mathbf{b}$), where we manually replace the original mechanism by which age and gender affect beard. The new mechanism is created such that being female is strongly positively correlated with having a beard, unlike in the model that was learned from data.

We show DeepBC for multivariable antecedents and highlight further properties in App. C.

5 Discussion

The following section discusses our approach in a broader context and highlights limitations and potential future work.

Sampling Counterfactuals from a Distribution. Prior work has raised the importance of obtaining multiple and diverse explanations for a single example (Mothilal et al., 2020), which our method currently only allows by varying the choice of distance function d in (3). As to fulfill this objective, it has been suggested to sample counterfactuals from a probability distribution (Guidotti, 2022, §3.1). The practical DeepBC formulation proposed in this work considers a simplification of the backtracking framework to a constrained optimization problem that yields a single prediction only (§ 3). Prior methods have explored the use of amortized inference to obtain distributions over counterfactual explanations (Mahajan et al., 2019). However, this approach did not yield satisfactory results in the context of our method, since the true underlying distribution is often complex and we only considered Gaussians to approximate the latent posterior (see App. A.1). Yet, a possible way forward may be to consider approaches such as flow-based models (Kingma et al., 2016) or semi-amortization (Kim et al., 2018).

Non-Invertible Generative Models. A possible future line of research could be to explore how backtracking could be implemented for generative models whose latent variables cannot be inferred deterministically from the factual realization, such as diffusion models (Ho et al., 2020) and generative adversarial networks (Goodfellow et al., 2014), both of which are not invertible in general. One conceivable solution might be to adapt (4) as to jointly optimize over $\mathbf{u}$ and $\mathbf{u}^*$ in the latent space.

Model-free Counterfactual Explanations. One can think of DeepBC as a model-based method for generating counterfactual explanations. The explicit access to a causal model allows for its versatility, modularity and the capability to obtain causally compliant solution for varying choices of distance functions (§ 3.2, § 4). In general, one may however argue that contemporary methods for counterfactual explanations that act on a latent space (see (6)) perform some sort of backtracking implicitly, without the need for a causal model (see Fig. 6 **(d)** in App. C).

6 Conclusion

In this work, we have presented DeepBC, a practical algorithm for computing backtracking counterfactuals for deep structural causal models. We compared DeepBC to the main formulations employed in the field of counterfactual explanations, and found that compared to these prior works, DeepBC is versatile in that it supports complex graph structures, compliant with the given causal model and modular in that it enables generalization to out-of-domain settings. In fact, DeepBC can be seen as a general method for computing counterfactual explanations that measures distances between factual and counterfactual in the structured latent space of a causal model. We empirically demonstrated the merits of our approach in comparison to prior work in counterfactual explanations, where we highlight the importance of taking causal relationships into account.

Reproducibility Statement

Our anonymized source code is available at https://anonymous.4open.science/r/DeepBC_review-FE8B. The instructions for reproducing all visualizations are provided in the `README.md` file at the top level of the repository. All parameters can be found in the `config` folders within the respective subfolders. In addition, we provide a detailed description of the training procedures in App. A.4 and deep learning architectures in App. A.5.

References

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust Counterfactual Explanations on Graph Neural Networks. *Advances in Neural Information Processing Systems*, 34:5644–5655, 2021. 19

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl's Hierarchy and the Foundations of Causal Inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 507–556. 2022. 1

Daniel C. Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning. *Journal of Machine Learning Research*, 20(178):1–29, 2019. 6

Saloni Dash, Vineeth N. Balasubramanian, and Amit Sharma. Evaluating and Mitigating Bias in Image Classifiers: A Causal Perspective Using Counterfactuals. In *Winter Conference on Applications of Computer Vision*, pp. 915–924, 2022. 19

Ricardo Dominguez-Olmedo, Amir-Hossein Karimi, Georgios Arvanitidis, and Bernhard Schölkopf. On Data Manifolds Entailed by Structural Causal Models. 2023. 7

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. *Advances in Neural Information Processing Systems*, 32:7509–7520, 2019. 16

Falcon, William and The PyTorch Lightning team. PyTorch Lightning, 2019. 6

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked Autoencoder for Distribution Estimation. In *International Conference on Machine Learning*, pp. 881–889, 2015. 16

Tobias Gerstenberg, Christos Bechlivanidis, and David A. Lagnado. Back on track: Backtracking in counterfactual reasoning. In *Annual Meeting of the Cognitive Science Society*, volume 35, pp. 2386–2391, 2013. 1

