Estimating Categorical Counterfactuals via Deep Twin Networks

Athanasios Vlontzos
Spotify, UK
Imperial College London, UK
athanasiosv@spotify.com

Bernhard Kainz
FAU Erlangen-Nuremberg, Erlangen, Germany
Imperial College London, London UK

Ciarán M. Gilligan Lee
Spotify, UK
University College London, UK

Abstract

Counterfactual inference is a powerful tool, capable of solving challenging problems in high-profile sectors. To perform counterfactual inference, one requires knowledge of the underlying causal mechanisms. However, causal mechanisms cannot be uniquely determined from observations and interventions alone. This raises the question of how to choose the causal mechanisms so that resulting counterfactual inference is trustworthy in a given domain. This question has been addressed in causal models with binary variables, but the case of categorical variables remains unanswered. We address this challenge by introducing for causal models with categorical variables the notion of *counterfactual ordering*, a principle that posits desirable properties causal mechanisms should posses, and prove that it is equivalent to specific functional constraints on the causal mechanisms. To learn causal mechanisms satisfying these constraints, and perform counterfactual inference with them, we introduce *deep twin networks*. These are deep neural networks that, when trained, are capable of *twin network* counterfactual inference—an alternative to the *abduction, action, & prediction* method. We empirically test our approach on diverse real-world and semi-synthetic data from medicine, epidemiology, and finance, reporting accurate estimation of counterfactual probabilities while demonstrating the issues that arise with counterfactual reasoning when counterfactual ordering is not enforced.

1 Introduction

"If my credit score had been better, would I have been approved for this loan?", "What is the effect of the diabetes type on the risk of stroke?". Causal questions like these are routinely asked by scientists and the public alike. Recent machine learning advances have enabled the field to address causal questions in high-dimensional datasets to a certain extent Schwab et al. [2018], Alaa et al. [2017], Shi et al. [2019]. However, most of these methods focus on `Interventions`, which only constitute the second-level of Pearl's three-level causal hierarchy Pearl [2009], Bareinboim et al. [2020]. At the top of the hierarchy sit `Counterfactuals`. These subsume interventions and allow one to assign fully causal explanations to data.

Counterfactuals investigate alternative outcomes had some pre-conditions been different. The crucial difference between counterfactuals and interventions is that the evidence the counterfactual is "counter-to" can contain the variables we wish to intervene on or predict. The first question posed at the start of this paper, for instance, is a counterfactual one. Here we want to know if improving

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

our credit score will lead to loan approval in the explicit context that the loan has just been declined. A corresponding interventional query would be "what is the impact of the credit score on the loan approval chances?". Here, evidence that the loan has just been denied is *not* used in estimating the impact. The second question posed at the start of this paper—regarding the effect of diabetes type on the risk of stroke—is an interventional question. By utilising this additional information, counterfactuals enable more nuanced and personalised reasoning and decision making. Counterfactual inference has been applied in high profile sectors like medicine Richens et al. [2020], Oberst and Sontag [2019], legal analysis Lagnado et al. [2013], fairness Kusner et al. [2017], explainability Galhotra et al. [2021], and advertising Ang Li [2019].

To perform counterfactual inference, one requires knowledge of the causal mechanisms. However, the causal mechanisms cannot be uniquely determined from observations and interventions alone. Indeed, two causal models that have the same conditional and interventional distributions can disagree about certain counterfactuals Pearl [2009]. Hence, without additional constraints on the form of the causal mechanisms, they can generate "non-intuitive" counterfactuals that conflict with domain knowledge, as originally pointed out by Oberst and Sontag [2019].

This raises the question of how best to choose the causal mechanisms so that resulting counterfactual inference is trustworthy in a given domain. Despite the importance of counterfactual inference, this question has only been addressed in causal models with binary treatment and outcome variables Tian and Pearl [2000]. The case of categorical variables remains unanswered. Beyond binary variables, previous work has only derived upper and lower bounds for counterfactual probabilities Zhang and Bareinboim [2020]. In many cases, these bounds can be too wide to be informative. We address this challenge by introducing for causal models with categorical variables the notion of *counterfactual ordering*, a principle that posits desirable properties causal mechanisms should posses, and prove that it is equivalent to specific functional constraints on the causal mechanisms. Namely, we prove that causal mechanisms satisfying counterfactual ordering must be monotonic functions.

To learn such causal mechanisms, and perform counterfactual inference with them, we introduce *deep twin networks*. These are deep neural networks that, when trained, are capable of *twin network* counterfactual inference—an alternative to the *abduction, action, & prediction* method of counterfactual inference. Twin networks were introduced by Balke and Pearl [1994] and reduce estimating counterfactuals to performing Bayesian inference on a larger causal model, known as a *twin network*, where the factual and counterfactual worlds are jointly graphically represented. Despite their potential importance, twin networks have not been widely investigated from a machine learning perspective. We show that the graphical nature of twin networks makes them particularly amenable to deep learning.

We empirically test our approach on a variety of real and semi-synthetic datasets from medicine and finance, showing our method achieves accurate estimation of counterfactual probabilities. Moreover, we demonstrate that if counterfactual ordering is not enforced, the model generates "non-intuitive" counterfactuals that contradict domain knowledge in these cases. Our contributions are as follows:

1. We introduce *counterfactual ordering* for causal models with categorical variables, which posits desirable properties causal mechanisms should posses.
2. We prove *counterfactual ordering* is equivalent to specific functional constraints on the causal mechanisms. Namely, that they must be monotonic.
3. We introduce *deep twin networks* to learn such causal mechanisms and perform counterfactual inference. These are deep neural networks that, when trained, can perform *twin network* counterfactual inference.
4. We test our approach on real and semi-synthetic data, achieving *accurate* counterfactual estimation that complies with domain knowledge.

2 Preliminaries

2.1 Structural causal models

We work in the Structural Causal Models (SCM) framework. Chapter 7 of Pearl [2009] gives an in-depth discussion and for an up-to-date review of counterfactual inference and Pearl's Causal Hierarchy, see Bareinboim et al. [2020].

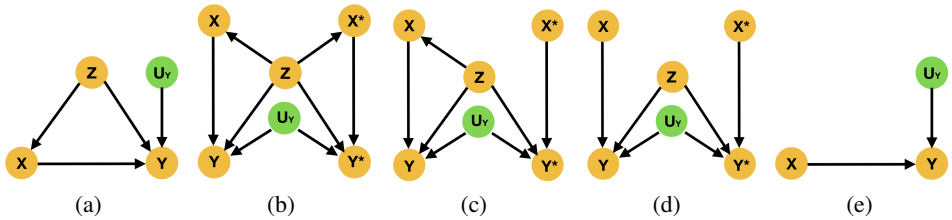


Figure 1: Orange nodes are observed, green latent. (a) Example SCM; (b) twin network of (a); (c) intervention in the twin network on node X^*; (d) interventions in the twin network on X & X^*; (e) Uncounfounded version of (a).

Definition 1 (Structural Causal Model)**.** *A structural causal model (SCM) specifies a set of latent variables $U = \{u_1, \ldots, u_n\}$ distributed as $P(U)$, a set of observable variables $= \{v_1, \ldots, v_m\}$, a directed acyclic graph (DAG), called the* causal structure *of the model, whose nodes are the variables $U \cup V$, a collection of functions $F = \{f_1, \ldots, f_n\}$, such that $v_i = f_i(PA_i, u_i)$, for $i = 1, \ldots, n$, where PA denotes the parent observed nodes of an observed variable.*

The collection of functions and distribution over latent variables induces a distribution over observable variables: $P(V = v) := \sum_{\{u_i | f_i(PA_i, u_i) = v_i\}} P(u_i)$. An example causal structure, represented as a directed acyclic graph (DAG), is depicted in Fig. 1a.

Definition 2 (Submodel)**.** *Let M be a structural causal model, X a subset of observed variables with realization x. A submodel M_x is the causal model with the same latent and observed variables as M, but with functions replaced with $F_x = \{f_i \mid v_i \notin X\} \cup \{f_j^{'}(PA_j, u_j) := x_j \mid v_j \in X\}$.*

Definition 3 (do-operator)**.** *Let M be a structural causal model, X a set of observed variables. The effect of action $do(X = x)$ on M is given by the submodel M_x.*

The do-operator forces variables to take certain values, regardless of the original causal mechanism. Graphically, $do(X = x)$ means deleting edges incoming to X and setting $X = x$. Probabilities involving $do(x)$ are normal probabilities in submodel M_x: $P(Y = y \mid \text{do}(X = x)) = P_{M_x}(y)$.

2.2 Counterfactual inference

Definition 4 (Counterfactual)**.** *The counterfactual "Y would be y in situation $U = u$, had X been x", denoted $Y_x(u) = y$, equates to $Y = y$ in submodel M_x for $U = u$.*

The latent distribution $P(U)$ allows one to define probabilities of counterfactual queries, $P(Y_y = y) = \sum_{u|Y_x(u)=y} P(u)$. For $x \neq x'$ one can also define joint counterfactual probabilities, $P(Y_x = y, Y_{x'} = y') = \sum_{u|Y_x(u)=y, \&Y_{x'}(u)=y'} P(u)$. Moreover, one can define a counterfactual distribution given seemingly contradictory evidence. Given a set of observed evidence variables E, consider the probability $P(Y_x = y' \mid E = e)$. Despite the fact that this query may involve interventions that contradict the evidence, it is well-defined, as the intervention specifies a new submodel. Indeed, $P(Y_x = y' \mid E = e)$ is given by Pearl [2009] $\sum_u P(Y_x(u) = y')P(u|e)$. The following theorem outlines how to compute such distributions.

