Artificial Intelligence Research and Development
M. Villaret et al. (Eds.)

doi:10.3233/FAIA210162

Deep Causal Graphs for Causal Inference, Black-Box Explainability and Fairness

Álvaro PARAFITA and Jordi VITRIÀ
Departament de Matemàtiques i Informàtica,
Universitat de Barcelona, Spain
parafita.alvaro@ub.edu, jordi.vitria@ub.edu

Abstract. Causal Estimation is usually tackled as a two-step process: identification, to transform a causal query into a statistical estimand, and modelling, to compute this estimand by using data. This reliance on the derived statistical estimand makes these methods *ad hoc*, used to answer one and only one query. We present an alternative framework called Deep Causal Graphs: with a single model, it answers any identifiable causal query without compromising on performance, thanks to the use of Normalizing Causal Flows, and outputs complex counterfactual distributions instead of single-point estimations of their expected value. We conclude with applications of the framework to Machine Learning Explainability and Fairness.

Keywords. causality, counterfactual, causal graph, flow, explainability, fairness

1. Introduction

Artificial Intelligence requires causal knowledge to determine how its actions or predictions may affect or be affected by the outside world. As an example, the association between ice-cream sales and shark attacks might seem causal, when it is in fact due to a latent confounder between them, summer, which makes it spurious. Defining the causal graph that specifies each dependency is just the first step, as we also need an estimation engine to compute the result of causal queries (*i.e.*, "what is the effect of administering the treatment on a patient?"). Causality is also key to black-box Explainability and Fairness, as we can explain the effect of certain input features as interventional effects, or a prediction's fairness on an individual as the counterfactual effect of protected features, such as gender or race.

This relies on our ability to estimate causal queries (*i.e.*, $p(\text{salary} \mid do(\text{education} = \text{undergraduate}))$, meaning, the probability of having a certain salary were we forced to get an undergraduate degree). Estimation usually entails a preprocessing step, *identification*, which transforms our causal query into a statistical estimand that can be estimated with observational data. The problem with this framework is that its models are *ad hoc* to the causal query at hand: were we to ask a new question, we would need to train an additional model. Moreover, most are designed to only estimate the expected value of the target distribution, which does not account for multimodality, skewness, etc.

The contributions of this paper are twofold. Firstly, we introduce **Deep Causal Graphs** (DCG), a sampling-based framework with which we can answer any *identifi-*

able causal queries in a graph with the same trained model. Secondly, we propose **Normalizing Causal Flows**, an implementation of this specification based on Conditional Normalizing Flows, which allows us to model complex continuous distributions that can be integrated in our DCGs. Finally, we provide experiments that showcase the quality of our model's estimations and the applicability of these techniques to the fields of **Machine Learning Explainability and Fairness**. Additionally, we provide a PyTorch library which includes all DCG implementations in this paper and functionality for running experiments on the showcased applications. The code can be found in a public repository (`github.com/aparafita/ccia2021supp`) along with the supplementary material.

2. Related Work

Causal Theory and their applications have mainly been studied from two perspectives: Potential Outcomes [1] and Causal Graphs [2]. Our work focuses on the latter, specially regarding Structural Equation Models (SEM). Nevertheless, the Potential Outcomes perspective is compatible with the Causal Graph literature, and we incorporate some of the findings in [3] in our work.

Deep Causal Graphs model the aforementioned SEMs, but also leveraging the expressive power of deep neural networks. From this point of view, they are directly related to two previous works. CausalGAN [4] represented each random variable as a neural network with their parents' values as the input. Our previous work, Distributional Causal Nodes [5], extended this idea by assuming a known parametric probability distribution for each node. Our current approach subsumes the latter by modelling arbitrary distributions using Normalizing Flows [6], instead of known parametric families. Independent work in [7] also proposes Normalizing Flows for Counterfactual Estimation applied to MRI scans.