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From Variational to Deterministic Autoencoders. In *International Conference on Learning Representations*, 2020. 17

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27:53–65, 2014. 9

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning Functional Causal Models with Generative Neural Networks. *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pp. 39–80, 2018. 19

Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pp. 1–55, 2022. 5, 9, 19

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, 2017. 6, 16

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 9

Zhiting Hu and Li Erran Li. A Causal Lens for Controllable Text Generation. *Advances in Neural Information Processing Systems*, 34:24941–24955, 2021. 19

Frank Jackson. A Causal Theory of Counterfactuals. *Australasian Journal of Philosophy*, 55(1): 3–21, 1977. 1, 19

Paul Jacob, Éloi Zablocki, Hedi Ben-Younes, Mickaël Chen, Patrick Pérez, and Matthieu Cord. STEEX: Steering Counterfactual Explanations with Semantics. In *European Conference on Computer Vision*, pp. 387–403, 2022. 5, 8

Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. Causal Normalizing Flows: From Theory to Practice. *arXiv preprint arXiv:2306.05415*, 2023. 19

Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in Neural Information Processing Systems*, 33:265–277, 2020. 19

Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic Recourse: from Counterfactual Explanations to Interventions. In *ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362, 2021. 19

Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal Autoregressive Flows. In *International Conference on Artificial Intelligence and Statistics*, pp. 3520–3528, 2021. 19

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-Amortized Variational Autoencoders. In *International Conference on Machine Learning*, pp. 2678–2687, 2018. 9

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2014. 3

Durk P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved Variational Inference with Inverse Autoregressive Flow. *Advances in Neural Information Processing Systems*, 29:4743–4751, 2016. 9

Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. *International Conference on Learning Representations*, 2018. 19

Jana Lang, Martin Giese, Winfried Ilg, and Sebastian Otte. Generating Sparse Counterfactual Explanations for Multivariate Time Series. *arXiv preprint arXiv:2206.00931*, 2022. 5, 7

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision*, pp. 3730–3738, 2015. 7

Christopher G. Lucas and Charles Kemp. An Improved Probabilistic Account of Counterfactual Reasoning. *Psychological Review*, 122(4):700, 2015. 1

Ana Lucic, Maartje A. Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 4499–4511, 2022. 19

Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. CLEAR: Generative Counterfactual Explanations on Graphs. *arXiv preprint arXiv:2210.08443*, 2022. 19

Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. *arXiv preprint arXiv:1912.03277*, 2019. 9

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *ACM Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020. 5, 6, 7, 9, 19

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*, volume 32, pp. 8024–8035. 2019. 6

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. 3, 6, 17, 19

Judea Pearl. *Causality*. Cambridge University Press, 2009. 1

Ronan Perry, Julius Von Kügelgen, and Bernhard Schölkopf. Causal Discovery in Heterogeneous Environments Under the Sparse Mechanism Shift Hypothesis. *Advances in Neural Information Processing Systems*, 35:10904–10917, 2022. 5

Patrik Reizinger, Luigi Gresele, Jack Brady, Julius von Kügelgen, Dominik Zietlow, Bernhard Schölkopf, Georg Martius, Wieland Brendel, and Michel Besserve. Embrace the Gap: VAEs Perform Independent Mechanism Analysis. *Advances in Neural Information Processing Systems*, 35:12040–12057, 2022. 3

Danilo Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015. 3

Lance J. Rips. Two Causal Theories of Counterfactual Conditionals. *Cognitive Science*, 34(2):175–221, 2010. 1

Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam Laradji, Laurent Charlin, and David Vazquez. Beyond Trivial Counterfactual Explanations with Diverse Valuable Explanations. In *International Conference on Computer Vision*, pp. 1056–1065, 2021. 5, 8

Neal J. Roese. Counterfactual Thinking. *Psychological Bulletin*, 121(1):133–148, 1997. 1

Pedro Sanchez and Sotirios A. Tsaftaris. Diffusion Causal Models for Counterfactual Estimation. *Conference on Causal Learning and Reasoning*, pp. 1–21, 2022. 19

Pablo Sanchez-Martin, Miriam Rateike, and Isabel Valera. VACA: Design of Variational Graph Autoencoders for Interventional and Counterfactual Queries. *AAAI Conference on Artificial Intelligence*, pp. 8159–8168, 2022. 19

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward Causal Representation Learning. *IEEE*, 109(5): 612–634, 2021. 5