Theorem 1 (Theorem 7.1.7 in Pearl [2009])**.** *Given SCM M with latent distribution $P(U)$ and evidence e, the conditional probability $P(Y_x \mid e)$ is evaluated as follows: 1)* ***Abduction:*** *Infer the posterior of the latent variables with evidence e to obtain $P(U \mid e)$, 2)* ***Action:*** *Apply $do(x)$ to obtain submodel M_x, 2)* ***Prediction:*** *Compute the probability of Y in the submodel M_x with $P(U \mid e)$.*

2.3 Twin network counterfactual inference

A practical limitation of Theorem 1 is that the abduction step requires large computational resources. Indeed, even if we start with a Markovian model in which background variables are mutually independent, conditioning on evidence—as in abduction—normally destroys this independence and makes it necessary to carry over a full description of the joint distribution over the background variables Pearl [2009]. Balke and Pearl [1994] introduced a method to address this difficulty. Their

method reduces estimating counterfactuals to performing Bayesian inference on an larger causal model, known as a *twin network*, where the factual and counterfactual worlds are jointly graphically represented, described in technical detail below. We have included an extended discussion of the computational distinction between twin networks and abduction-action-prediction in the Appendix.

A twin network consists of two interlinked networks, one representing the real world and the other the counterfactual world being queried. Constructing a twin network given a structural causal model and using it to compute a counterfactual query is as follows: First, one duplicates the given causal model, denoting nodes in the duplicated model via superscript *. Let $V = \{v_1, \ldots, v_n\}$ be observable nodes in the causal model and $V^* = \{v_1^*, \ldots, v_n^*\}$ the duplication of these. Then, for every node v_i^* in the duplicated, or "counterfactual," model, its latent parent u_i^* is replaced with the original latent parent u_i in the original, or "factual," model, such that the original latent variables are now a parent of two nodes, v_i and v_i^*. The two graphs are linked only by common latent parents, but share the same node structure and generating mechanisms. To compute a general counterfactual query $P(Y = y \mid E = e, \text{do}(X = x))$, one modifies the structure of the counterfactual network by dropping arrows from parents of X^* and setting them to value $X^* = x$. Then, in the twin network with this modified structure, one computes the following probability $P(Y^* = y \mid E = e, X^* = x)$ via standard inference techniques, where E are factual nodes. That is, in a twin network one has:

$$\begin{aligned} P(Y = y \mid & E = e, \text{do}(X = x)) = \\ & P(Y^* = y \mid E = e, X^* = x) \end{aligned} \tag{1}$$

To illustrate this concretely, consider the causal model with causal structure depicted in 1a, where variables X, Y are binary. The counterfactual statement to be computed is $P\left(Y = 0 \mid Y = 1,\ \text{do}(X = 0)\right)$. The twin network approach to this problem first constructs the linked factual and counterfactual networks depicted in 1b. The intervention $\text{do}(X^* = 0)$ is then performed in the counterfactual network; all arrows from the parents of X^* are removed and X^* is set to the value 0—graphically depicted in 1c. The above counterfactual query is reduced to the following conditional probability in 1c: $P\left(Y^* = 0 \mid Y = 1, X^* = 0\right)$, which can be computed using Bayesian inference techniques.

Our contribution: Despite their importance for counterfactual inference, twin networks have not been widely studied—particularly from a machine learning perspective. In the Methods section we demonstrate how to combine twin networks with neural networks to estimate counterfactuals.

2.4 Non-identifiability of counterfactuals

As a general principle, observational and interventional data do not allow for identification of counterfactual distributions, as multiple parametrizations of the causal phenomenon can be consistent with the observed data. Eq. 2 illustrates this phenomenon. Assume the simple case of DAG 1e, with X, Y *binary*, U_Y a four-valued variable distributed under $q(U_Y)$ and $\neg X$ to be the logical negation of X.

$$Y = \begin{cases} X, & \text{if } U_Y = 0 \\ 0, & \text{if } U_Y = 1 \\ 1, & \text{if } U_Y = 2 \\ \neg X, & \text{if } U_Y = 3 \end{cases} \tag{2}$$

In this case the conditional probabilities are given by $P(Y \mid X)$ are: $P(Y = 0|X = 0) = q(U_Y = 0) + q(U_Y = 1)$ and $P(Y = 0|X = 1) = q(U_Y = 1) + q(U_Y = 3)$. As there are no confounders in our example, these coincide with the interventional distributions $P(Y \mid \text{do}(X))$.

The causal model in Eq. 2 has 3 parameters, but the conditional distributions only provide 2 constraints on these parameters. Hence, due to the existence of this free parameter, there can exist models with the same conditional distributions, but different counterfactuals. This observation can bee seen as follows. Consider the following counterfactual query $P(Y_{X=1} = 1 \mid Y = 0, X = 0)$. It can be written as:

$$P(Y_{X=1} = 0 \mid Y = 1, X = 0) = \frac{q(U_Y = 3)}{q(U_Y = 2) + q(U_Y = 3)}$$

which follows by noting $Y = 1$ and $X = 0$ implies either $Y = \neg X$ or $Y = 1$, which happens with probability $q(U_Y = 2) + q(U_Y = 3)$, and within this context the only way $Y = 0$ when $X = 1$ is if $Y = \neg X$, which occurs with probability $q(U_Y = 3)$. Note that there is no way to write this counterfactual probability in terms of the conditional distributions $P(Y \mid X)$ alone. Moreover, one can have two models with the same functional form as Eq. 2 that yield the same conditional distributions, but give *different* predictions for the above counterfactual query. An example is one model with distribution over U_Y given by $\{q(U_Y)\}_0^3 = \{1/2, 1/6, 1/6, 1/6\}$, another with $\{q(U_Y)\}_0^3 = \{1/3, 1/3, 1/3, 0\}$. Hence, there are counterfactuals that are not *identifiable* from data. Such counterfactual distributions cannot be expressed in terms of observational or interventional distributions alone. This can lead to counterfactual predictions that conflict with domain knowledge, as shall be shown in the next section.

3 Methods

3.1 Non-identifiability & domain knowledge

Is non-identifiability of counterfactuals a problem? Given a causal model trained on observations and interventions, can we always trust its counterfactual predictions? In general the answer is no: counterfactual predictions from a causal model can conflict with domain knowledge—even if it perfectly reproduces observations and interventions, as we now show.

In epidemiology, causal models with the structure of Figure 1e are studied, where X is the presence of a risk factor and Y is the presence of a disease. From epidemiological domain knowledge, it is believed that risk factors always increase the likelihood of a disease being present Tian and Pearl [2000]—referred to as "no-prevention", that no individual in the population can be helped by exposure to the risk factor Pearl [1999]. Hence, if one observes a disease, but not the risk factor, then, in that context, if we had intervened to give that individual the risk factor, the likelihood of them not having the disease must be zero—as having the risk factor can only increase the likelihood of a disease.

We'll now describe two causal models that generate the same observations and interventions, yet the counterfactuals generated by one model satisfy the above domain knowledge and the other do not. Consider the two different parameterizations of the causal model discussed in Section 2.4: $\{q(U_Y)\}_0^3 = \{1/2, 1/6, 1/6, 1/6\}$ and $\{q(U_Y)\}_0^3 = \{1/3, 1/3, 1/3, 0\}$. Both models have the same conditional distributions, and we have $P(Y_{X=1} = 1) > P(Y_{X=0} = 1)$ and $P(Y_{X=1} = 0) < P(Y_{X=0} = 0)$. This tells us that intervening to set $X = 1$ *always* makes $Y = 1$ more likely, and doesn't increase the likelihood of $Y = 0$ relative to $X = 0$. So, at the interventional-level, these models seem to comply with Epidemiological domain knowledge that says that the presence of a risk factor, that is, $X = 1$, always makes disease, $Y = 1$, more likely.

Despite this, in the first model $P(Y_{X=1} = 1 \mid Y = 1, X = 0) < P(Y_{X=0} = 1 \mid Y = 1, X = 0)$. According to this model, $Y = 1$ becomes *less* likely when we intervene with $X = 1$ in the counterfactual context $Y_{X=0} = 1$—even though intervening to set $X = 1$ can only make $Y = 1$ more likely, and does not increase the likelihood of $Y = 0$. This is a very "non-intuitive" counterfactual prediction from the point of view of an Epidemiologist. However, in the second model, $P(Y_{X=1} = 1 \mid Y = 1, X = 0) = P(Y_{X=0} = 1 \mid Y = 1, X = 0)$. Indeed, in this model, no matter the counterfactual context, intervening to set $X = 1$ *never* reduces the likelihood $Y = 1$. Thus the second model complies fully with Epidemiological domain knowledge.

As the models agree on the data they're trained on, we must impose extra constraints to learn the model that generates domain-trustworthy counterfactuals. In the next section, we present a simple principle that provide such constraints.