In terms of applications, Deep Causal Graphs are specially suited to counterfactual explanations. Counterfactual reasoning has been proposed as an important ingredient for explainability and fairness analysis (*i.e.*, [8,9,10]), but in most of these frameworks, counterfactuals are understood as samples with minimal alterations in the input that change a black-box prediction. This definition does not take into account the causal effects of these alterations on the rest of the variables, therefore providing non-actionable explanations. Our model does work with intervened distributions, circumventing this issue. Additionally, it allows a practical implementation of Counterfactual Fairness [11], which measures the effects of interventions of protected variables on the target variable.

3. Background

We define a **Structural Equation Model** (SEM) as the tuple $M = (V, E, U, P(E), P(U), F)$:

1. $V = \{V_1, \dots, V_K\}$, the measured random variables.
2. $E = \{E_1, \dots, E_K\}$, the exogenous noise variables, one for each V_k.
3. $\emptyset \subseteq U \subseteq \{U_{\{k,l\}}\}_{k,l=1..K,k<l}$, the latent (non-measured) confounder variables $U_{\{k,l\}}$ that explain unmeasured common causes between V_k and V_l.
4. $P(E)$ and $P(U)$, the prior distributions for all non-measured variables.

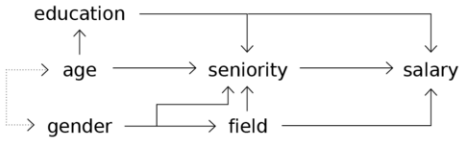

Figure 1. *Salary* dataset causal graph. Bidirectional arrows represent latent confounders.

5. $F = \{f_k = f_k(Pa_k, U_{\{k,.\}}, E_k)\}_{k=1..K}$, the functional relationships[1] $V_k = f_k(.)$ between a node V_k, its measured parent set $Pa_k \subsetneq V$, its parent latent variables $U_{\{k,.\}}$ (if any) and its corresponding E_k.

F implicitly defines a directed graph $G_M = (\mathscr{V} := V \cup E \cup U, \mathscr{E})$ where $\mathscr{E}$, its edges, are defined by all input-output relationships in F: $\mathscr{E} = \bigcup_{k=1..K}\{(V, V_k) \mid V \in Pa_k\} \cup \{(U, V_k) \mid U \in U_{\{k,.\}}\} \cup \{(E_k, V_k)\}$. Any directed edge represents a causal dependency between source/cause and target/effect. A common assumption is that G_M is a Directed Acyclic Graph (DAG). See figure 1 for an example.

3.1. Interventions and Counterfactuals

There are two important operations to consider in Causal Theory. Firstly, the **intervention**, that alters the distribution represented by the model. Specifically, a constant-value intervention, normally represented by $do(X = x)$, means replacing the generative function $X = f_X(.)$ by an assignment $X = x$. This constant value x comes predefined by the intervention and does not depend on the parents of X. Therefore, the intervened SEM replaces f_X by this assignment and the corresponding intervened graph is the subgraph where all edges pointing at X are removed. This subgraph encodes a different distribution, the **intervened distribution**.

Secondly, the **counterfactual**. Given a certain observational sample e' of $E' \subseteq V$ and an intervention $do(X = x)$, a counterfactual is the result of an hypothetical experiment in the past, what would have happened to the values of variables $Y \subseteq V$ had we intervened on X by assigning value x. Counterfactual expressions are of the form $p(Y_x \mid e')$, with Y_x the counterfactual variables under intervention $do(X = x)$. Pearl [2] defines counterfactuals as a three-step process: **abduction**, compute the posterior distribution[2] of the latent variables E and U conditioned on evidence e', $p(E, U \mid e')$; **intervention**, apply the desired intervention $do(X = x)$; **prediction**, compute the required prediction in the intervened, counterfactual model $\widehat{M}$ defined by the abducted priors and the modified set of functions $\widehat{F}$, where f_X has been replaced by $X := x$.

3.2. Identifiability and Structural Causal Models

An essential matter in causal inference is that of a query's **identifiability**. Given a causal query for a certain generative process described by a graph G and a distribution $P(V)$, we say it is identifiable if we can derive an statistical estimand (only using observational terms) for this query using the rules of do-calculus [2]. Two SCMs both representing $P(V)$ with the same causal structure G will output the same result for such a causal query. In

[1] Note that, although f_k is deterministic, the effect of E_k makes it stochastic *w.r.t.* Pa_k, $U_{\{k,.\}}$.

