Towards Learning and Explaining Indirect Causal Effects in Neural Networks

Abbavaram Gowtham Reddy[1], **Saketh Bachu**[1], **Harsharaj Pathak**[1], **Benin L. Godfrey**[1], **Varshaneya V**[2], **Vineeth N. Balasubramanian**[1], **Satyanarayan Kar**[2]

[1] Indian Institute of Technology Hyderabad, India
[2] Honeywell, Bengaluru, India

Abstract

Recently, there has been a growing interest in learning and explaining causal effects within Neural Network (NN) models. By virtue of NN architectures, previous approaches consider only direct and total causal effects assuming independence among input variables. We view an NN as a structural causal model (SCM) and extend our focus to include indirect causal effects by introducing feedforward connections among input neurons. We propose an ante-hoc method that captures and maintains direct, indirect, and total causal effects during NN model training. We also propose an algorithm for quantifying learned causal effects in an NN model and efficient approximation strategies for quantifying causal effects in high-dimensional data. Extensive experiments conducted on synthetic and real-world datasets demonstrate that the causal effects learned by our ante-hoc method better approximate the ground truth effects compared to existing methods.

1 Introduction

Neural network (NN) models enriched with causal knowledge have demonstrated their ability to achieve robustness (Schölkopf et al. 2021), invariance (Parascandolo et al. 2018; Goyal et al. 2021), and provide interpretable explanations for human understanding (Chattopadhyay et al. 2019; O' Shaughnessy et al. 2020; Kancheti et al. 2022). In training such NN models imbued with causal knowledge, two primary tasks emerge: (1) acquiring a comprehension of causal relationships between input and output neurons (Janzing 2019; Kyono, Zhang, and van der Schaar 2020; Kancheti et al. 2022), and (2) validating and explaining the acquired causal relationships (Chattopadhyay et al. 2019; Janzing, Minorics, and Bloebaum 2020; O' Shaughnessy et al. 2020). Previous studies have tended to address these two tasks separately, despite their close interconnectedness. This separation of dependent tasks also makes it challenging to study and model more nuanced aspects such as the indirect causal effects of input neurons on the output of an NN. To address this limitation, in this work, we propose an Ante-Hoc Causal Explanations (AHCE) approach that simultaneously performs both these tasks.

Task 1 - Learning Causal Effects in NNs: A common practice in learning causal effects in NN models involves considering the NN as a Structural Causal Model (SCM), representing the parametric causal relationships between the features (Kocaoglu et al. 2018; Chattopadhyay et al. 2019; Janzing, Minorics, and Bloebaum 2020). Given our focus on input-output causal relationships in an NN, following (Kocaoglu et al. 2018; Chattopadhyay et al. 2019; Kancheti et al. 2022), we marginalize the hidden layers and view the output as a function of inputs as shown in Fig 1 (a) (the motivating example in the next paragraph describes the variables). It becomes evident that the SCM embodied by a conventional feedforward NN model lacks causal relationships among input features (neurons in the first layer, we use *input features* and *input neurons* interchangeably in this work). Consequently, the causal effects that are learned and quantified are restricted solely to *direct causal effects* (*viz.* causal effects that do not propagate through other input features – see Appendix §A for preliminaries). Hence, there is currently no feasible approach for explaining *indirect causal effects* (*viz.* causal effects that propagate through other input features). We extend the basic architecture of an NN by adding feedforward connections among input neurons (Fig 1(b)) based on domain knowledge of how features interact in the real-world, thus enabling the learning and explaining of indirect causal effects.

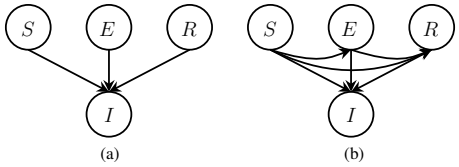

Figure 1: (a) A marginalized NN whose inputs S, E, R are not causally related. (b) A marginalized NN whose inputs are connected through feedforward connections (e.g., $S \rightarrow E$) to capture underlying causal relationships (e.g., S causes E) to learn the indirect causal effects of inputs on output (e.g. effect of S on I via E).