3.2 Counterfactual ordering

We continue to consider the causal structure from Section 2.4 with a DAG as depicted in Figure 1e. However, now X, Y are categorical variables with an arbitrary number of categories N, M each. Inspired by the Epidemiological example from the previous section, we now define *counterfactual ordering*, which posits an intuitive relationship between counterfactual and intervantional distributions.

Definition 5 (Counterfactual Ordering)**.** *A causal model with categorical treatment variable X and categorical outcome variable Y satisfies* counterfactual ordering *if there exists an ordering on interventions and outcomes $\{x_0, x_1, \ldots, x_N\}$, $\{y_0, y_1, \ldots, y_M\}$ such that $P(Y_{x_i} = y_k) \geq P(Y_{x_j} = y_k)$ and $P(Y_{x_i} = y_h) \leq P(Y_{x_j} = y_h)$ for all $i > j$ and $k > h$, then it must be the case that $P(Y_{x_i} \geq y_k | Y_{x^*} = y^*) \geq P(Y_{x_j} \geq y_k | Y_{x^*} = y^*)$ for all counterfactual contexts $\{y^*, x^*\}$.*

This encodes to the following intuition: If intervention x_i only increases the likelihood of outcome y_k, relative to any intervention x_j with $j < i$, without increasing the likelihood of y_h for all $h < k$, then intervention x_i must increase the likelihood that the outcome we observe is at least as high as y_k, regardless of the context. Counterfactual ordering places the following constraints on a causal model.

Theorem 2. *If counterfactual ordering holds $P(Y_{x_j} = y_l | Y_{x_i} = y_h) = 0$ for all $l > h$ and $i > j$.*

Proofs are in Appendix. Equality's of the form $P(Y_x = y' | Y_{x'} = y) = 0$ are equivalent to the statement $\{Y_x = y'\} \wedge \{Y_{x'} = y\} =$ False, where $\wedge$ is the logical AND operator. Therefore, the conjunction of the input-output pairs $X = x, Y = y'$ and $X = x, Y = y$ cannot occur in such a causal model. This yields constraints on the model parameters beyond those imposed by observations and interventions that can be enforced during causal model training.

It is important to note that we are not saying every causal model should satisfy counterfactual ordering. As in all works on causal inference, it is ultimately up to the analyst to decide if such an assumption appears reasonable in a given domain. As discussed in the previous section, counterfactual ordering appars a reasonable assumption in Epidemiology. In the Experiments section, we empirically demonstrate on data from medicine and finance that models that are trained to satisfy counterfactual ordering comply with domain knowledge, while models that aren't appear in conflict with domain knowledge.

3.3 Counterfactual ordering and Counterfactual Stability

Counterfactual stability has been proposed by Oberst and Sontag [2019] as a different way to restrict the type of counterfactuals a causal model can output, to ensure they are "intutive". We define counterfactual stability then prove a relation between it and counterfactual ordering.

Definition 6 (Counterfactual Stability)**.** *A causal model of categorical variable Y satisfies* counterfactual stability *if it has the following property: If we observe $Y_x = y$, then for all $y' \neq y$, the condition $\frac{P(Y_x=y)}{P(Y_{x'}=y')} \geq \frac{P(Y_x=y')}{P(Y_{x'}=y)}$ implies that $P(Y_x = y' | Y_{x'} = y) = 0$. That is, if we observed $Y = y$ under intervention $X = x$, then the counterfactual outcome under intervention $X = x'$ cannot be equal to $Y = y'$ unless the multiplicative change in $P(Y_x = y)$ is less than the multiplicative change in $P(Y_x = y')$.*

This encodes the following intuition about counterfactuals: If we had taken an alternative action that would have only increased the probability of $Y = x$, without increasing the likelihood of other outcomes, then the same outcome would have occurred in the counterfactual case. Moreover, in order for the outcome to be different under the counterfactual distribution, the relative likelihood of an alternative outcome must have increased relative to that of the observed outcome.

Counterfactual stability is weaker than counterfactual ordering, as it imposes fewer constraints on the model. In fact, counterfactual ordering between intervention and outcome values implies counterfactual stability holds between them:

Theorem 3. *If a causal model satisfies* counterfactual ordering *then it satisfies* counterfactual stability.

3.4 Counterfactual ordering functionally constrains causal mechanisms to be monotonic

Oberst and Sontag [2019] were unable to derive any general functional constraints counterfactual stability places on the causal mechanisms underlying a given causal model. They were only able to compute counterfactuals satisfying it in a single, specific type of causal model. Namely, one where the mechanisms are parameterised using the Gumbel-Max trick. By contrast, we now derive general a functional constraint on causal mechanisms that is equivalent to counterfactual ordering. In the next section we will show how to learn causal models that satisfy this constraint. Thus we are able

to learn causal models that satisfy counterfactual ordering without the need for specific parametric assumptions—such as the Gumbel-Max trick, as was required for counterfactual stability by Oberst and Sontag [2019].

Definition 7 (Monotonicity)**.** *If there exists an ordering on interventions and outcomes:* $\{x_0, x_1, \ldots, x_N\}$, $\{y_0, y_1, \ldots, y_M\}$ *such that* $P(Y_{x_i} = y_k) \geq P(Y_{x_j} = y_k)$ *and* $P(Y_{x_i} = y_h) \leq P(Y_{x_j} = y_h)$ *for all* $i > j$ *and* $k > h$ *then* $Y_x(u) \geq Y_{x'}(u)$ *for all* u*. Equivalently, the events* $\{Y_{x_i} = y_h\} \wedge \{Y_{x_j} = y_l\} =$ *False for all* $i > j$ *and* $h < l$*.*

Theorem 4. *Given an intervention & outcome ordering, counterfactual ordering & monotonicity are equivalent.*

Tian and Pearl [2000] proved that in an SCM with DAG 1a with binary X, Y, where Y is monotonic in X, the probabilities of causation—important counterfactual queries that quantify the degree to which one event was a necessary or sufficient cause of another—can be uniquely identified from observational and interventional distributions. See Appendix D for a definition of the probabilities of causation. We thus have the follow corollary to theorem B.

Corollary 1. *In a counterfactually ordered SCM with DAG 1a and binary* X, Y*, the probabilities of causation are identified from observational and interventional distributions.*

For categorical variables beyond the binary case, it is unknown whether monontonicity implies unique identifiability. However, in this work we are not concerned with counterfactuals being uniquely defined, as long as "non-intuitive" counterfactuals are ruled out. Constraints on the model beyond those imposed by observations and experimental data are said to *partially-identify* counterfactual distributions. In the next section we demonstrate how to learn causal models satisfying counterfactual ordering from data.

3.5 Deep twin networks

We now present *deep twin networks* which combine twin networks with neural networks to learn the causal mechanisms and estimate counterfactuals. Importantly, we will discuss how to ensure the function space learned by the neural network satisfies counterfactual ordering.

Contrary to prior Bayesian network approaches deep twin networks allow us not only to estimate counterfactual probabilities from data but to learn the underlying functions that dictate the interactions between causal variables. Hence we are able to gain a deeper insight on the mechanisms represented in the structural causal model that generate the counterfactuals we want to predict. Moreover we are able to quantify the uncertainty about the outcome by learning the latent noise distribution. Finally, the use of neural networks allow us to gain flexibility and computational advantages not present in previous plug-in estimators. Specifically we will see that our methods admits an arbitrary number and type of confounders Z while estimating counterfactual probabilities, a stark difference to plug-in estimators, such as those presented in Cuellar and Kennedy [2020].

Our approach has two stages, training the neural network such that it learns the counterfactually ordered causal mechanisms that best fit the data, then interpreting it as a twin network on which standard inference is performed to estimate counterfactual distributions. Note that if one can generate counterfactuals, one can also generate interventions.

For clarity, we confine our explanations to the causal structure from Figure 1a where X, Y are categorical variables with $X \in \{1, \ldots, N\}$ and $Y \in \{1, \ldots, M\}$, and Z can be categorical or numerical. Note there can be many Z. Our method can be extended to multiple causes and a single output straightforwardly. To generalize this approach to an arbitrary causal structure, one applies our method to each parent-child structure recursively in the topological specified by the direction of the arrows in the causal structure.

Training deep twin networks: To determine the architecture of our neural network, we start with the causal structure of the SCM we wish to learn, and consider the graphical structure of its twin network representation. Our neural network architecture then exactly follows this graphical structure. This is graphically illustrated for the case of binary X, Y from Figure 1a with twin network in Figure 1c in Figure 2. In the case of binary X, Y, the neural network has two heads, one for the outcome under the factual treatment and the other for the outcome under the counterfactual treatment. Furthermore two shared—but independent of one another—base representations, one corresponding to a

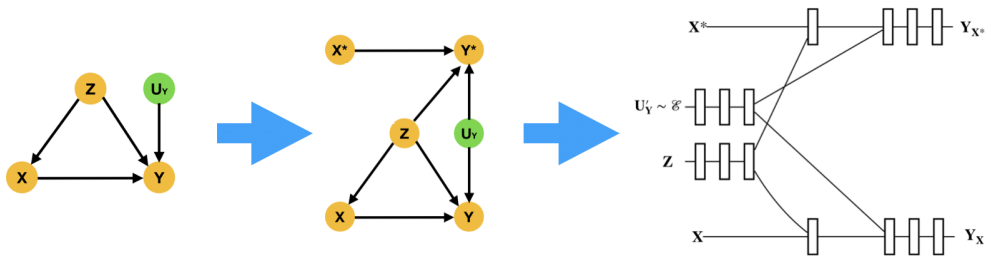


Figure 2: From DAG to twin network DAG to deep neural network (NN) architecture for binary X, Y. Rectangular blocks are NN blocks, like FCN layers or Lattices; forward intersections are concatenation of features.

representation of the observed confounders, Z, and the other to the latent noise term on the outcome, U_Y, are employed. For multiple treatments we have N neural network heads, each corresponding to the categories of X. To interpret this as a twin network for given evidence X, Y, Z and desired intervention X^*, we marginalize out the heads indexed by the elements of $\{1, \ldots, N\}/X, X^*$. To train this neural network, we require two things: 1) a label for head Y^*, and 2) a way to learn the distribution of the latent noise term U_Y.