[2] We refer to these new distributions as the *abducted priors*.

other words, if we model an SCM M that represents the same distribution while following the same causal structure as the generative process, it is guaranteed to generate the correct estimations for any identifiable causal query (including counterfactuals), **no matter if the functions in F_M nor the latent priors $P(E), P(U)$ do not match the real generative process.**

In order to determine identifiability, there are automatized solutions for different types of queries: purely interventional queries (*i.e.*, $p(y \mid do(X = x))$) [12], post-intervention conditional queries (*i.e.*, $p(y \mid do(X = x), z)$) [13], and counterfactual queries in an arbitrary number of parallel, counterfactual worlds [14]. As such, one can automatically determine if our techniques can be used for the estimation of a certain causal query. If not, alternatives such as instrumental variables, more restrictive parametric assumptions (when they apply) or randomized experiments (even for the counterfactual case, using the latter referenced work) could circumvent this issue.

4. Method

4.1. Deep Causal Graph

A Deep Causal Graph (DCG) is an abstract specification of the required functionality for a Deep Neural Network to work with causal queries. The only assumption is positivity: $p(v) > 0$ for all v in the domain of V. In other words, all possible configurations of the graph's variables are possible, no matter how unlikely. DCGs model the SEM described in section 3: given a SEM $M = (V, E, U, P(E), P(U), F)$, we represent each random variable in V as a subcomponent called the Deep Causal Unit (DCU), with three operations:

- ***sample***(**parents**): sample a new realization of the variable, given its parents values.
- ***loglk***(**sample**, **parents**): compute the log-likelihood of the sample, given its parents values. This operation must be **differentiable** *w.r.t.* the DCU's parameters.
- ***abduct***(**sample**, **parents**): sample from the abducted noise $(E_k \mid X_k, Pa_k, U_{\{k,.\}})$.

Given these three operations for each node, we can perform estimation across the graph. Assuming nodes in topological order, sampling consists of iteratively applying each node's ***sample*** operation. Any latent variables in E and U can be sampled from their priors $P(E)$ and $P(U)$ directly. Interventions are performed by replacing the *sample* operation with an assignment to the intervened value. However, computing **log-likelihoods** is more nuanced. Given a sample v of variables $\emptyset \subsetneq V' \subseteq V$ (some may be missing), let us define $Z := U \cup \{E_X \in E \mid X \in V \setminus V'\}$. If Z is empty, no variable is missing, which allows us to apply the conditional independencies entailed by the graph G_M (*d-separability*, [2]): $\log p(v) = \sum_{k=1..K} \log p(v_k \mid pa_k)$. If Z is not empty, given a sample $z \sim P(Z)$, we can generate a value for each missing variable in V' deterministically. Then, $\log p(v) = \log \mathbb{E}_Z [p(v \mid Z)] \approx \log \frac{1}{N} \sum_{i=1}^{N} \exp \sum_{k; V_k \in V'} \log p(v_k \mid pa_{k,i}, u_{\{k,.\},i})$, with N i.i.d samples $z_i \sim P(Z)$, from which we can obtain every $u_{\{k,.\},i}$ and every $pa_{k,i}$. In this case, we only compute the log-likelihood of the variables in V', not every measurable variable in the graph. Additionally, we employ the log-sum-exp trick for numerical stability.

The requirement for the DCU's ***loglk*** operation to be differentiable entails that the graph's ***loglk*** is also differentiable. As a result, we can train all nodes simultaneously by Maximum Likelihood Estimation: assuming i.i.d. data, we can train with Stochastic

Gradient Descent by maximizing the average log-likelihood of the dataset. Be warned that, despite being able to compute log-likelihoods of incomplete samples, one should not train with missing data blindly, as the missingness mechanism could induce biases in $P(V)$. An identification of feasible scenarios for training with missing data is left for future work.