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems*, 28: 3483–3491, 2015. 3

Peter Spirtes. Introduction to Causal Inference. *Journal of Machine Learning Research*, 11(5): 1643–1662, 2010. 3

Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A PyTorch Package for Normalizing Flows. *Journal of Open Source Software*, 8(86):5361, 2023. 6

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review. *arXiv preprint arXiv:2010.10596*, 2020. 19

Julius von Kügelgen, Abdirisak Mohamed, and Sander Beckers. Backtracking Counterfactuals. In *Conference on Causal Learning and Reasoning*, pp. 177–196, 2023. 1, 4

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology*, 31: 841–887, 2017. 2, 4, 5

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning Likelihoods with Conditional Normalizing Flows. *arXiv preprint arXiv:1912.00042*, 2019. 3

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Structured Causal Disentanglement in Variational Autoencoder. In *Conference on Computer Vision and Pattern Recognition*, pp. 9593–9602, 2021. 7, 8

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 32:9240–9251, 2019. 19

A Appendix

A.1 Formal Definition of Interventional and Backtracking Counterfactuals

Both kinds of counterfactuals can be computed in a three-step-procedure.

Interventional Counterfactuals

1. **Abduction**: Compute the distribution of $\mathbf{U} \mid \mathbf{x}$, given the factual realization $\mathbf{x}$ of $\mathbf{X}$.

2. **Action**: Obtain an altered collection of structural assignments $(f_1^*, f_2^*, ..., f_n^*)$ by setting $x_i \leftarrow x_i^* = f_i^*$, for all $i \in S$. Leave all other structural assignments unmodified, i.e., $f_j^* = f_j$, for all $j \notin S$.

3. **Prediction**: Compute a distribution over $\mathbf{X}_\mathrm{I}^*$ as the pushforward of the distribution of $\mathbf{U} \mid \mathbf{x}$ by $\mathbf{F}^*$.

Backtracking Counterfactuals

1. **Cross-World Abduction**: Use the antecedent $\mathbf{x}_S^*$ and the factual realization $\mathbf{x}$ to obtain $p(\mathbf{u}^*, \mathbf{u} \mid \mathbf{x}_S^*, \mathbf{x})$, using the backtracking conditional $p(\mathbf{u}^*|\mathbf{u})$ and latent prior density $p(\mathbf{u})$:

$$p(\mathbf{u}^*, \mathbf{u} \mid \mathbf{x}_S^*, \mathbf{x}) \;=\; \frac{p(\mathbf{u}^*, \mathbf{u}, \mathbf{x}_S^*, \mathbf{x})}{p(\mathbf{x}_S^*, \mathbf{x})} \;=\; \frac{p(\mathbf{u}^*|\mathbf{u})\; p(\mathbf{u})\; \delta_{\mathbf{x}}(\mathbf{F}(\mathbf{u}))\delta_{\mathbf{x}_S^*}(\mathbf{F}_S(\mathbf{u}^*))}{\int\int p(\mathbf{u}^*|\mathbf{u})\; p(\mathbf{u})\; \delta_{\mathbf{x}}(\mathbf{F}(\mathbf{u}))\delta_{\mathbf{x}_S^*}(\mathbf{F}_S(\mathbf{u}^*))\; d\mathbf{u}\, d\mathbf{u}^*},$$

where $\delta_{\mathbf{x}}(\,\cdot\,)$ refers to the dirac delta at $\mathbf{x}$.

2. **Marginalization**: Marginalize over $\mathbf{U}$ to obtain the density $p(\mathbf{u}^* \mid \mathbf{x}_S^*, \mathbf{x})$ of the counterfactual posterior:

$$p(\mathbf{u}^* \mid \mathbf{x}_S^*, \mathbf{x}) \;=\; \int p(\mathbf{u}^*, \mathbf{u} \mid \mathbf{x}_S^*, \mathbf{x})\; d\mathbf{u}.$$

3. **Prediction**: Compute a distribution over $\mathbf{X}_\mathrm{B}^*$ by marginalizing over the counterfactual latents $\mathbf{U}^*$:

$$p(\mathbf{x}^* \mid \mathbf{x}_S^*, \mathbf{x}) \;=\; \int p(\mathbf{u}^* \mid \mathbf{x}_S^*, \mathbf{x})\delta_{\mathbf{x}^*}(\mathbf{F}(\mathbf{u}^*))\; d\mathbf{u}^*.$$