For 1), we must ask what the expected value of Y^* is, for fixed covariates Z, under a change in input X^*. This corresponds to $\mathbb{E}(Y^*|X^*, Z)$. Given the correspondence between twin networks and the original SCM outlined in Equation (1) from Section 2.3, this corresponds to $\mathbb{E}(Y|do(X), Z)$, which is the expected value of Y under an intervention on X for fixed Z. There are many approaches to estimating this quantity in the literature Shalit et al. [2016], Alaa et al. [2017], Johansson et al. [2016], Shi et al. [2019]. We follow Schwab et al. [2018]due to their methods simplicity and empirical high performance. Any method that computes $\mathbb{E}(Y|do(X), Z)$ can be used, however. In addition to specifying the causal structure, the following standard assumptions are needed to estimate $\mathbb{E}(Y|do(X), Z)$ Schwab et al. [2018]: 1) *Ignorability:* there are no unmeasured confounders; 2) *Overlap:* every unit has non-zero probability of receiving all treatments given their observed covariates. Computing this expectation provides the labels for Y^*.

For 2), consider the following. Formally, the causal structure of Figure 1a has $Y = f(X, Z, U_Y)$ with $U_Y \sim q(U_Y)$ for some q. Without loss of generality Goudet et al. [2018], one can rewrite this as $Y = f(X, Z, g(U'_Y))$ with $U'_Y \sim \mathcal{E}$ and $U_Y = g(U'_Y)$, where $\mathcal{E}$ is some easy-to-sample-from distribution, such as a Gaussian or Uniform. Hence we have reduced learning $q(U_Y)$ to learning function g, whose input corresponds to samples from a Guassian or Uniform. Taken together, this provides a method to train our deep twin network. A summary is provided in Algorithm 1.

Algorithm 1 Training a deep twin network

Input: X: Treatment, Z: Confounders, X^*: Counterfactual Treatment; Y: Outcome; C: DAG of causal structure; I: loss imposing constraint on causal mechanisms
Output: F: trained deep twin network

```
Set F's architecture to match twin network representation of C, as in Figure 2
To obtain label for counterfactual head, estimate E(Y|do(X), Z), yielding training dataset D := {X, X*, Z; Y, Y*}
for x, x*, z; y, y* ∈ D and u_y ~ N(0, 1) do
    y', y'* = F(x, x*, u_y, z)
    Train F by minimizing MSE(y, y') + MSE(y*, y'*) + I(D)
end for
```

Enforcing constraints on the causal mechanisms: There are a few approaches to ensure that the function space learned by a neural network satisfies counterfactual ordering. Recall from Theorem B that such constraints correspond to limits on the type of input-output pairs consistent with the function. One approach is to specify a loss function penalising the network for outputs that violate the constraints, as done in Sill and Abu-Mostafa [1997]. Alternatively, counterexample-guided learning Sivaraman et al. [2020] can be employed, to ensure the trained network does not produce any of these outputs when given the corresponding input. Lastly, as Theorem B equates counterfactual ordering with monotonicity, a recent method uses "look-up tables" Gupta et al. [2016] to enforce monotonicity, and has been implemented in TensorFlow Lattice. Theorem B requires that the treatment and

outcome categorical variables are themselves ordered. One method of ordering our treatment variables is based on their perceived preference, i.e. based on domain knowledge. In the absence of domain knowledge one could look at the Average Treatment Effect (ATE) $=: \frac{1}{N}\sum_i y_1(i) - y_0(i)$, where y_1 represents the treated outcome while y_0 the control outcome, as well as the interventional probabilities $P(Y \mid do(X))$. We then determine an estimate of the relationship governing the treatment and the outcomes by observing the trend of the ATE or $P(Y \mid do(X))$ as we change the treatments. The treatments are then ordered such that their relationship towards the outcomes can be characterized as monotonic. If one does not have access to interventional probabilities, then one can estimate them from observational data Schwab et al. [2018].

Estimating counterfactuals: We can now use the trained model to perform counterfactual inference. The reason our neural network architecture matches the Twin Network structure is that performing Bayesian inference on the neural network explicitly equates to performing counterfactual inference. For Figure 1a, there are two counterfactual queries one can ask: (1) $P(Y_{X=x'} = y' \mid X = x, Y = y, Z = z)$, (2) $P(Y_{X=x} = y, Y_{X=x'} = y' \mid Z = z)$, where any of x, y, z can be the empty set. Recall from Section 2.3 that in a twin network (1) corresponds to $P(Y^* = y' \mid X = x, Y = y, X^* = x')$, and (2) corresponds to $P(Y = y, Y^* = y' \mid X = x, X^* = x')$. Any method of Bayesian inference can then be employed to compute these probabilities, such as Importance or Rejection Sampling, or Variational methods. See Algorithm 2.

Algorithm 2 Deep twin network counterfactual inference

Input: X: Treatment, U_Y: Noise, Z: Confounders, X^*: Counterfactual Treatment; Y: Outcome Y^*: Counterfactual Outcome; F : Trained deep twin network; Q: desired counterfactual query (in this example, $Y_{X=x'} = y' \mid X = x, Y = y, Z = z$))
Output: P(Q): Estimated distribution of Q.

Convert P(Q) to twin network distribution: $P(Y_{X=x'} = y' \mid X = x, Y = y, Z = z) \rightarrow P(Y^* = y' \mid X = x, Y = y, X^* = x')$
Compute $P(Y^* = y' \mid X = x, Y = y, X^* = x')$:
for $x, x', z \in \mathcal{D}_{test}$ **do**
    for $[u_y]_N \sim \mathcal{N}(0, 1), N \in \mathbb{N}$ **do**
        Sample $(\tilde{y}, \tilde{y}^* = F(x, x', u_y, z))$ such that $\tilde{y} = y$
        The frequency of these samples for which $\tilde{y}^* = y'$ yields P(Q)
    end for
end for

4 Related Works

Despite the large body of work using machine learning to estimate interventional queries, which we discuss in detail in the Appendix, relatively little work has explored using machine learning to estimate counterfactual queries.

Machine learning to estimate counterfactual queries: Recent work from Pawlowski et al. [2020] used normalising flows and variational inference to compute counterfacual queries using abduction-action-prediction. A limitation of this work is that identifiability constraints required for the counterfactual queries to be uniquely defined given the training data are not imposed. Work by Oberst and Sontag [2019], expanded by Lorberbom et al. [2021], used the Gumbel-Max trick to estimate counterfactuals, again using abduction-action-prediction. While this methodology satisfies generalisations of the monotonicity constraint, it does so because the Gumbel-Max trick has a limit on the type of conditional distributions it can generate—not because the authors imposed partial-identfiability constraints during the learning process. Hence the Gumbel-Max may not be suitable for the computation of counterfactual queries requiring different (partial-)identifiability constraints. Additional work by Cuellar and Kennedy [2020] devised a non-parametric method to compute the Probability of Necessity using an influence-function-based estimator. This estimator was derived under the assumption of monotonicity. A limitation of this approach is that a separate estimator must be derived and trained for each counterfactual query.

A large body of work address the issue of *partial* identifiability of counterfactuals from data. Historically, this line of work was initiated by Balke and Pearl [1997], who explored bounds on the probabilities of causal queries of binary variables using linear programming. Recently, Zhang and Bareinboim [2020] extended such linear programming derived upper and lower bounds beyond binary outcomes to the case of continuous outcomes. Additional work by Junzhe Zhang [2021] bounded counterfactuals by mapping the SCM space onto a new one that is discrete and easier to

P(T',T)	0.0	1.0	2.0	3.0
0	0	0.0816 ± 0.1414	0.0860 ± 0.1191	0.0344 ± 0.0208
1	0.1439 ± 0.1396	0	0.1156 ± 0.0984	0.1297 ± 0.1306
2	0.1286 ± 0.0726	0.1290 ± 0.1347	0	0.0741 ± 0.0752
3	0.0680 ± 0.0457	0.1854 ± 0.1452	0.0974 ± 0.1140	0

Table 1: **Non-constrained model**. $P(T', T) = P(\text{Risk}_{\text{Account Status}=T'} = \text{good} \mid \text{Account Status} = T, \text{Risk} = \text{bad})$. Columns and rows are Treatments. We observe counter-intuitive probabilities as the lower triangular sub-matrix offers higher probabilities than the upper triangular one. That is, if we observe evidence where bad account status led to bad risk, the non-constrained model predicts an increase in net worth would have led to a *lower* chance of being deemed a good risk—even though all other factors are kept fixed. An un-intuitive resul that conflicts with domain knowledge of the finance industry.

infer upon. Finally, Imbens and Angrist [1994] proposed Local average treatment effects as means of identifications of interventional queries from observational data.