The final operation is ***counterfactual***, which, given evidence e of variables $\emptyset \subsetneq E \subseteq V$ and an intervention $do(X = x)$, generates $v_x = (v_x^{(i)})_{i=1}^N \sim P(V_x \mid e)$, N counterfactual samples. To do that, we follow the three-step process defined before: abduction (call each node's *abduct* operation), intervention (apply $do(X = x)$) and prediction (sample each counterfactual node V_x). If there were missing values in our evidence ($E \subsetneq V$) or latent confounders in the graph ($U \neq \emptyset$), we would not have access to every parent's value, which is required by the *abduct* operation. In that case, we use **importance sampling**, which provides us with samples $v_x = (v_x^{(i)})_{i=1}^N$ and corresponding weights $w = (w^{(i)})_{i=1}^N$. These weighted samples allow us to 1) compute counterfactual queries from this distribution, using weighted averages, and 2) study the counterfactual distribution directly, by generating a weighted Bootstrap subsample of (v_x, w). The generation of these samples and weights, along with practical considerations on the implementation of these techniques, is left for the supplementary material due to space restrictions.

4.2. Deep Causal Unit

This subsection discusses two possible implementations of the DCU. Distributional Causal Nodes (DCN) [5] assign a parametric probability distribution to every node V_k (*i.e.*, Gaussians, Exponentials or Categoricals) with parameters Θ_k and model their distribution by defining a neural network f_k that takes the node's parents as input and computes the distribution's parameters Θ_k as the output ($\Theta_k = f(Pa_k, U_{\{k,.\}})$). With this we can perform all three DCU operations: 1) *sample*, by using E_k as an independent noise signal with a reparametrization trick [15] for the assumed distribution; 2) *loglk*, by using the density of the assumed parametric distribution; 3) *abduction*, by inverting the reparametrization formula. This inversion might not always be possible, in which case we could use rejection sampling. In particular, Bernoulli and Categorical distributions can be abducted when using the Gumbel-softmax trick [16] as the sampling step. There are, however, two disadvantages to DCNs. On the one hand, users need to specify a well-matched distribution for each node in the graph, which can be costly on graphs with many variables. On the other hand, standard distributions might not be sufficient to properly adjust complex datasets.

To avoid these two issues, for continuous DCUs, we propose instead **Normalizing Causal Flows** (NCF). A Conditional Normalizing Flow models probability distributions $P(X \mid Z)$ by transforming a continuous random variable X into a base distribution E_X (usually a standard normal distribution) of the same dimension as X, with a parametric function f_Z invertible *w.r.t.* X whose parameters depend on the conditioning variable Z. Here Z consists of the parents values of node X, Pa_X and $U_{\{X,.\}}$; therefore, f allows us to model $P(X \mid Pa_X, U_{\{X,.\}})$, which is precisely what we need in our DCU. Moreover, this invertible function between X and E_X is guaranteed to exist under certain regularizing conditions [6]. The main advantage of this setup is that we can compute $P_X(x \mid z) = P_{E_X}(f_z(x)) \cdot |\det J_{f_z}(x)|$, where $J_{f_z}(x)$ is the Jacobian of f_z on x. Normally, we model this function using a conditioner-transformer architecture (see [6] for more details) with the

Table 1. Metrics on IHDP and Jobs datasets, for the training and test sets. Lower is better.