To motivate the need for the study of indirect causal effects in NN models, consider the task of predicting an individual's income (I) using the features: education (E), socioeconomic status (S), and job role (R). In the real world, S causes E and R; E causes R; S, E, and R cause I (Fig 1(b)). However, in an NN model, the relationships among input

Method	Dir. Eff.	Indir. Eff.	Total Eff.	Causal Eff.	Ante-hoc Exp.
IG	✓	✗	✗	✗	✗
CA	✓	✗	✗	✓	✗
CSHAP	✓	✓	✓	✗	✗
CREDO	✓	✗	✓	✓	✓
AHCE	✓	✓	✓	✓	✓

Table 1: Comparison of various explanation methods. IG = Integrated Gradients; CA = Causal Attributions.

features S, E, R are not modeled (Fig 1(a)). As a result, for feature S, an NN model can only learn and explain direct causal effects while neglecting the indirect causal effects on I propagating via E and R. From a fairness standpoint, S should have no direct causal effect on I but can exhibit a non-zero indirect causal effect on I through E and R.

If a model learns a non-zero direct causal effect of S on I, the corresponding model explanations may not align with the real-world and can indicate unacceptable learned causal effects. Thus, learning indirect effects can also find application in identifying and comprehending model biases. We provide the ability to differentiate between direct and indirect causal effects in an NN model by introducing feedforward connections among input features (see Appendix §H for another motivating example).

Task 2 - Explaining Causal Effects in NNs: Explainability methods for NN models have encompassed a wide range of techniques ranging from various gradient-based methods to Shapley values. Recently, there has been increased attention towards causal explanations due to their enhanced reliability (Wachter, Mittelstadt, and Russell 2018; Hendricks et al. 2018), as well as their potential for aiding in debugging (Geva et al. 2022) and improving NN model performance (Kyono, Zhang, and van der Schaar 2020; Kancheti et al. 2022). We refer to explanations such as gradients and Shapley values as *effects* and causal explanations as *causal effects* to separate the non-causal explanations from causal explanations. Most explanation methods provide direct effects, such as gradients and marginal Shapley values (Lundberg and Lee 2017). Causal Shapley values (CSHAP) (Heskes et al. 2020) account for indirect effects mediated through other features. However, they are not equal to the causal effects obtained through backdoor adjustment (Pearl 2009) (see Appendix §B for details). Except for causal regularization using domain priors (CREDO) (Kancheti et al. 2022), all existing efforts in causal explanations are post-hoc approaches, quantifying the causal effects of input features on the output for a pre-trained NN model. These post-hoc explanation methods, though causal, only capture direct effects, and assign zero indirect causal effects to all features. This may not accurately represent the true underlying indirect causal effects among input features in the real world (Janzing, Minorics, and Bloebaum 2020). Although (Kancheti et al. 2022) adopts an ante-hoc approach, it does not model indirect causal effects. See Tab 1 for a comparison of related explanation methods. To the best of our knowledge, this is the first work that that provides an ante-hoc approach to explain indirect causal effects. Our key contributions are summarized below.

- We propose a novel ante-hoc training algorithm to capture indirect causal effects in NN models. Our approach aligns with the demand for intrinsically interpretable techniques rather than post-hoc explanations (Rudin et al. 2021).
- We propose an algorithm to quantify the learned indirect causal effects in NNs using the lateral connections among input neurons.
- We also present effective implementation strategies to scale causal explanation methods to high-dimensional data w.r.t. time and space complexity.
- We present a wide range of empirical results on both synthetic and real-world datasets to showcase the usefulness of the proposed method.

2 Related Work

Learning Structural Causal Models: Learning the structural causal model (SCM) is a core component of tasks in causal inference, including causal effect estimation (Xia et al. 2021), and counterfactual generation (Pawlowski, Coelho de Castro, and Glocker 2020). In a work possibly closest to ours, (Xia et al. 2021) propose the learning of neural causal models (NCM) utilizing the underlying causal graph as an inductive bias, with a specific emphasis on identifying and learning ground truth causal effects. However, our objective is different from NCM; our focus lies in the causal effects pertaining to an NN model, primarily designed to enhance predictive accuracy. Our methodology remains applicable even when only partial knowledge of the underlying causal graph is accessible.