5 Experimentation

We now evaluate our *counterfactual ordering* principle and our *deep twin network* computational tool. We focus on four publicly available real-world datasets, the German Credit Dataset Dua and Graff [2017], the International Stroke Trial (IST) Sandercock and Niewada [2011], the Kenyan Water task Cuellar and Kennedy [2020] and the Twin mortality dataset Louizos et al. [2017]. We further use two synthetic and two semi-synthetic tasks to test our proposed methods. Full dataset description is in Appendix E.

We break our research questions (RQ) into two distinct types: ones that assess our counterfactual ordering principle, and ones that assess the counterfactual estimation accuracy of deep twin networks. Specifically, to asses counterfactual ordering, we wish to determine if not enforcing it leads to domain knowledge conflicting counterfactuals in real datasets, relative to enforcing it. To estimate counterfactual estimation accuracy of deep twin networks satisfying counterfactual ordering, we test on synthetic and semi-sythentic datasets—where we by design have access to the ground truth—as well as on a dataset involving twins, where we use features of one twin as a the counterfactual for the other. Finally, we also test on a real dataset involving binary treatment and outcome. We do this as Corollary 1 showed that in counterfactually ordered causal models with binary variables, certain counterfactual probabilities are uniquely identified from data. Hence for binary variables we can determine how accurate our deep twin network method is relative to these known identified expressions.

- **RQ1:** If counterfactual ordering isn't enforced, do counterfactuals conflict with domain knowledge?
- **RQ2:** By imposing counterfactual ordering, do generated counterfactuals comply with domain knowledge?
- **RQ3:** Can we accurately estimate counterfactual probabilities using deep twin networks?

5.1 Answering RQ1 & RQ2

We investigate the German Credit real-world dataset and explore the International Stroke Trial dataset in the Appendix. We train a deep twin network on German Credit data using algorithm 1. In the Appendix we outline how we determined the monotonicity direction. In Table 2 and Table 1 we estimate the counterfactual probability $P(\text{Risk}_{\text{Account Status}=T'} = \text{good} \mid \text{Account Status} = T, \text{Risk} = \text{bad})$ for a model satisfying counterfactual ordering and an unconstrained model respectively. That is, we ask what the probability that our loan risk would be good if we improved our account status, given that our account status is currently bad and we were just deemed a poor risk of a loan. We note that the unconstrained model offers us non-intuitive probabilities that, when put in context, do not make sense in the real world. We observe that when we condition on evidence

P(T',T)	0.0	1.0	2.0	3.0
0	0	0.3022 ± 0.0415	0.3977 ± 0.0382	0.4040 ± 0.0381
1	0.1079 ± 0.0322	0	0.3891 ± 0.0988	0.4118 ± 0.0545
2	0.0670 ± 0.0156	0.2816 ± 0.0653	0	0.4470 ± 0.0751
3	0.1383 ± 0.0442	0.2953 ± 0.0344	0.3522 ± 0.0577	0

Table 2: **Counterfactual Ordering**. $P(T', T) = P(\text{Risk}_{\text{Account Status}=T'} = \text{good} \mid \text{Account Status} = T, \text{Risk} = \text{bad})$. Columns and rows are Treatments. We observe intuitive results as the lower triangular sub-matrix offers lower probabilities than the upper triangular one. That is, when we observe evidence in which bad account status led to bad risk, the counterfactually ordered model predicts an increase in net worth would have led to a higher chance of being deemed a good risk—an intuitive result that complies with domain knowledge in the finance industry.

	Credit Dataset		IST - Aspirin		IST - Heparin	
F1 Scores	No Constrains Linear Layers	Counterfactual Ordering	No Constrains Linear Layers	Counterfactual Ordering	No Constrains Linear Layers	Counterfactual Ordering
Factual	0.4929	0.8637	0.6113	0.6417	0.3497	0.9758
Counterfactual	0.4698	0.9795	0.7152	0.9501	0.4103	0.9851

Table 3: F1 score of counterfactual predictions for semi-synthetic German Credit Dataset with Treatment: Existing account status, Outcome: Synthetic; & International Stroke Trial (IST) Dataset with Treatment: Aspirin, Outcome: Synthetic; Treatment: Heparin, Outcome: Synthetic. See the Appendix for dataset description.

in which bad account status led to bad risk, the unconstrained model predicts that increasing an individuals net worth would have resulted in a lower probability of being deemed a good risk than *decreasing* their net worth. This result defies common sense, answering RQ1. On the other hand, when we observe bad account status led to bad risk, the counterfactually ordered model predicts that an increase in an individuals net worth would have led to a higher chance of them being deemed a good risk than a decrease in their worth in this context. This result fits with our understanding of the financial industry, answering RQ2.

We observe the same "intuitive" versus "non-intuitive" behavior for the Heparin treatment from the International Stroke Trial dataset in Appendix F.3.

5.2 Answering RQ3

Synthetic data We first evaluate whether we can accurately estimate the probabilities of causation—defined in the Appendix—on synthetic data. We test on data generated by an unconfounded as well as a confounded synthetic causal model, whose functional forms are outlined in the Appendix. Following the algorithms 1, 2 we train a deep twin network on data from each case and enforce monotonicity. In both cases, we show accurate estimation. Results are in Table 3 and Fig. 3 in the Appendix.

Semi-synthetic data In Table 3 we show the results of our semi-synthetic experiments, described in the Appendix, for both the German Credit Datasest as well as the International Stroke Trial. Here, as the outcomes are synthetic, the ground truth is known a priori, hence we are able to calculate the associated F1 scores for each of the models. The counterfactually ordered models are more accurate at predicting both factual and counterfactual outcomes answering RQ3. Moreover, not enforcing counterfactual ordering leads to reduced performance in counterfactual estimation.

Real-world data We show performance on the Twin Mortality data of Louizos et al. [2017] now, and discuss the Kenyan Water task from Cuellar and Kennedy [2020] in the Appendix. In the case of Kenyan Water dataset, treatment ad outcome are binary, so we can compare the deep twin networks estimated counterfactual distributions to the uniquely identified counterfactual distribution via Corollary 1. We report accurate estimation in Table 5 in the Appendix.

In the Twin mortality dataset the goal is to understand the effect being born the heavier of the twins has on mortality one year after birth, given confounders regarding the health of the mother and background of the parents. Previous work addressed this with intervention queries. We use counterfactual queries—specifically the probabilities of causation. We follow Louizos et al. [2017], Yoon et al. [2018]'s preprocessing. As in Louizos et al. [2017], we treat each twin as the counterfactual of their sibling—providing a ground truth reported in 5 in the Appendix. Again, monotonicity is justified here as we do not expect increasing birth weight to lead to reduced mortality.

First, given birth weight and mortality evidence provided by one twin, we aim to estimate the expected counterfactual outcome had their weight been different. That is, compute $\mathbb{E}(\text{Mortality}_{\text{Weight}}|\text{Mortality}^*, \text{Weight}^*, Z)$, where Z are observed confounders. We achieve a counterfactual AUC-ROC of 86% and F1 score of 83%. Louizos et al. [2017] addressed this same question using only used interventional queries. That is, they computed $\mathbb{E}(\text{Mortality}_{\text{Weight}}|Z)$ and only achieve AUC 83%. We thus outperform Louizos et al. [2017]'s AUC by 3%. Full results in Appendix Table 5. Here, by explicitly conditioning on and using the fact that the observed twins had birth weight and mortality, we are able to update our knowledge about the latent noise term of the other twin. Our improved AUC score showed using this allowed more accurate estimation of the "hidden" twins outcome. This cleanly illustrate the difference between interventions and counterfactuals. To give a comparison to prior work, we computed the average treatment effect from our model, yielding $-2.34\% \pm 0.019$ which matches Louizos et al. [2017]—showing our model accurately estimates interventions as well as counterfactuals.

Table 5 in the Appendix reports our estimation of the Probabilities of Causation. Note that no previous work has computed these counterfactual distributions. Despite accurate estimation of the Probability of Sufficiency, and Necessity & Sufficiency, our model underestimates the Probability of Necessity. This can be explained by a large data imbalance regarding the mortality outcome—affecting the Probability of Necessity the most as mortality is the evidence conditioned here. Nevertheless, we correctly reproduce the relative sizes of the Probabilties of Causation, with Probability of Necessity an order of magnitude larger than the others.

6 Conclusions

We motivated and introduced *counterfactual ordering*, a principle that posits desirable properties causal mechanisms should posses. We proved it is equivalent to causal mechanisms being monotonic. To learn such mechanisms, and perform counterfactual inference with them, we introduced *deep twin networks*. We emprically tested our approach on real and semi-synthetic data, achieving *accurate* counterfactual estimation that complies with domain knowledge.

7 Acknowledgments

We would like to acknowledge and thank our sources of funding and support for this paper. Funding for this work was received by Imperial College London and the MAVEHA (EP/S013687/1) project and the UKRI London Medical Imaging & Artificial Intelligence Centre for Value-Based Healthcare (A.V., B.K.). The authors also received GPU donations from NVIDIA.