A.2 Formal Derivation of DeepBC

We derive (3) from the three-step-procedure of backtracking counterfactuals (see App. A.1) as follows:

1. **Cross-World Abduction**: By the deterministic relationship between latents and observables, we see that

$$\begin{aligned} p(\mathbf{u}^*, \mathbf{u} \mid \mathbf{x}_S^*, \mathbf{x}) \;&=\; p(\mathbf{u}^* \mid \mathbf{u}, \mathbf{x}_S^*, \mathbf{x})\; p(\mathbf{u} \mid \mathbf{x}_S^*, \mathbf{x}) = \; p(\mathbf{u}^* \mid \mathbf{u}, \mathbf{x}_S^*)\; p(\mathbf{u} \mid \mathbf{x}) \\ &=\; p(\mathbf{u}^* \mid \mathbf{u}, \mathbf{x}_S^*)\; \delta_{\mathbf{F}^{-1}(\mathbf{x})}(\mathbf{u}). \end{aligned}$$

2. **Marginalization**: All the probability is located at $\mathbf{F}^{-1}(\mathbf{x})$, which is why marginalization reduces to

$$p(\mathbf{u}^* \mid \mathbf{x}_S^*, \mathbf{x}) \;=\; p(\mathbf{u}^*,\; \mathbf{u} = \mathbf{F}^{-1}(\mathbf{x}) \mid \mathbf{x}_S^*, \mathbf{x}).$$

3. **Prediction**: By the deterministic relationship between latents and observables, we obtain samples from $\mathbf{X}^* \mid \mathbf{x}_S^*, \mathbf{x}$ simply by sampling from $\mathbf{U}^* \mid \mathbf{F}^{-1}(\mathbf{x}), \mathbf{x}_S^*$ and then subsequently mapping these samples through the function $\mathbf{F}(\mathbf{u}^*)$ to obtain the corresponding observables $\mathbf{x}^*$:

$$\mathbf{u}^* \;\sim\; \mathbf{U}^* \mid \mathbf{F}^{-1}(\mathbf{x}),\, \mathbf{x}_S^*, \quad \mathbf{x}^* \;=\; \mathbf{F}(\mathbf{u}^*).$$

Instead of sampling, however, we restrict ourselves to the mode of the distribution of $\mathbf{U}^* \mid \mathbf{F}^{-1}(\mathbf{x}), \mathbf{x}_S^*$. We assume that the backtracking conditional density $p(\mathbf{u}^*|\mathbf{u})$ has the following form

$$p(\mathbf{u}^*|\mathbf{u}) \;\propto\; \exp\left\{ -\sum_{i=1}^{n} d(u_i^*, u_i) \right\},$$

where d is a distance function. Then, we have

$$p(\mathbf{u}^* \mid \mathbf{F}^{-1}(\mathbf{x}), \mathbf{x}_S^*) \;\propto\; \begin{cases} \exp\left\{-\sum_{i=1}^{n} d\left(u_i^*, \mathbf{F}_i^{-1}(\mathbf{x})\right)\right\}, & \text{if } \mathbf{F}_S(\mathbf{u}^*) \;=\; \mathbf{x}_S^* \\ 0, & \text{otherwise.} \end{cases}$$

By taking the logarithm and ignoring constants, we obtain

$$\log p(\mathbf{u}^* \mid \mathbf{F}^{-1}(\mathbf{x}), \mathbf{x}_S^*) \;=\; \begin{cases} -\sum_{i=1}^{n} d\left(u_i^*, \mathbf{F}_i^{-1}(\mathbf{x})\right), & \text{if } \mathbf{F}_S(\mathbf{u}^*) \;=\; \mathbf{x}_S^* \\ -\infty, & \text{otherwise.} \end{cases}$$

We conclude by noting that $\arg\max_{\mathbf{u}^*} \log p(\mathbf{u}^* \mid \mathbf{F}^{-1}(\mathbf{x}), \mathbf{x}_S^*)$, composed with $\mathbf{F}$, is equivalent to (3).