Explainability: In addition to promoting transparency in decision-making processes, the elucidation of NN models serves several purposes, including the identification of concealed biases present in data (Alvarez-Melis and Jaakkola 2017), the revelation of fairness (Došilović, Brčić, and Hlupić 2018), the debugging (Geva et al. 2022) and enhancement of models through explanation-based regularizers (Ross, Hughes, and Doshi-Velez 2017; Rieger et al. 2020; Kancheti et al. 2022). Numerous existing methods for explaining NN models quantify the impact of input features on model outputs using saliency maps (Zeiler and Fergus 2014; Simonyan, Vedaldi, and Zisserman 2013; Selvaraju et al. 2017), local model approximations (Ribeiro, Singh, and Guestrin 2016), approximations of output gradients with respect to inputs (Sundararajan, Taly, and Yan 2017; Smilkov et al. 2017), Shapley values (Lundberg and Lee 2017; Heskes et al. 2020), among others. In this work, we focus on the causal effects of input features on output in an NN model, which can be very useful in safety-critical domains such as healthcare, aerospace, law, and defense. See Appendix §I for a real-world example.

Causal Explanations: By considering an NN as an SCM, assuming that input features are d-separated from each other, (Chattopadhyay et al. 2019) proposed a post-hoc causal explanation method to find the average causal effects (ACE) in a trained NN. However, the assumption of inde-

pendence among inputs limits their ability to consider indirect causal effects. Subsequent studies by (Khademi and Honavar 2020; Yadu, Suhas, and Sinha 2021; Wang et al. 2022; cxp 2019; Goyal et al. 2019a) have followed ACE as defined therein to quantify the learned causal effects. Other causal explanation methods utilize counterfactuals to analyze model behavior under semantically meaningful changes applied to inputs (Verma et al. 2020; Goyal et al. 2019b; Wachter, Mittelstadt, and Russell 2018; Dandl et al. 2020; Van Looveren and Klaise 2021; Mothilal et al. 2021; Mahajan, Tan, and Sharma 2019). However, these methods are commonly employed for qualitative analysis of the model rather than computing causal effects.

Direct and Indirect Explanations: Among existing efforts that explicitly investigate interactions among input variables while computing explanations for NN models, prominent methods are those based on Shapley values (Lundberg and Lee 2017). For instance, in the context of handling missing features in Shapley explanations, it is discouraged to sample from the conditional distribution (rather than the marginal distribution) because the inputs are independent with respect to the causal graph of the NN (Janzing, Minorics, and Bloebaum 2020). While (Heskes et al. 2020) considers both direct and indirect effects motivated by the direct and indirect pathways in the underlying causal graph, even if input neurons of the NN model being explained do not have causal connections, its focus is on providing Shapley values that may not necessarily be causal effects obtained from the adjustment formula (see Appendix §B). We consider input feature interactions while learning and explaining causal effects in NNs. Our approach explicitly estimates and preserves indirect causal effects in an NN model. While (Kancheti et al. 2022) discusses direct and total causal effects for NN model explanations, it does not focus on indirect causal effects. The work most closely related to ours is presented in (Vig et al. 2020), which examined both direct and indirect causal effects in Transformer-based language models for capturing gender bias. However, that study conducted a post-hoc analysis of such models for a different objective, whereas our proposed method represents an ante-hoc approach to learning and explaining both direct and indirect causal effects. Other related work is discussed in Appendix §G.

3 Causal Effects in Neural Networks

Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a causal graph where $\mathbf{V} = \{X_1, X_2, \ldots, X_n, Y\}$ is the set of random variables and $\mathbf{E}$ is the set of edges denoting the causal influences among the variables in $\mathbf{V}$. Let $\mathbf{X} = \{X_1, \ldots, X_n\} = \mathbf{V} \setminus \{Y\}$, $ch(X_i) = \{X_j | X_i \rightarrow X_j\} \subseteq \mathbf{V} \setminus \{X_i, Y\}$ be the set of children of X_i except Y, and $pa(X_i) = \{X_j | X_i \leftarrow X_j\} \subseteq \mathbf{V} \setminus \{X_i, Y\}$ be the set of parents of X_i except Y. This definition of $ch(X_i)$ and $pa(X_i)$ allows us to model indirect effects between input variables. Let $\mathcal{N}$ be an NN model that is trained to predict Y given $\mathbf{X}$ as input by minimizing the empirical loss $\mathcal{R}_{ERM}$ in Eq 1 for a given set $\mathcal{D} = \{(x_1^j, \ldots, x_n^j, y^j)\}_{j=1}^N$.