	IHDP				JOBS			
	$\sqrt{e_{PEHE}}$		e_{ATE}		R_{POL}		e_{ATT}	
	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST
LR_1	$5.8 \pm .3$	$5.8 \pm .3$	$.73 \pm .04$	$.94 \pm .06$	$.22 \pm .00$	$.23 \pm .02$	$\mathbf{.01 \pm .00}$	$.08 \pm .04$
LR_2	$2.4 \pm .1$	$2.5 \pm .1$	$\mathbf{.14 \pm .01}$	$.31 \pm .02$	$.21 \pm .00$	$.24 \pm .01$	$.01 \pm .01$	$.08 \pm .03$
BNN	$2.2 \pm .1$	$2.1 \pm .1$	$.37 \pm .03$	$.42 \pm .03$	$.20 \pm .01$	$.24 \pm .02$	$.04 \pm .01$	$.09 \pm .04$
TAR	$.88 \pm .02$	$.95 \pm .02$	$.26 \pm .01$	$.28 \pm .01$	$.17 \pm .01$	$.21 \pm .01$	$.05 \pm .02$	$.11 \pm .04$
CFR	$\mathbf{.71 \pm .02}$	$\mathbf{.76 \pm .02}$	$.25 \pm .01$	$.27 \pm .01$	$\mathbf{.17 \pm .01}$	$\mathbf{.21 \pm .01}$	$.04 \pm .01$	$.09 \pm .03$
DCG	$1.0 \pm .05$	$1.2 \pm .09$	$.20 \pm .01$	$\mathbf{.25 \pm .02}$	$.22 \pm .01$	$.24 \pm .05$	$.05 \pm .02$	$\mathbf{.04 \pm .01}$
DCG*	$.94 \pm .04$	$1.0 \pm .06$	$.20 \pm .01$	$\mathbf{.23 \pm .02}$	$\mathbf{.11 \pm .02}$	$\mathbf{.12 \pm .04}$	$.05 \pm .01$	$\mathbf{.05 \pm .01}$

conditioner depending on the conditioning input z, while the transformer is an invertible neural network with certain architectural constraints to make this inversion possible and its determinant tractable. As a result, our flow f is capable of: 1) ***sampling*** from $P(X \mid pa_X, u_{\{X,.\}})$ by sampling an $\varepsilon \sim p(E_X)$ and transforming it back to X with $x = f^{-1}_{pa_X, u_{\{X,.\}}}(\varepsilon)$; 2) computing the ***log-likelihood*** of a realization x as described before; 3) computing the $\varepsilon_x \sim p(E_X)$ such that $f_{pa_X, u_{\{X,.\}}}(x) = \varepsilon_x$ (***abduction***).

In summary, any type of conditional Normalizing Flow can be used in a graph to model complex continuous random variables thanks to their high expressiveness, while also avoiding DCN's node-wise distributional assumptions.

5. Experiments

To evaluate our technique, we will study two benchmark datasets: the Infant Health and Development Program (IHDP dataset, [17]) and the study in [18] about National Supported Work (Jobs dataset). We follow the setup and results from [3], focusing on four metrics, two for each dataset, respectively: *estimated Precision on Estimation of Heterogeneous Effect* (e_{PEHE}), *error in Average Treatment Effect* (e_{ATE}), *error in Average Treatment effect for the Treated* (e_{ATT}) and *Policy Risk* ($R_{pol}(\pi_f)$). Both datasets contain many replications of its samples, so that it is possible to obtain a confidence interval of each metric by training a model with each replication. For fairness in comparison, we only train the DCU of the target variable, as the rest of the models do. Details about the experiment setup, model implementation and the actual code can be found in the supplementary material. For reference, training each of our models in a GPU takes on average 16 seconds on IHDP and 28 seconds on Jobs.

Additionally to the base DCG estimation, we consider a variant that computes the counterfactual outcome also using y_f, the factual outcome. When analyzing alternative outcomes *ex post facto* (such as discussing a loan rejection in a bank, and asking for the alternative outcome had we changed a certain variable), we do have access to the factual outcome, and using it results in better estimation of the counterfactual, as we will see. In contrast with DCGs, some of the methods with which we compare do not provide this functionality, since they do not consider the required abduction step in the counterfactual process. We include this extra case to compare the performance of our method with the rest of the benchmark, for cases where such an input is available.