References

Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3427–3435, 2017.

Ahmed M Alaa, Michael Weisz, and Mihaela Van Der Schaar. Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*, 2017.

Judea Pearl Ang Li. Unit selection based on counterfactual logic. 2019.

Serge Assaad, Shuxi Zeng, Chenyang Tao, Shounak Datta, Nikhil Mehta, Ricardo Henao, Fan Li, and Lawrence Carin Duke. Counterfactual representation learning with balancing weights. In *International Conference on Artificial Intelligence and Statistics*, pages 1972–1980. PMLR, 2021.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *AAAI*, 1994.

Alexander Balke and Judea Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1171–1176, 1997.

Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. Technical report, Columbia University, Stanford University, 2020.

Maria Cuellar and Edward H Kennedy. A non-parametric projection-based estimator for the probability of causation, with application to water sanitation in kenya. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 183(4):1793–1818, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. *arXiv preprint arXiv:2103.11972*, 2021.

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. In *Explainable and interpretable models in computer vision and machine learning*, pages 39–80. Springer, 2018.

Logan Graham, Ciarán M Lee, and Yura Perov. Copy, paste, infer: A robust analysis of twin networks for counterfactual inference. *NeurIPS Causal ML workshop 2019,*, 2019.

Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *The Journal of Machine Learning Research*, 17(1):3790–3836, 2016.

Guido W. Imbens and Joshua D. Angrist. Identification and estimation of local average treatment effects. *Econometrica*, 62(2):467–475, 1994. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/2951620`.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pages 3020–3029, 2016.

Elias Bareinboim Junzhe Zhang, Jin Tian. Partial counterfactual identification from observational and experimental data. 2021.

Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018.

Michael Kremer, Jessica Leino, Edward Miguel, and Alix Peterson. Replication data for: Spring Cleaning: Rural Water Impacts, Valuation, and Property Rights Institutions. 2015. doi: 10.7910/DVN/28063. URL `https://doi.org/10.7910/DVN/28063`.

MJ Kusner, J Loftus, Christopher Russell, and R Silva. Counterfactual fairness. *Advances in Neural Information Processing Systems 30 (NIPS 2017) pre-proceedings*, 30, 2017.

David A Lagnado, Tobias Gerstenberg, and Ro'i Zultan. Causal responsibility and counterfactuals. *Cognitive science*, 37(6):1036–1073, 2013.

Guy Lorberbom, Daniel D Johnson, Chris J Maddison, Daniel Tarlow, and Tamir Hazan. Learning generalized gumbel-max causal mechanisms. *Advances in Neural Information Processing Systems*, 34:26792–26803, 2021.

Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6449–6459, 2017.

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.

Nick Pawlowski, Daniel C Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *arXiv preprint arXiv:2006.06485*, 2020.

Judea Pearl. Probabilities of causation: three counterfactual interpretations and their identification. *Synthese*, 121(1):93–149, 1999.

Judea Pearl. *Causality (2nd edition)*. Cambridge University Press, 2009.

Jonathan G Richens, Ciarán M Lee, and Saurabh Johri. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, 11(1):1–9, 2020.

Peter Sandercock and Anna. Niewada, Maciej; Czlonkowska. International stroke trial database (version 2). Nov 2011. doi: 10.7488/DS/104. URL `https://datashare.ed.ac.uk/handle/10283/128`.

Patrick Schwab, Lorenz Linhardt, and Walter Karlen. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. *arXiv preprint arXiv:1606.03976*, 2016.

Claudia Shi, David M Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *arXiv preprint arXiv:1906.02120*, 2019.

Joseph Sill and Yaser S Abu-Mostafa. Monotonicity hints. 1997.

Aishwarya Sivaraman, Golnoosh Farnadi, Todd Millstein, and Guy Van den Broeck. Counterexample-guided learning of monotonic neural networks. *arXiv preprint arXiv:2006.08852*, 2020.

Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000.

Sam Witty, Kenta Takatsu, David Jensen, and Vikash Mansinghka. Causal inference using gaussian processes with structured latent confounders. In *International Conference on Machine Learning*, pages 10313–10323. PMLR, 2020.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, 31, 2018.

Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Junzhe Zhang and Elias Bareinboim. Bounding causal effects on continuous outcomes. 2020.

Yao Zhang, Alexis Bellot, and Mihaela Schaar. Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics*, pages 1005–1014. PMLR, 2020.

Appendix

A Distinction between twin networks and abduction-action-prediction

As was discussed in the main text, abduction-action-prediction counterfactual inference requires large computational resources. Twin networks were specifically designed to address this difficulty Pearl [2009], Balke and Pearl [1994]. Indeed, consider the following passage from Pearl's "Causality" [Pearl, 2009, Section 7.1.4, page 214]:

> "The advantages of delegating this computation [abduction] to inference in a Bayesian network [i.e., a twin network] are that the distribution need not be explicated, conditional independencies can be exploited, and local computation methods can be employed"

This suggests that the computational resources required for counterfactual inference using twin networks can be less than in abduction-action-prediction. This was put to the test by Graham et al. [2019] and shown empirically to be correct, with their abstract stating

> "twin networks are faster and less memory intensive by orders of magnitude than standard [abduction-action-prediction] counterfactual inference"

A key difference is that in a twin network, inference can be conducted in parallel rather than in the serial nature of abduction-action-prediction. For instance, sampling in twin networks is faster than in the abduction-action-prediction, as twin networks propagate samples simultaneously through the factual and counterfactual graphs—rather than needing to update, store and resample as in abduction-action-prediction. Thus, full counterfactual inference in a twin network can take up to no more than the amount of time sampling takes, while in abduction-action-prediction one incurs the additional cost of reusing samples and evaluating function values in the new mutilated graph. This is potentially advantageous for very large graphs, or for graphs with complex latent distributions that are expensive to sample.

B Theorem Proofs

We now restate and prove all the theorems from section 3.

Theorem. *If counterfactual ordering holds* $P(Y_{x_j} = y_l | Y_{x_i} = y_h) = 0$ *for all* $l > h$ *and* $i > j$.

Proof. First note that $P(Y_{x'} = y' | Y_{x'} = y) = 0$ for any $y' \neq y$ follows from the definition of counterfactuals. The conjunction of this and counterfactual ordering implies $0 = P(Y_{x_i} \geq y_k | Y_{x_i} = y_h) \geq P(Y_{x_j} \geq y_k | Y_{x_i} = y_h)$. As probabilities are bounded below by 0, we have $P(Y_{x_j} \geq y_k | Y_{x_i} = y_h) = \sum_{l>h} P(Y_{x_j} = y_l | Y_{x_i} = y_h) = 0$. Again, as probabilities are non-negative, we have $P(Y_{x_j} = y_l | Y_{x_i} = y_h) = 0$ for all $l > h$ and $i > j$. □

Theorem. *If a causal model satisfies* counterfactual ordering *then it satisfies* counterfactual stability.

Proof. We need to show that when counterfactual ordering holds, $\frac{P(Y_x=y)}{P(Y_{x'}=y')} \geq \frac{P(Y_x=y')}{P(Y_{x'}=y)} \implies P(Y_x = y' | Y_{x'} = y) = 0$. From counterfactual ordering we have $P(Y_x = y) \geq P(Y_{x'} = y)$ and $P(Y_x = y') \leq P(Y_{x'} = y')$. The latter implies $\frac{P(Y_x=y')}{P(Y_{x'}=y')} \leq 1$, which when combined with the former yields: $P(Y_x = y) \geq \frac{P(Y_x=y')}{P(Y_{x'}=y')} P(Y_{x'} = y)$. Additionally, Appendix B says counterfactual ordering implies $P(Y_x = y' | Y_{x'} = y) = 0$, concluding the proof. □

Theorem. *Given an intervention & outcome ordering, counterfactual ordering & monotonicity are equivalent.*

Proof. First we show counterfactual ordering implies monotonicity. From Theorem B, counterfactual ordering implies $\{Y_{x_i} = y_h\} \wedge \{Y_{x_j} = y_l\} =$ False for all $i > j$ and $h < l$, as $P(Y_{x_i} = y_h | Y_{x_j} = y_l) = 0$. Monotonicity follows.

Next we show monotonicity implies counterfactual ordering. Monotonicity implies that for intervention $X = x$, the likelihood of the outcome being higher in the outcome ordering increases, while the likelihood of the outcome being lower in the ordering decreases relative to the likelihoods imposed by intervention $X = x'$ which lies lower in the intervention ordering. All that remains is to show that for such interventions, x, x', and outcome $Y = y$ for which $P(Y_x = y) \geq P(Y_{x'} = y)$, it follows that $P(Y_x \geq y | Y_{x^*} = y^*) \geq P(Y_{x'} \geq y | Y_{x^*} = y^*)$ for all counterfactual contexts $\{y^*, x^*\}$.