$$\mathcal{R}_{ERM} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}(y^j, \mathcal{N}(x_1^j, \ldots, x_n^j)) \tag{1}$$

where $\mathcal{L}$ is an appropriate loss function such as root mean squared error for regression and cross-entropy loss for classification. Let $\hat{Y} = \mathcal{N}(X_1, \ldots, X_n)$ be the overall output of the final layer of $\mathcal{N}$. $\mathcal{N}$ can be conceptualized as a directed acyclic graph (DAG) comprising directed edges connecting successive layers of neurons. Consequently, the output $\hat{Y}$ can be understood as the outcome arising from a series of interactions from the first to the final layer. When studying the causal effects of inputs on the output of $\mathcal{N}$, solely the neurons in the first and final layers are considered. Consequently, similar to (Chattopadhyay et al. 2019), we can marginalize the influence of hidden layers within $\mathcal{N}$ and focus solely on the causal structure involving inputs and outputs (see Fig 1 (a)). Note that while we follow (Chattopadhyay et al. 2019) in our view of NN as an SCM, they do not consider or model indirect effects, which is the focus of our work. To this end, we begin by defining various causal effects of input features on the output of a trained NN model.

Definition 3.1. ***(Average Causal Effect in an NN)*** *The Average Causal Effect (ACE) of an input feature X_i at an intervention x_i with respect to a baseline intervention x_i^* on the output $\hat{Y}$ of an NN $\mathcal{N}$ is defined as*

$$ACE_{X_i}^{\hat{Y}} = \mathbb{E}[\hat{Y}|do(X_i = x_i)] - \mathbb{E}[\hat{Y}|do(X_i = x_i^*)]$$

where $do(X_i = x_i)$ denotes an external intervention to the variable X_i with the value x_i (see Defn. A.3 in Appendix A). We use $do(X_i)$ to refer to $do(X_i = x_i)$ when there is no ambiguity. ACE is also called the average total causal effect, which is the sum of direct and indirect causal effects.

Definition 3.2. ***(Average Direct Causal Effect in an NN)*** *The Average Direct Causal Effect (ADCE) measures the causal effect of a feature X_i on the output $\hat{Y}$ of an NN when $\mathbf{Z} = ch(X_i)$ are intervened with values under the baseline intervention $do(X_i = x_i^*)$, denoted by $\mathbf{Z}_{X_i^*}$.*

$$ADCE_{X_i}^{\hat{Y}} = \mathbb{E}[\hat{Y}|do(X_i, \mathbf{Z}_{X_i^*})] - \mathbb{E}[\hat{Y}|do(X_i^*, \mathbf{Z}_{X_i^*})]$$

Definition 3.3. ***(Average Indirect Causal Effect in an NN)*** *The Average Indirect Causal Effect (AICE) measures the causal effect of a feature X_i on the output $\hat{Y}$ of an NN when $\mathbf{Z} = ch(X_i)$ are intervened with values under $do(X_i = x_i)$, denoted by $\mathbf{Z}_{X_i}$, while keeping the X_i value fixed at the baseline intervention $do(X_i = x_i^*)$.*

$$AICE_{X_i}^{\hat{Y}} = \mathbb{E}[\hat{Y}|do(X_i^*, \mathbf{Z}_{X_i})] - \mathbb{E}[\hat{Y}|do(X_i^*, \mathbf{Z}_{X_i^*})]$$

4 Learning and Explaining Direct and Indirect Causal Effects in Neural Networks

We now present our methodology for learning and explaining indirect causal effects within NNs. Following (Shalit, Johansson, and Sontag 2017; Schwab et al. 2020; Zhang, Liu, and Li 2021), we make the following assumption concerning the underlying causal graph $\mathcal{G}$.

Assumption 4.1. *There are no latent (unobserved) confounders in the underlying causal graph $\mathcal{G}$.*

To quantify direct and indirect causal effects of an input X_i on the output $\hat{Y}$ of an NN, it is required to perform an intervention on $ch(X_i)$ with specific values based on X_i's value (as formally stated in Defns 3.2 and 3.3). The above assumption allows us to get the values to perform an intervention on $ch(X_i)$.