A.3 Derivation of (3)

As a result of the linearization of $\mathbf{F}$, (7) simplifies to

$$\begin{aligned} &||\mathbf{u}' - \mathbf{u}||_2^2 \;+\; \lambda||\mathbf{J}_S(\mathbf{u}' - \mathbf{u}) + \mathbf{F}_S(\mathbf{u}) - \mathbf{x}_S^*||_2^2 \\ &\qquad = \quad ||\mathbf{u}' - \mathbf{u}||_2^2 \;+\; \lambda||\mathbf{J}_S\mathbf{u}' - \tilde{\mathbf{x}}_S^*||_2^2 \;=:\; \tilde{\mathcal{L}}(\mathbf{u}'). \end{aligned} \tag{9}$$

We see that $\tilde{\mathcal{L}}(\mathbf{u}')$ is convex and differentiable with respect to $\mathbf{u}'$, which means that $\nabla_{\mathbf{u}'}\tilde{\mathcal{L}}(\mathbf{u}') = \mathbf{0}$ implies optimality of $\mathbf{u}'$. To derive $\mathbf{u}'_{\text{opt}}$, we observe that

$$\nabla_{\mathbf{u}'}\tilde{\mathcal{L}}(\mathbf{u}') \;=\; 2(\mathbf{u}' - \mathbf{u} + \lambda\mathbf{J}_S^\top\mathbf{J}_S\mathbf{u}' - \mathbf{J}_S^\top\tilde{\mathbf{x}}_S^*).$$

As a result, $\mathbf{u}'_{\text{opt}}$ is given by

$$\mathbf{u}'_{\text{opt}} \;=\; (\mathbf{I} + \lambda\mathbf{J}_S^\top\mathbf{J}_S)^{-1}(\mathbf{u} + \lambda\mathbf{J}_S^\top\tilde{\mathbf{x}}_S^*).$$

A.4 Training Procedures

We train all models with the following parameters:

optimizer	train/val. split ratio	regularization	max. # epochs
Adam	0.8	early stopping	1000

A.4.1 Morpho-MNIST

We use the same training parameters for both normalizing flow models. Patience refers to the number of epochs without further decrease in validation loss that early stopping regularization waits.

model	batch size train	batch size val.	learning rate	patience
Flow	64	full	10^{-3}	2
VAE	128	256	10^{-6}	10

A.4.2 CelebA

We use the same training parameters for all normalizing flow models.

model	batch size train	batch size val.	learning rate	patience
Flow	64	256	10^{-3}	2
VAE	128	256	10^{-6}	50

A.5 Network Architectures

Notation. We denote concatenations of variables by $[\,\cdot\,,\,\cdot\,,...,\,\cdot\,]$. We denote modules that are repeated n times by a superscript (n). For instance, Linear$^{(2)}(u)$ is shorthand for Linear $\circ$ Linear (u), i.e., two linear layers.

Flow Layers. In all of our experiments, we make use of common types of flow layers:

$\text{QuadraticSpline}(u_i)$ is a standard quadratic spline flow (Durkan et al., 2019).

$\text{ConstScaleShift}(u_i)$ performs a constant affine transformation with learned, but unconditional, location and scale parameters μ and σ:

$$\text{ConstScaleShift}(u_i) = \sigma \cdot u_i + \mu.$$

$\text{ScaleShift}(u_i, \mathbf{x}_{\text{pa}(i)})$ performs the same operation as $\text{ConstScaleShift}(u_i)$, but μ and σ are computed as a function of u_i and $\mathbf{x}_{\text{pa}(i)}$ via a two-layer Masked Autoencoder for Distribution Estimation (MADE) module (Germain et al., 2015) with ReLU activation functions and one-dimensional hidden units.

A.5.1 MORPHO-MNIST

For the thickness variable, we construct the flow as

$$f_T(u_T) = \text{ConstScaleShift} \circ \text{QuadraticSpline}^{(5)}\,(u_T).$$

For intensity, we use

$$f_I(t, u_I) = \text{ConstScaleShift} \circ \text{Sigmoid} \circ \text{QuadraticSpline}^{(3)} \circ \text{ScaleShift}\,([t, u_I]),$$

where Sigmoid denotes the (constant) sigmoid function.