Results can be found in table 1 for: Linear Regression (LR_1), separate LR for each treatment (LR_2), Balancing Neural Network (BNN, [19]), as well as Treatment Agnostic Representation Network (TAR, TARNet) and its variant with balancing regularization, Counterfactual Regression with Wasserstein distance (CFR) (both from [3]). We include our model as DCG and its variant using y_f as DCG*. DCG achieves comparable results with TARNet in $\sqrt{e_{PEHE}}$ in both sets. For e_{ATE}, DCG is the best model (only surpassed by LR_2 in training, but not in test). In the Jobs dataset, DCG achieves comparable results to BNN on the Policy Risk metric, far from the results with TARNet, but not in the case of DCG*, with which we surpass every other method. Finally, e_{ATT} in training seems to be significantly worse than far-simpler methods, but not in test, where DCG surpasses every other model in both its variants.

TARNet and CFR obtain better results in some instances (PEHE and Policy Risk), even though our network structure also uses their bi-headed networks and they are mostly equivalent (except for CFR's balancing regularization). This difference in results can be attributed to the fact that DCGs learn the distribution of these outcomes, in contrast with the other networks, which just return the expected value. This broader objective might hinder the accuracy of the expectation estimation, but it can be worthwhile nevertheless since we can analyze the distribution afterwards, looking for multi-modality, high skewness, etc., as we will see in the next section. Hence, considering this additional feature, we find that our method is comparable to the other two.

6. Applications

The objective of this section is to apply DCGs to Explainability and Fairness of black-box predictors. We created a synthetic gender wage gap dataset containing several mechanisms that create bias against women in terms of salary (see figure 1). *Gender* affects the choice of *field*, due to societal pressures, and *seniority*, due to management bias in promoting men and women. Additionally, we consider a plausible form of selection bias resulting from parents who leave work to take care of their children, with more incidence on women. This process biases the data so that people who are still working while being older (hence more likely to have children) are mostly men. We model this bias with a latent confounder between *gender* and *age*. The final dataset contains 6,479 samples.

We train a DCG with NCFs and Bernoulli-DCNs (depending on each node having continuous or Bernoulli distributions, respectively) with the same configurations as in the experiments. Additionally, we train a Multi-Layer Perceptron (MLP) regressor of *salary*, which will be our black-box. The objective is twofold: explain the predictions of the MLP with a causal perspective (how would our prediction be had we changed any variable) and measure and reduce the Counterfactual Unfairness of these predictions *w.r.t.* gender.

6.1. Machine Learning Explainability

If we want to estimate the effect of gender on the salary prediction, we compute $\mathbb{E}[\widehat{S} \mid do(G = male)] - \mathbb{E}[\widehat{S} \mid do(G = female)]$. For that we sample using each intervention, generating values for every variable in the graph, and then use them as the input of the MLP, obtaining $(\widehat{s}_i)_{i=1..n}$. The averages of these samples approximate each expectation, which allows us to compute the desired effect: \$3,549. In contrast, were we to estimate the observational query $\mathbb{E}[\widehat{S} \mid G = male] - \mathbb{E}[\widehat{S} \mid G = female]$, we would get \$4,878.

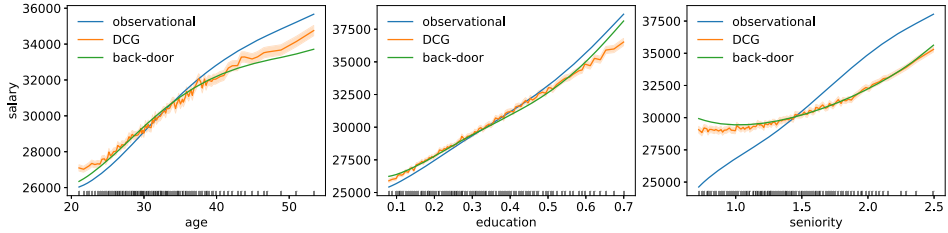

Figure 2. Observational (blue) and interventional effect of three continuous variables on *salary*. The latter is estimated with DCGs (orange) and with the Back-Door Adjustment formula (green) for comparison.