$P(Y_x \geq y | Y_{x^*} = y^*)$ is computed by first updating $P(U)$ under x^*, y^* and computing $P(Y \geq y)$ in the submodel M_x. That is, it corresponds to the expected value of $P(Y_x(u) \geq y)$ under $u \sim P(U|x^*, y^*)$. From monotonicity one has $Y_x(u) \geq Y_{x'}(u)$ for all u. Hence, for any $U = u$ that results in $Y_{x'}(u) \geq y$, that same $U = u$ yields $Y_x(u) \geq y$, as $Y_x(u) \geq Y_{x'}(u) \geq y$. Hence, as there are at most as many values of U that lead to $Y_x \geq y$ as lead to $Y_{x'} \geq y$, one has $P(Y_x(u) \geq y) \geq P(Y_{x'}(u) \geq y)$ for any $U = u$. This follows because $P(Y_x(u) \geq y) = \sum_{u|Y_x(u) \geq y} P(U = u)$ together with the observation that summands in $\sum_{u|Y_{x'}(u) \geq y} P(U = u)$ are a subset of summands in $\sum_{u|Y_x(u) \geq y} P(U = u)$. Taking expectations under $P(U|y^*, x^*)$ yields the proof. □

C Further Related Work

Machine learning to estimate interventional queries: Recently there has been much interest in using machine learning to estimate interventional conditional distributions. These were aimed at learning conditional average treatment effects: $\mathbb{E}(Y_{X=1} \mid Z) - \mathbb{E}(Y_{X=0} \mid Z)$. Examples include PerfectMatch Schwab et al. [2018], DragonNet Shi et al. [2019], PropensityDropout Alaa et al. [2017], Treatment-agnostic representation networks (TARNET) Shalit et al. [2016], Balancing Neural Networks Johansson et al. [2016]. Each work utilised neural network architectures similar to the one depicted in Figure 2, with sight modifications based on the specific approach. Other machine learning approaches to estimating interventional queries made use of GANs, such as GANITE Yoon et al. [2018] and CausalGAN Kocaoglu et al. [2018], Gaussian Processes Witty et al. [2020], Alaa and van der Schaar [2017], Variational Autoencoders Louizos et al. [2017], and representation learning Zhang et al. [2020], Assaad et al. [2021], Yao et al. [2018].

While our architecture shares similarities with these, there is one main difference. By interpreting our architecture as a twin network in the sense of Balke and Pearl [1994]—as discussed in Section 3.5—and explicitly including an input for the latent noise term U_Y, we can elevate our network from estimating interventional queries to fully counterfactual ones by performing Bayesian inference on it.

D Probabilities of Causation: Definitions

The probabilities of causation are important counterfactual queries that quantify the degree to which one event was a necessary or sufficient cause of another. Recently, variants on these have been used in medical diagnosis Richens et al. [2020] to determine if a patient's symptoms would not have occurred had it not been for a specific disease. Here, the proposition binary variable W is true is denoted $W = 1$, and its negation, $W = 0$, denotes the proposition W is false.

1. **Probability of necessity:**
$$P(Y_{X=0} = 0 \mid X = 1, Y = 1)$$
The probability of necessity is the probability event Y would not have occurred without event X occurring, given that X, Y did in fact occur.
2. **Probability of sufficiency:**
$$P(Y_{X=1} = 1 \mid X = 0, Y = 0)$$
The probability of sufficiency is the probability that in a situation where X, Y were absent, intervening to make X occur would have led to Y occurring.
3. **Probability of necessity & sufficiency:**
$$P(Y_{X=0} = 0, Y_{X=1} = 1 \mid Z)$$
The probability of necessity & sufficiency quantifies the sufficiency and necessity of event X to produce event Y in context Z. As discussed in section 2.2, joint counterfactual probabilities are well-defined.

E Description of Datasets Used

In the German Credit Dataset the treatment is a four-valued variable corresponding to current account status, and the outcome is loan risk. The International Stroke Trial database was a large, randomized trial of antithrombotic therapy after stroke onset. The treatment is a three-valued variable corresponding to heparin dosage, and the outcome is a three-valued variable corresponding to different levels of patient recovery. In both cases we explore semi-synthetic settings, where the treatment and confounders are derived from the original dataset but the outcome is defined in a synthetic fashion. Synthetic outcomes allow us to determine the ground truth counterfactuals and probabilities of causation (see D for definition). We also evaluate our algorithm with real world outcome of the German Credit Dataset.

The Kenyan Water task is to understand whether protecting water springs in Kenya by installing pipes and concrete containers reduced childhood diarrhea, given confounders. First, monotonicity is a reasonable assumption here as protecting a spring is not expected to increase the bacterial concentration and hence increase the incidence of diarrhea. Cuellar and Kennedy [2020] reported a low value for Probability of Necessity here—suggesting that children who developed diarrhea after being exposed to a high concentration of bacteria in their drinking water would have contracted the disease regardless. However, as there is no ground truth here, further studies reproducing this result with alternate methods are required to gain confidence in Cuellar and Kennedy [2020]'s result. We follow the same data processing as in Cuellar and Kennedy [2020], The Kenyan water dataset originates from Kremer et al. [2015] lincenced under a non commercial use clause and with the requirement for secure storage, both conditions have been fulfilled by the authors. The data was preprocessed following Cuellar and Kennedy [2020].

For the Twin Mortality data, two versions were used. First databases provided by Louizos et al. [2017] were processed to remove NaNs. No further processing was administered. This constituted the completely real version of the Twin Mortality dataset. However, as both Louizos et al. [2017] and Yoon et al. [2018] process the their data to create a semi-synthetic task, in the spirit of proper comparison we used the data as processed and provided by Yoon et al. [2018], with no additional processing.

E.1 Unconfounded Synthetic example for RQ3

$$Y = \begin{cases} X & \text{if } U_Y = 0 \\ 0, & \text{if } U_Y = 1 \\ 1, & \text{if } U_Y = 2 \end{cases} \tag{3}$$

Hence

$$P(N) = \frac{P(U_Y = 0)}{P(U_Y = 2) + P(U_Y = 0)} \tag{4}$$

$$P(S) = \frac{P(U_Y = 0)}{P(U_Y = 1) + P(U_Y = 0)} \tag{5}$$

$$P(NS) = P(U_Y = 0) \tag{6}$$

E.2 Confounded Synthetic example for RQ3

$$Y = \begin{cases} X \times Z, & \text{if } U_Y = 0 \\ 0, & \text{if } U_Y = 1 \\ 1, & \text{if } U_Y = 2 \end{cases} \tag{7}$$

where

$$X := U_x \oplus Z$$

Hence

$$P(N) = \frac{P(U_Y = 0)P(Z = 1)}{P(U_Y = 2) + P(U_Y = 0)P(Z = 1)} \tag{8}$$

$$P(S) = \frac{P(U_Y = 0)P(Z = 1)}{P(U_Y = 1) + P(U_Y = 0)} \tag{9}$$

$$P(NS) = P(U_Y = 0)P(Z = 1) \tag{10}$$

Method	U_y	P(N)	P(S)	P(N&S)
Synth Ground Truth	Uniform	0.5	0.5	0.33333
Synth Twin Net	Uniform	0.50214 ± 0.00387	0.50046 ± 0.00631	0.33449 ± 0.00401
Synth w/ Conf Ground Truth	Gaussian	0.54706	0.35512	0.27443
Synth w/ Conf Twin Net	Gaussian	0.54563 ± 0.00276	0.35177 ± 0.00144	0.27207 ± 0.00125

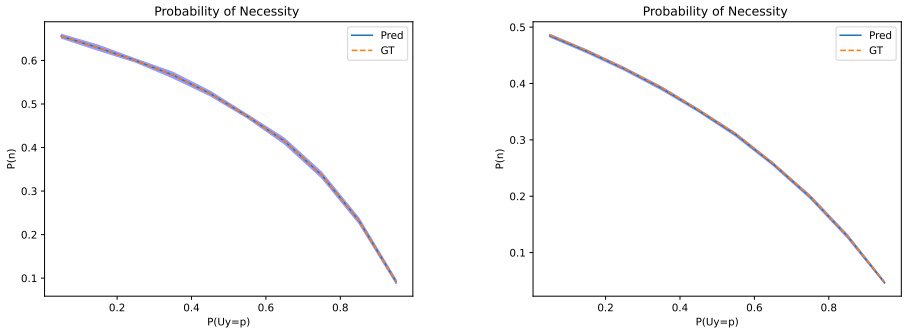
Table 4: Results of Synthetic experiments. P(N): Prob. of Necessity; P(S): Prob. of Sufficiency; P(N&S): Prob. of Necessity and Sufficiency. Our model achieves highly accurate estimations of the probabilities of causation on synthetic data.

F Supplementary Results

F.1 Answering RQ3:

F.1.1 Synthetic data experiments

We test on both unconfounded and confounded causal models, with causal structure from Fig.1a and Fig.1e respectively.The data generation functions are defined in Equations 7 and 3 The functions remain monotonic in X. Given these, we construct synthetic datasets of 200000 points split into training and testing under an $80 - 20$ split. The samples U_y were drawn from either a uniform or a Gaussian distribution, depending on the experiment. Confounders Z were taken from a uniform distribution. We opt for a high number of samples such that we do not bias our analysis due to small sample sizes. In the real world experiments the dataset sizes are smaller. Results for a trained twin network are in 4. We accurately estimate all Probabilities of Causation in both unconfounded and confounded cases when ground truth and candidate distributions are the same. In Figure 3 we also show performance of (a) unconfounded and (b) confounded cases as ground truth distribution of U_Y in synthetic generating functions changes, but candidate training distributions remain fixed—showing robust estimation.