Hypothesis 4.1. *In an NN $\mathcal{N}$, the indirect effect of a variable X_i on Y via $ch(X_i)$, $AICE^{\hat{Y}}_{X_i}$, is identifiable in $\mathcal{N}$ iff there are feedforward edges from X_i to $ch(X_i)$ in the architecture of $\mathcal{N}$.*

The supporting proof for the above hypothesis is straightforward and provided in Appendix §C. Note that the edges between X_i and $ch(X_i)$ capture the true causal relationships in the real-world. In such an architecture of $\mathcal{N}$ with lateral edges between X_i and $ch(X_i)$, the weights parametrizing these edges are also learned by $\mathcal{N}$ along with other weights in the model while optimizing for $\mathcal{N}$'s objective.

Although Hypothesis 4.1 may appear self-evident, it has been overlooked in existing methods for explaining NN models. For example, (Janzing, Minorics, and Bloebaum 2020) argue that *Shapley* explanations in a simple feedforward NN should treat all input features to be independent because the causal graph of a simple feedforward NN has no causal connections among input neurons. A similar argument is given by (Datta, Sen, and Zick 2016) focusing on only *direct* effects while quantifying input influence on the output of an NN. Not accounting for indirect effects when modeling statistical relationships in the observed data distribution (e.g., using conditional expectation instead of marginal expectation for missing features while calculating Shapley values) may generate incorrect explanations (Janzing, Minorics, and Bloebaum 2020).

4.1 Learning Indirect Causal Effects

Following the above discussion, given a standard NN $\mathcal{N}$, we propose an augmented NN architecture $\mathcal{N}^{Ind}$ for capturing indirect causal effects of input features on the output. $\mathcal{N}^{Ind}$ contains lateral directed connections among the input neurons based on the available knowledge of the true causal graph (see Fig 2). Our methodology remains applicable even when only a partial causal graph is available, capturing indirect effects exclusively on the available connections. We call the set of NN edges introduced among input neurons as layer 0 connections to separate them from NN connections in hidden layers. These connections among input features have learnable parameters akin to other parameters within the NN.

To train the augmented $\mathcal{N}^{Ind}$ model, we propose an ante-hoc training algorithm consisting of two phases, each of which is invoked sequentially in each epoch. In the first phase, we freeze the parameters of the layer 0 and train the remaining part of the NN. In the second phase, we train the entire model i.e., parameters of layer 0 to the final layer. In the second phase, the input to the $\mathcal{N}^{Ind}$ model is constructed as follows. Consider a specific input data point $(x^j_1, \ldots, x^j_n) \sim \mathcal{D}$. The value of each input variable X_i for which $pa(X_i) = \varnothing$ is taken from $(x^j_1, \ldots, x^j_n)$, and the remaining input feature values are *derived topologically* by feeding the other input variables into layer 0. That is, for each X_i with $pa(X_i) \neq \varnothing$, if f^0_i is the function of its parents $pa(X_i)$ in layer 0, we derive $X_i = f^0_i(pa(X_i))$. Please note that f^0_i is modeled by the NN connections in layer 0. These two training phases are carried out sequentially in every epoch until we reach the desired minimum loss value (or appropriate stopping condition). To aid better learning of parameters of layer 0, we add a regularization term to the empirical loss $\mathcal{R}_{ERM}$ in Eqn 1 that incurs a penalty if the derived feature values deviate from actual feature values in the training data. Eqn 2 shows the overall loss value used in phase 2 with regularization term and corresponding regularization hyperparameter λ. $\mathcal{N}^{Ind}$ is trained using stochastic gradient descent (SGD), as with any other NN model. Algorithm 1 summarizes this training procedure.

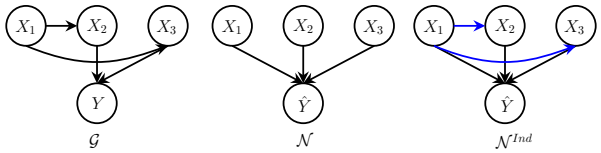

Figure 2: Comparison of the proposed architecture $\mathcal{N}^{Ind}$ with a traditional NN architecture $\mathcal{N}$. $\mathcal{G}$ is the ground truth causal graph. $\mathcal{N}$ and $\mathcal{N}^{Ind}$ differ in input layer such that the inputs in $\mathcal{N}^{Ind}$ are connected (shown in blue color) according to the causal edges in $\mathcal{G}$. In contrast, the inputs in $\mathcal{N}$ are independent. $\mathcal{N}$ and $\mathcal{N}^{Ind}$ may contain edges that are not present in $\mathcal{G}$ due to the feedforward connections from input layer to predictions in NN architecture (e.g., $X_1 \rightarrow \hat{Y}$ is present in $\mathcal{N}$, $\mathcal{N}^{Ind}$ but not in $\mathcal{G}$).