For the MNIST image, we use a convolutional β-VAE (Higgins et al., 2017) with $\beta = 3$ and the following encoder parameterization:

$$\begin{aligned} f_{\text{Img}}(t, i, \text{img}) &\approx e_{\text{Img}}(t, i, \text{img}) \\ &= \text{Linear}\left(\left[t,\ i,\ \left(\text{Linear} \circ \text{Pool2D} \circ (\text{ReLU} \circ \text{Conv2D})^{(4)}\right)\ (\text{img})\right]\right), \end{aligned}$$

where the Conv2D layers (starting with parameters from the layer closest to the input) are parameterized by $\texttt{out\_channels} = (8, 16, 32, 64)$, $\texttt{kernel\_size} = (4, 4, 4, 3)$, $\texttt{stride} = (2, 2, 2, 2)$, $\texttt{padding} = (1, 1, 1, 0)$. The linear layers are analogously parameterized with the output dimensions $\texttt{out} = (128, 16, 16)$, i.e., $\dim(u_{\text{Img}}) = 32$. For the decoder, we use

$$\begin{aligned} f^{-1}_{\text{Img}}(t, i, u_{\text{Img}}) &\approx d_{\text{Img}}(t, i, u_{\text{Img}}) \\ &= \text{TransConv2D} \circ (\text{ReLU} \circ \text{TransConv2D})^{(4)} \circ \text{Linear}\,([t, i, u_{\text{Img}}]), \end{aligned}$$

where the linear layer has output dimension $\texttt{out} = 64$ and the transpose convolution layers (starting with parameters from the layer closest to the input) are parameterized by $\texttt{out\_channels} = (64, 32, 16, 1)$, $\texttt{kernel\_size} = (3, 4, 4, 4)$, $\texttt{stride} = (2, 2, 2, 2)$, $\texttt{padding} = (0, 1, 0, 1)$.

A.5.2 CELEBA

We preprocess all attributes via separate classifiers C_{Attr}, i.e., one individual classifier per attribute. The classifier has the following architecture:

$$C_{\text{Attr}}(\text{img}) = \text{Linear} \circ \text{Dropout} \circ \text{ReLU} \circ \text{Linear} \circ (\text{MaxPool2D} \circ \text{ReLU} \circ \text{Conv2D})^{(4)}\,(\text{img}). \quad (10)$$

We then standardize the output logits of C_{Attr}, for each attribute individually.

As for MorphoMNIST, we train one normalizing flow for each attribute. For this, we use the standardized logits from the classifiers rather than the original binary attributes from the data set. To model the non-Gaussian distributions, we employ the following flow architecture:

$$f_{\text{Attr}}(t, u_{\text{Attr}}) = \text{ScaleShift}\left(\left[\left(\text{QuadraticSpline}^{(10)} \circ \text{ConstScaleShift}\right)(u_{\text{Attr}}),\ \mathbf{x}_{\text{pa(Attr)}}\right]\right),$$

For the β-VAE with $\beta = 3$, we follow a slightly different approach as for A.5.1. Rather than concatenating the conditional variables $\mathbf{x}_{\text{pa}(i)}$ at the end of the encoder, we instead create an additional channel $\text{chan}_{\text{attr}}$ for each attribute attr that we concatenate to the RGB channels of the image. Specifically, we obtain the channel by broadcasting the continuous attribute value x_{Attr} like

$$\text{ch}_{\text{Attr}} = \mathbf{1}_{128\times128} \cdot x_{\text{Attr}},$$

where we replace the MADE module by a linear function for Bald, since the signal-to-noise ratio is low for this variable. The reason is that Beard is the only variable that cannot be modeled well as a linear function of its causal parents Age and Gender.

where $\mathbf{1}_{128\times128}$ is a matrix of dimensionality 128×128 that consists only of 1. We then feed $\tilde{\mathbf{x}} := [x_R, x_G, x_B, \text{ch}_{\text{Beard}}, \text{ch}_{\text{Bald}}, \text{ch}_{\text{Gender}}, \text{ch}_{\text{Age}}] \in \mathbb{R}^{128\times128\times7}$ directly into the encoder with the following architecture (roughly inspired by Ghosh et al. (2020)):

$$\begin{aligned} f_{\text{Img}}(\tilde{\mathbf{x}}) &\approx e_{\text{Img}}(\tilde{\mathbf{x}}) \\ &= \text{Linear} \circ \text{Pool2D} \circ \left(\text{ReLU} \circ \text{BatchNorm2D} \circ \text{Conv2D}\right)^{(6)} (\tilde{\mathbf{x}}), \end{aligned}$$

where the final linear layer has output dimension $\texttt{out} = 512$ and the transpose convolution layers (starting with parameters from the layer closest to the input) are parameterized by $\texttt{out\_channels} = (128, 128, 128, 256, 512, 1024)$, $\texttt{kernel\_size} = (3, 3, 3, 3, 3, 3)$, $\texttt{stride} = (2, 2, 2, 2, 2, 2)$, $\texttt{padding} = (1, 1, 1, 1, 1, 1)$. For the decoder, noting that $\mathbf{x}_{\text{pa(Img)}} = [x_{\text{Beard}}, x_{\text{Bald}}, x_{\text{Gender}}, x_{\text{Age}}]$, we use