Next, we will study the effects of several continuous variables on salary. Most causal estimation methods argue that their techniques can be easily extended to the continuous case, but many fail to provide examples of this. Figure 2 contains the effect of *age*, *education* and *seniority* on *salary*. In orange, we plot the expected value for each intervention (each x value) with a 95% confidence interval (generated by sampling 1,000 points per intervention and aggregating them). In blue, we include the observational effect ($p(salary \mid X)$) for every variable as a reference to compare between interventional and observational effects. In green, we estimate each interventional effect with an estimand derived from the Back-Door Adjustment formula. Both the blue and green curves are fitted using a 5-degree polynomial basis with Linear Regression. Both interventional curves should and do match, with the only exception of the outlying values (notice the ticks on the x-axis), which is to be expected. Nevertheless, note that with the usual method, we need a different *ad hoc* model for each new query, in contrast with DCGs, which are trained once and can be used on all (identifiable) queries. No matter how complex the estimand might be, DCGs will always be trained equally (using MLE) and will estimate any identifiable query using the same procedures.

Finally we study counterfactuals, to explain predictions on a particular individual. Given evidence e, what would the predicted salary $\hat{S}$ be had we intervened with $do(X = x)$. This kind of query is normally answered as an expectation; however, since we can sample from the counterfactual distribution, we will plot its density instead. Figure 3 shows counterfactual distributions based on evidence {woman, 30 years old, $30,000 annual salary}. On the left, we intervene by changing her gender and find a bi-modal distribution. The green vertical line represents the average counterfactual salary; note that this average would not inform us of the two modes and provides an unlikely counterfactual outcome. On the right, we intervene on 50 values of age (50 equidistant quantiles of age from 2.5% to 97.5%) and plot the corresponding densities, with colour representing each value of age. In orange we can see the original salary as the factual outcome. As age increases, the *salary* distribution moves to higher values, as one would expect.

6.2. Counterfactual Fairness

To conclude, we will study how to mitigate the unfairness of the MLP regressor. Before, we saw that *gender* did have an effect on *salary* that created a significant gender wage gap. We can also measure it on a specific individual: on sample 4,746, a woman with predicted salary $34,551, the average counterfactual salary had she been a man results in $43,077, a difference of $8,526 (much higher than the average for the population, in her case).

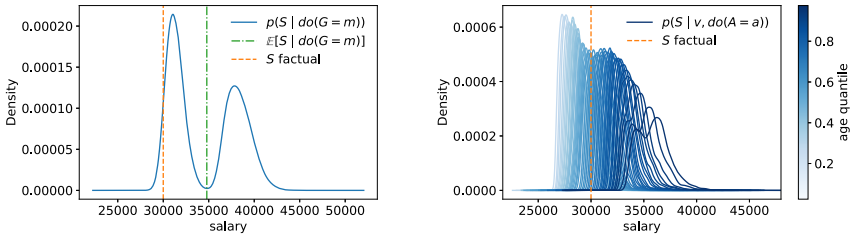

Figure 3. Counterfactual *salary* density curves with interventions on *gender* (left) and *age* (right). Both plots are based on evidence $v := \{\text{woman, 30 years old, \$30,000 annual salary}\}$.

Next, we will train a fairer predictor through Counterfactual Fairness [11], a measure of bias between observational and counterfactual samples. Let us define Counterfactual Unfairness of degree k as $CU_k := \mathbb{E}_V\left[\mathbb{E}_{E,U|V}\left[|Y_{1-x}(E,U) - Y(E,U)|^k\right]\right]$ where X are the intervened (protected) variables, Y the observed target variable and Y_{1-x} the counterfactual target variable. The CU_1 of *salary* (the average unsigned difference between counterfactual and real values) is \$3,883, which shows a significant bias in the model. Note that we can train a differentiable model adding CU_2 as a regularization term; the resulting model (with regularization weight of 10) has a CU_1 of \$234, making it, indeed, fairer. If we test this new regressor on the previous woman, the predicted and counterfactual *salary* estimations become \$38,960 and \$39,379, respectively, with a bias of \$419.