$$\mathcal{R} = \mathcal{R}_{ERM} + \lambda \sum_{j=1}^{N} \sum_{\{\forall i | pa(X_i) \neq \varnothing\}} (x^j_i - f^0_i(pa(x^j_i)))^2 \quad (2)$$

4.2 Explaining Indirect Causal Effects

On training the ante-hoc model $\mathcal{N}^{Ind}$, we now present a methodology to compute the acquired indirect causal effects in the learned model. We begin by formally defining causal effect *identifiability* in this context.

Definition 4.1. ***Causal Effect Identifiability in an NN.*** *The causal effect of an input feature X_i on the output $\hat{Y}$ of an NN is identifiable if $p(\hat{Y}|do(X_i))$ can be computed uniquely from any positive probability distribution $p(X_1, \ldots, X_n, \hat{Y})$.*

Under the *no latent confounding* assumption (Assumption 4.1), following Theorem 3.2.5 and Corollary 3.2.6 of (Pearl 2009), it is easy to show that $ADCE^{\hat{Y}}_{X_i}$ and $AICE^{\hat{Y}}_{X_i}$ are identifiable in $\mathcal{N}^{Ind}$ (we provide formal proofs in Appendix §C). Now, to evaluate $ADCE^{\hat{Y}}_{X_i}$ and $AICE^{\hat{Y}}_{X_i}$ in $\mathcal{N}^{Ind}$, we need to in turn evaluate the following quantities: $\mathbb{E}[\hat{Y}|do(X^*_i, \mathbf{Z}_{X^*_i})]$, $\mathbb{E}[\hat{Y}|do(X_i, \mathbf{Z}_{X^*_i})]$ and $\mathbb{E}[\hat{Y}|do(X^*_i, \mathbf{Z}_{X_i})]$ (see Defns 3.2, 3.3 and recall that $\mathbf{Z} = ch(X_i)$). These terms, which are of

Algorithm 1: Pseudocode for training $\mathcal{N}^{Ind}$ model

Input: True causal graph $\mathcal{G}$, $\mathcal{D} = \{(x_1^j, \ldots, x_n^j, y^j)\}_{j=1}^N$, parameters $\theta_0, \ldots, \theta_m$ of layers $l_0, \ldots, l_m$ of $\mathcal{N}^{Ind}$, λ, functions f_i^0 in l_0 learned by introducing edges among input features.
Output: Trained $\mathcal{N}^{Ind}$ model
for each epoch **do**
  for phase in $[1, 2]$ **do**
    if phase = 1 **then**
      $\mathcal{R}_{ERM} = \frac{1}{N}\sum_{j=1}^{N} \mathcal{L}(y^j, \mathcal{N}^{Ind}(x_1^j, \ldots, x_n^j))$
      Compute gradients of $\mathcal{R}_{ERM}$ w.r.t. $\theta_1, \ldots, \theta_m$
      Update the parameters $\theta_1, \ldots, \theta_m$ using SGD
    else
      for each $(x_1^j, \ldots, x_n^j, y^j)$ in $\mathcal{D}$ **do**
        $x_i^j = f_i^0(pa(x_i^j))\ \forall i\ s.t.\ pa(X_i) \neq \varnothing$
      end for
      $\mathcal{R} = \mathcal{R}_{ERM} + \lambda \sum_{j=1}^{N} \sum_{\{\forall X_i | pa(X_i) \neq \varnothing\}} (x_i^j - f_i^0(pa(x_i^j)))^2$
      Compute gradients of $\mathcal{R}$ w.r.t. $\theta_0, \ldots, \theta_m$
      Update parameters of $\theta_0, \ldots, \theta_m$ using SGD
    end if
  end for
end for
return trained $\mathcal{N}^{Ind}$

the form $\mathbb{E}[\hat{Y}|do(\mathbf{S})]$ where $\mathbf{S}$ is a set of features, often require us to marginalize over other input features $\mathbf{X} \setminus \mathbf{S}$ as:

$$\mathbb{E}[\hat{Y}|do(\mathbf{S})] = \mathbb{E}_{\mathbf{X}\setminus\mathbf{S}}\left[\mathbb{E}[\hat{Y}|\mathbf{S}, \mathbf{X} \setminus \mathbf{S}]\right] \tag{3}$$