$$\begin{aligned} f^{-1}_{\text{Img}}(\mathbf{x}_{\text{pa(Img)}}, u_{\text{Img}}) &\approx d_{\text{Img}}(\mathbf{x}_{\text{pa(Img)}}, u_{\text{Img}}) \\ = \text{TransConv2D} \circ \left(\text{ReLU} \circ \text{BatchNorm2D} \circ \text{TransConv2D}\right)^{(4)} &\circ \text{Linear}\left([\mathbf{x}_{\text{pa(Img)}}, u_{\text{Img}}]\right), \end{aligned}$$

where the first linear layer maps to $\mathbb{R}^{4\cdot1024}$, which is then reshaped to a feature map in $\mathbb{R}^{2\times2\times1024}$. The consecutive transposed convolutional layers have the parameters $\texttt{out\_channels} = (512, 256, 128, 128, 128)$, $\texttt{kernel\_size} = (3, 3, 3, 3, 3)$, $\texttt{stride} = (2, 2, 2, 2, 2)$, $\texttt{padding} = (1, 1, 1, 1, 1)$.

B Ground Truth Structural Equations Morpho-MNIST

The structural equation for thickness T and intensity I are given as

$$\begin{aligned} T &\leftarrow 0.5 + U_T, & U_T &\sim \Gamma(10,\, 5) \\ I &\leftarrow 191 \cdot \text{Sigmoid}\left(0.5 \cdot U_I + 2 \cdot T - 5\right) + 64, & U_I &\sim \mathcal{N}(0, 1). \end{aligned}$$

For details about how the MNIST images were modified as to change perceived thickness and intensity, we refer the reader to Pawlowski et al. (2020).