Note that a drop in performance is to be expected, as we are modelling a non-biased distribution, not the original. In this case, it might be preferable to ensure that the resulting ordering (inside each group) matches the one in the dataset. The original predictor, which had an R^2 score of 98%, has a Spearman correlation (*w.r.t.* the original data) of 99% for both genders, while the fair model's correlation is of 90% and 89% for women and men, respectively, even though its R^2 decreases to 68%. Therefore, the choice of metric is essential, since the original data might not reflect the fair world we want to model.

7. Conclusions

We propose Deep Causal Graphs, a flexible and powerful Causal Estimation framework that allows answering any identifiable causal query in a graph by training only once with Maximum Likelihood Estimation. Its fitting capabilities are guaranteed by the use of any Conditional Normalizing Flow and/or Distributional Causal Node. Instead of returning point estimates of causal queries, by its sampling-based nature, it can output complex distributions that result in richer objects of study. Additionally, we showcase many applications of Causal Estimation, powered by DCGs, to the fields of black-box Explainability and Counterfactual Fairness. Further work on alternative implementations of Deep Causal Units or on training with other forms of data (*i.e.*, multivariate node variables, like images, or non-i.i.d. data, like time series) could further extend the applicability of the framework. A PyTorch library is also included for further testing and extensions of the technique.

Acknowledgements

This work has been partially funded by projects RTI2018-095232-B-C21 (MINECO/FEDER), 2017.SGR.1742 (Generalitat de Catalunya) and Universitat de Barcelona's APIF program.

References

[1] Rubin DB. Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association. 2005;100(469):322–331.

[2] Pearl J. Causality. Cambridge university press; 2009.

[3] Shalit U, Johansson FD, Sontag D. Estimating individual treatment effect: generalization bounds and algorithms. In: International Conference on Machine Learning. PMLR; 2017. p. 3076–3085.

[4] Kocaoglu M, Snyder C, Dimakis AG, Vishwanath S. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. In: International Conference on Learning Representations (ICLR); 2018. .

[5] Parafita Á, Vitrià J. Explaining Visual Models by Causal Attribution. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). IEEE; 2019. p. 4167–4175.

[6] Papamakarios G, Nalisnick E, Rezende DJ, Mohamed S, Lakshminarayanan B. Normalizing Flows for Probabilistic Modeling and Inference. arXiv:191202762 [statML]. 2019.

[7] Pawlowski N, Coelho de Castro D, Glocker B. Deep Structural Causal Models for Tractable Counterfactual Inference. Advances in Neural Information Processing Systems. 2020;33.

[8] Wachter S, Mittelstadt B, Russell C. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. Harvard Journal of Law & Technology (Harvard JOLT). 2017;31:841.

[9] Goyal Y, Wu Z, Ernst J, Batra D, Parikh D, Lee S. Counterfactual Visual Explanations. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97; 2019. p. 2376–2384.

[10] Mothilal RK, Sharma A, Tan C. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 607617.

[11] Kusner MJ, Loftus J, Russell C, Silva R. Counterfactual fairness. In: Advances in Neural Information Processing Systems; 2017. p. 4066–4076.

[12] Shpitser I, Pearl J. Identification of joint interventional distributions in recursive semi-Markovian causal models. In: Proceedings of the National Conference on Artificial Intelligence. vol. 21; 2006. p. 1219.

[13] Shpitser I, Pearl J. Identification of conditional interventional distributions. In: 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006; 2006. p. 437–444.

[14] Shpitser I, Pearl J. What counterfactuals can be tested. In: 23rd Conference on Uncertainty in Artificial Intelligence, UAI 2007; 2007. p. 352–359.

[15] Kingma DP, Welling M. Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014. Banff, AB, Canada; 2014. .

[16] Jang E, Gu S, Poole B. Categorical Reparameterization with Gumbel-Softmax. In: 5th International Conference on Learning Representations, ICLR 2017. Toulon, France; 2017. .

[17] Hill JL. Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics. 2011;20(1):217–240.

[18] LaLonde RJ. Evaluating the econometric evaluations of training programs with experimental data. The American economic review. 1986:604–620.

[19] Johansson F, Shalit U, Sontag D. Learning representations for counterfactual inference. In: International conference on machine learning. PMLR; 2016. p. 3020–3029.