(a) (b)

Figure 3: Predicted & ground truth Probability of Necessity as distribution of U_Y varies in synthetic generating functions, but training distributions do not. Plots show robust estimation. (a) unconfounded, (b) confounded. Errors bars in both

F.1.2 Real world data experiments

Kenyan Water dataset: The Kenyan Water task is to understand whether protecting water springs in Kenya by installing pipes and concrete containers reduced childhood diarrhea. First, monotonicity is reasonable here as protecting a spring is not expected to increase the bacterial concentration and hence increase the diarrhea incidence. Cuellar and Kennedy [2020] reported a low value for Probability of Necessity here—suggesting that children who developed diarrhea after being exposed to a high concentration of bacteria in their drinking water would have contracted the disease regardless. However, as there is no ground truth, further studies reproducing this result with alternate methods are required to gain confidence in Cuellar and Kennedy [2020]'s result. We follow the same data

Method	P(N)	P(S)	P(N&S)	AUC-ROC / F1
KW Median Child *Cuellar et al. 2020*	0.12 ± 0.01	-	-	-
KW TN Median Child	0.13598 ± 0.049	0.09811 ± 0.031	0.31778 ± 0.012	-
KW TN Test Set	0.06273 ± 0.020	0.03914 ± 0.016	0.08521 ± 0.034	-
Twin Mortality Ground Truth	0.33372	0.01011	0.01353	0.83/- *Louizos et al. 2017*
TM TN Test Set	0.12241 ± 0.019	0.01401 ± 0.003	0.01174 ± 0.002	0.86/0.83

Table 5: Results of Kenyan Water (KW) & Twins Mortality (TM) with Twin Network (TN), P(N): Prob. of Necessity; P(S): Prob. of Sufficiency; P(N&S): Prob. of Necessity & Sufficiency. In KW we agree & improve on Cuellar and Kennedy [2020]. In TM we overestimate P(N), but report accurate P(S) & P(N&S), & better AUC than Louizos et al. [2017].

ATE	**0**	**1**	**2**
0	0	0.3059	0.8914
1	-0.3059	0	0.5854
2	-0.8914	-0.5854	0

Table 6: Treatment: Semi-Synthetic Existing account status, Outcome: Synthethic. As the change from 0 to $1\&2$ has a positive ATE, the relationship is increasing monotonic

processing as in Cuellar and Kennedy [2020], detailed in the Appendix. Our findings are in Table 5 of the Appendix and agree with Cuellar and Kennedy [2020] on Probability of Necessity. Moreover, unlike Cuellar and Kennedy [2020], we can also compute Probability of Sufficiency and Probability of Necessity and Sufficiency. We can thus offer a more comprehensive understanding of the role protecting water springs plays in childhood disease. Our results show that exposure to water-based bacteria is not a necessary condition to exhibit diarrhea and it is neither a sufficient, nor a necessary-and-sufficient condition. This provides further evidence that protecting water springs has little effect on the development of diarrhea in children in these populations, indicating the source of the disease is not related to water.

F.2 Determining monotonicity direction

In Table 6 we provide the ATE of a semi-synthetic existing account status from the German Credit Dataset Dua and Graff [2017] with a fully synthetic outcome defined in F.3.1. We observe that for a control treatment 0 changing the treatment to $[1, 2]$ the ATE increases, indicating a monotonic increasing relationship. In addition, we could observe the interventional probabilities where as we increase the value of the treatment the probability of a higher outcome increases, we show this in the Appendix's Table 7. This reinforces our beliefs regarding the type of monotonicity.

Similarly, Tables 8, 9 include the ATE and the interventional probabilities for a real world variant of the above dataset where the treatments are again the current account status of the individual but the outcome is their classification as good or bad risk. At this point, we may call upon our domain knowledge and determine if the break in the monotonic trend is due to noisy observations or a different ordering of the treatments. As this is a real world dataset in which the outcome attribution is inherently noisy we observe an outlier behavior from treatment 1. Upon closer inspection we observe that treatment 1 corresponds to a negative balance in the individuals checking account while treatment 0 indicates no existing checking account. As such one could either switch the treatment ordering to obey the monotonicity, or in the case that this break in monotonicity is suspected to be due to noisy data one could enforce prior knowledge-based monotonicity. Here, we follow our prior knowledge and attribute the break of monotonicity to noise. Our reasoning is based on the fact that an individual without a prior credit account is a larger unknown for a financial institution.

$P(Y\|do(X))$	**0.0**	**1.0**	**2.0**
0	0.6396	0.2056	0.1548
1	0.4635	0.2518	0.2847
2	0.1656	0.2620	0.5723

Table 7: Same data as Table 6. Rows are treatments, and columns are outcomes

ATE	**0**	**1**	**2**	**3**
0	0	-0.0791	0.1029	0.2174
1	0.0791	0	0.1820	0.2965
2	-0.1029	-0.1820	0	0.1145
3	-0.2174	-0.2965	-0.1145	0

Table 8: Treatment: Account status, Outcome: Risk Status

F.3 Answering RQ1, RQ2

F.3.1 Synthetic Outcome For German Credit Score data

$$Y = \begin{cases} X + Z & \text{if } U_Y = 0 \\ 0, & \text{if } U_Y = 1 \\ X * Z, & \text{if } U_Y = 2 \\ 2, & \text{if } U_Y = 3 \\ 1, & \text{if } U_Y = 4 \\ \text{step}(X-1), & \text{if } U_Y = 5 \\ 2 * \text{step}(X-1), & \text{if } U_Y = 6 \end{cases} \tag{11}$$

where the treatment X and the confounders Z span the range $X, Z \in [0, 2]$. Step is the Heaviside step function. In our experimentation U_Y was drawn from a uniform distribution

F.3.2 Synthetic Outcome for Internatinal Stroke Trial

For X the dosage of aspirin treatment, as detailed in the IST Dataset, and counfouders SEX := biological sex of patient, AGE:= age of patient thresholded at 71 years, CONSC := level of consciousness the patient arrived in hospital with. $Y = 1/(1 + e^{-g})$ with g being given by:

$$\begin{aligned} g = X + \text{SEX} &+ 0.2 * (\text{CONSC} - 1) \\ &+ 0.5 * X * \text{SEX} * \text{AGE} + U_y \end{aligned} \tag{12}$$

F.3.3 Treatment: Existing Account Status , Outcome: Risk, switched ordering

In Table 10 we show switching the ordering of treatment 0 and 1 leads to non-intuitive results akin to no constraints.

$P(Y\|do(X))$	**0.0**	**1.0**
0	0.1919	0.8081
1	0.5450	0.4549
2	0.3336	0.6663
3	0.1265	0.8735

Table 9: $P(Y|do(X))$ of the same dataset, rows indicate treatments while columns outcomes

P	0.0	1.0	2.0	3.0
0	0	0.4773 ± 0.0182	0.4243 ± 0.0272	0.3894 ± 0.0147
1	0.6049 ± 0.0386	0	0.5791 ± 0.0340	0.5616 ± 0.0183
2	0.6081 ± 0.0388	0.6025 ± 0.0806	0	0.5221 ± 0.0130
3	0.6265 ± 0.0297	0.6580 ± 0.0431	0.5188 ± 0.0272	0

Table 10: Switched counterfactual ordering – Probability of counterfactual $P(T, T') = P(Y_{X=T'} = 1 \mid X = T, Y = 0)$ – columns and rows are Treatments – We observe counter-intuitive probabilities of necessity as the lower triangular sub-matrix has higher probabilities than the upper triangular

P	1.0	2.0	
0	0	0.1260 ± 0.0070	0.0000 ± 0.0000
1	0.0000 ± 0.0000	0	0.2262 ± 0.0429
2	0.0000 ± 0.0000	0.0000 ± 0.0000	0

Table 11: $P = P(Y_{X=1} = Column | Y_{X=0} = Row)$.

F.3.4 Treatment: Semi-Synthetic Account Status, Outcome: Synthetic

Tables 11,12 show counterfactual probabilities from our method applied to the semi-synthetic account status treatment and synthetic outcome. We note that our model slightly violates the monotonicity constraints by providing non zero probabilities to two cases where they should be 0. However, both of these are within our acceptable experimental error, with one being less than 1%, the other being just over 1%

F.3.5 Treatment: Heparin, Outcome: Synthetic

Tables 13,14 show the counterfactual probabilities for the semi-synthetic heparin treatment and synthetic outcome.

P	0.0	1.0	2.0
0	0.0000	0.1190	0.0079
1	0.0000	0.0000	0.2037
2	0.0000	0.0000	0.0000

Table 12: $P = P(Y_{X=1} = Column | Y_{X=0} = Row)$.

P	1.0	2.0	
0	0	0.0482 ± 0.0006	0.1840 ± 0.0001
1	0.0011 ± 0.0019	0	0.1130 ± 0.0069
2	0.0000 ± 0.0000	0.0135 ± 0.0058	0

Table 13: $P = P(Y_{X=1} = Column | Y_{X=0} = Row)$.

P	0.0	1.0	2.0
0	0.0000	0.0266	0.0917
1	0.0000	0.0000	0.0911
2	0.0000	0.0000	0.0000

Table 14: $P = P(Y_{X=1} = Column | Y_{X=0} = Row)$.