Evaluating the above expression, typically using an *adjustment set* (see Defn A.4 in Appendix §A), can incur significant computational overhead, which grows exponentially with the number of features in $\mathbf{X} \setminus \mathbf{S}$, especially when they are continuous and real-valued. To avoid such prohibitive computational requirements, following earlier work (Montavon et al. 2017; Chattopadhyay et al. 2019), we consider the second-order Taylor's approximation to the NN output $\hat{Y} = f(\mathbf{X})$ around the mean vector μ, where $\mu_j = \mathbb{E}[X_j|do(\mathbf{S})]$ as follows:

$$f(\mathbf{X}) \approx f(\mu) + \nabla^T f(\mu)(\mathbf{X} - \mu) + \frac{1}{2}(\mathbf{X} - \mu)^T \nabla^2 f(\mu)(\mathbf{X} - \mu)$$

Taking interventional expectations on both sides gives:

$$\mathbb{E}[f(\mathbf{X})|do(\mathbf{S})] \approx f(\mu) + \frac{1}{2} Tr(\nabla^2 f(\mu)\mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T|do(\mathbf{S})]) \tag{4}$$

The first-order terms vanish because $\mathbb{E}[\mathbf{X}|do(\mathbf{S})] = \mu$. To evaluate Eqn 4, we need to calculate the interventional mean vector $\mu = \mathbb{E}[\mathbf{X}|do(\mathbf{S})]$ and the interventional covariance matrix $\mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T|do(\mathbf{S})]$.

We present the following steps 1 – 4 to evaluate interventional means and covariances for interventions: $do(X_i^*, \mathbf{Z}_{X_i^*})$, $do(X_i, \mathbf{Z}_{X_i^*})$, and $do(X_i^*, \mathbf{Z}_{X_i})$.

1. For an intervention on X_i with the value x_i, set $\mu[i] = x_i$.
2. To get interventional values $\mathbf{Z}_{X_i}$ for the variables in $\mathbf{Z}$ under the intervention $do(X_i = x_i)$, for each variable $X_p \in \mathbf{Z}$ taken in topological order, compute $X_p = f_0^p(pa(X_p))$ and $\mu[p] = \mathbb{E}_{pa(X_p)}\big[\mathbb{E}[X_p|X_i, pa(X_p) \setminus \{X_i\}]\big]$. This step accounts for updating the values of children of X_i based on the intervention on X_i.
3. For each variable $X_q \notin \mathbf{Z}$, set $\mu[q] = \mathbb{E}[X_q]$.
4. Compute the interventional covariance matrix from the interventional data distribution obtained after performing step 2.

After performing the above steps, we can substitute the interventional mean and covariance matrix in Eqn 4 to evaluate the expressions $\mathbb{E}[\hat{Y}|do(X_i^*, \mathbf{Z}_{X_i^*})]$, $\mathbb{E}[\hat{Y}|do(X_i, \mathbf{Z}_{X_i^*})]$, $\mathbb{E}[\hat{Y}|do(X_i^*, \mathbf{Z}_{X_i})]$. An algorithm summarizing this overall procedure of evaluating $ADCE_{X_i}^{\hat{Y}}$, $AICE_{X_i}^{\hat{Y}}$ in $\mathcal{N}^{Ind}$ is provided in Appendix § D.

4.3 Efficient Implementation Strategies

Computation of causal effects, in general, can be compute and memory intensive. We hence also provide a few efficient implementation strategies for such computations, which we also incorporate in our experiments. Let each input $X_i \in \mathbf{X}$ assume one of k possible values ($k = 2$ in the binary case). Evaluating causal effects takes roughly $\mathcal{O}(n^k)$ time because of the marginalization step in Eqn 3, where n is the dimensionality of the input vector $\mathbf{X}$. Evaluating the approximation in Eqn 4 also scales in the order of $\mathcal{O}(n^2)$ as an input intervention may affect all children (Defns 3.2, 3.3). These limitations get accentuated in architectures such as Recurrent Neural Networks (RNNs) (see Appendix F for complexity analysis in RNNs). To address these issues, we propose the following improvements.