C Extra Plots for CelebA

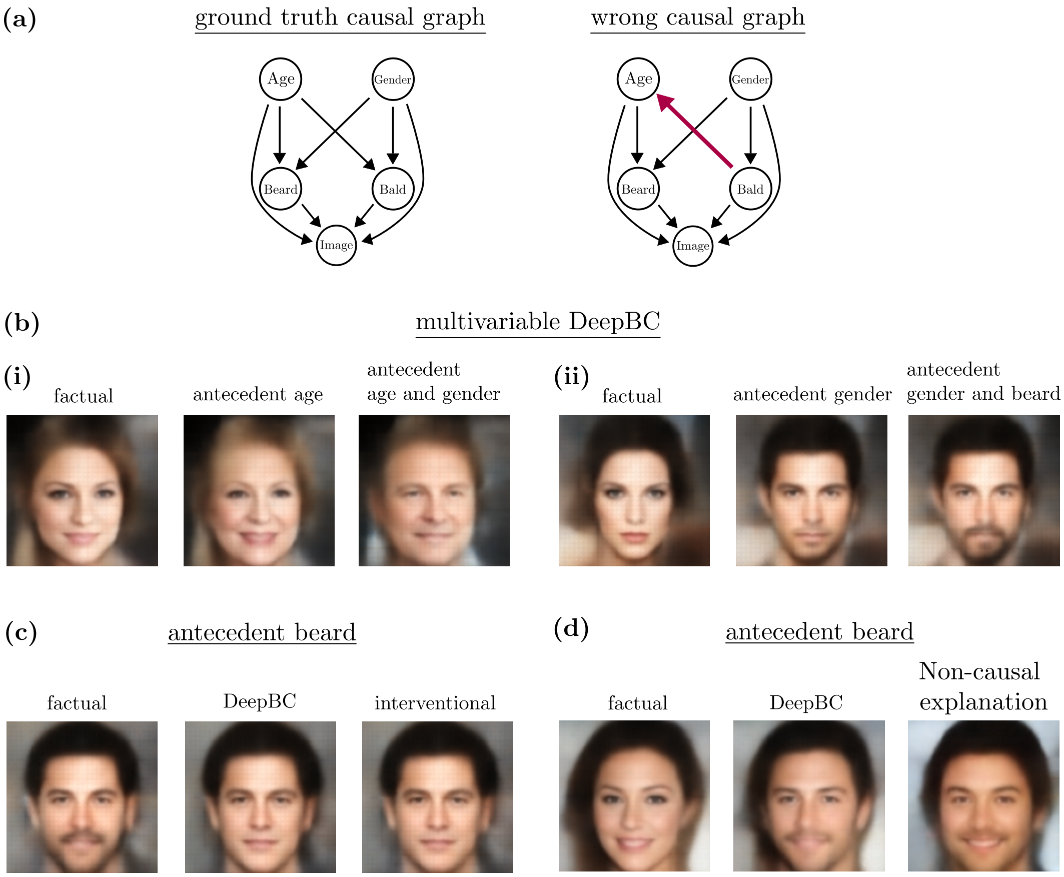

Figure 6: **Additional plots for CelebA. (a)** The right causal graph shows the wrong graph used in Fig. 5 (c). **(b)** Examples for multivariable DeepBC. Example (b) (ii) corresponds to Fig. 5 (e) and we show how the single variable antecedent contrasts the dual variable antecedent. **(c)** DeepBC takes into account non-deterministic relationships between variables: In this setting, the removed beard is traced back to u_{Beard} rather than other variables. The result is highly similar to the interventional example (plotted for comparison). **(d)** Non-causal explanation methods based on (6) yield similar results to vanilla DeepBC in some settings.

D Related Work

This section is organized into two lines of prior work. The first line encompasses methods that incorporate causality into the field of counterfactual explanations. We do however note that the general field of counterfactual explanations has made many significant advances that are not directly related to causality in recent years. For a comprehensive overview over these developments, we refer to Guidotti (2022) and Verma et al. (2020). The second line discusses how deep neural networks have been used within the context of structural causal models, as to facilitate counterfactual computation.

Causality in Counterfactual Explanations. As explained in § 3.1, our DeepBC approach is related to the field of counterfactual explanations. Our work builds therefore on earlier approaches that generate counterfactual explanations by incorporating causal models: One line of work focuses on the setting of explaining the prediction of a machine learning model based on features along with a (causal) graph in the sense of (2) (Ying et al., 2019; Bajaj et al., 2021; Lucic et al., 2022; Ma et al., 2022). Whereas our approach never manipulates the given graph structure of the causal graph, these prior works alter the graph structure in a fashion that corresponds to neither interventional nor backtracking counterfactuals.

Another line of work raises the importance of causality to ensure actionability of counterfactual explanations in a sense that an alternative outcome could have been achieved by performing alternative actions, without violating causal relationships (Karimi et al., 2020; 2021). These works fundamentally differ from the present work in that actions break causal relationships, which lies in stark contrast to the backtracking approach. The latter seeks to trace back counterfactuals to changes in latent variables rather than changes in causal relationships. However, this line of research is related to ours in that it argues for respecting causal mechanisms in generating counterfactuals.

The most similar existing work to ours is that of Mothilal et al. (2020). Similarly to the present work, the authors employ deep generative modelling and measure distance between factual and counterfactual examples in a latent space. The most distinctive difference to our work is that Mothilal et al. (2020) impose causal constraints via a *causal proximity loss* in the observable variables $\mathbf{x}$. This is in contrast to the backtracking philosophy (Jackson, 1977) that we follow. In our approach, all changes are traced back solely to latent variables $\mathbf{u}$ that are embedded into the causal model such that causal constraints are fulfilled automatically, obviating the need for an additional loss. At the same time, our approach is more versatile as any of the given variables could be used as antecedent, whereas Mothilal et al. (2020) requires a specific label variable.

Counterfactuals in Deep Structural Causal Models. The integration of deep generative components such as normalizing flows and variational auto-encoders into structural causal models can be traced back to works from Kocaoglu et al. (2018); Goudet et al. (2018); Pawlowski et al. (2020) and others. Subsequently, this approach has been adopted in various works for computing counterfactuals in applications such as natural language processing (Hu & Li, 2021) and bias reduction (Dash et al., 2022). Other recent works have explored the use of graph neural networks (Sanchez-Martin et al., 2022), normalizing flows (Khemakhem et al., 2021; Javaloy et al., 2023) or diffusion models (Sanchez & Tsaftaris, 2022) to construct structural causal models.

In the present work, we employ variational auto-encoders and normalizing flows to construct deep structural causal models (outlined in § 2.2). Nevertheless, we regard the design choices within our implementation as agnostic to various choices of architecture. Specifically, we deem our approach applicable to any deep structural causal model architecture that yields a reduced-form that is both invertible and differentiable.