Runtime Efficiency using Binning: Computing causal effects using Eqn 4 requires computing the interventional mean and interventional covariance (interventional statistics). To speed up this calculation, we divide the computation into offline and online phases. In the offline step (which can be done independent of the NN training phase), for every data point $\mathbf{X}$ in the training set, we generate and store the interventional statistics for all features $X_i \in \mathbf{X}$ for all interventional values. In the online phase, to find the causal effect for feature X_i with intervention value of x_i in a test data point $\mathbf{X}_{te}$, we first find the data point $\mathbf{X}_{tr}$ in the training set that is most similar to $\mathbf{X}_{te}$. Let α be the value taken by feature X_i in $\mathbf{X}_{tr}$, closest to X_i. We access the interventional statistics stored for $\mathbf{X}_{tr}$ corresponding to feature X_i with intervention α (computed in the offline phase). This retrieved *nearest* interventional statistics is used for causal effect computation. This procedure, detailed further in Appendix § F, reduces significant runtime leveraging offline computations. We refer to this approach as *binning* since a training sample captures a bin and acts as a proxy for other samples/values in its neighborhood. To further speed up ACE computation, we exploit the fact that the Hessian term $\nabla^2 f$ in Eqn 4 can be approximated using $J^T J$ where J is the Jacobian of the NN model function (Gauss-Newton Hessian approximation).

Metric	Feature	IG	CA	CSHAP	CREDO	AHCE
			Synthetic			
RMSE (↓)	W	0.10±0.00	0.09±0.00	0.24±0.00	0.08±0.01	0.04±0.02
	Z	0.11±0.04	0.04±0.01	0.30±0.00	0.06±0.00	0.05±0.00
	X	0.12±0.00	0.11±0.00	0.25±0.01	0.11±0.00	0.10±0.02
	Average	0.11±0.02	0.08±0.00	0.26± 0.00	0.08±0.01	**0.06±0.01**
Frechet (↓)	W	0.25±0.00	0.25±0.00	0.25± 0.05	0.23±0.03	0.14±0.06
	Z	0.19±0.05	0.09±0.05	0.33± 0.02	0.16±0.01	0.13±0.02
	X	0.24±0.07	0.23±0.04	0.32± 0.04	0.26±0.03	0.24±0.03
	Average	0.23±0.04	0.19±0.03	0.30± 0.04	0.22±0.02	**0.17±0.04**
			Auto-MPG			
RMSE (↓)	Num. of Cylinders	0.12±0.00	0.13±0.00	0.20±0.00	0.11±0.02	0.01±0.00
	Displacement	0.11±0.00	0.11±0.00	0.20±0.00	0.09±0.02	0.11±0.01
	Horse Power	0.21±0.02	0.04±0.01	0.17±0.00	0.07±0.02	0.09±0.01
	Weight	0.27±0.04	0.09±0.00	0.05±0.00	0.09±0.02	0.07±0.00
	Acceleration	0.07±0.01	0.07±0.00	0.02±0.00	0.15±0.05	0.07±0.00
	Average	0.16±0.02	0.09±0.00	0.13±0.00	0.10±0.02	**0.07±0.00**
Frechet (↓)	Num. of Cylinders	0.27±0.00	0.25±0.00	0.37±0.00	0.22±0.04	0.03±0.03
	Displacement	0.25±0.00	0.21±0.01	0.38±0.00	0.19±0.03	0.21±0.02
	Horse Power	0.25±0.02	0.07±0.02	0.30±0.00	0.15±0.03	0.18±0.03
	Weight	0.45±0.08	0.15±0.02	0.06±0.02	0.17±0.06	0.09±0.01
	Acceleration	0.12±0.01	0.09±0.01	0.06±0.01	0.33±0.16	0.10±0.00
	Avgerage	0.27±0.02	0.16±0.01	0.23±0.00	0.21±0.06	**0.12±0.02**
			Lung Cancer			
RMSE (↓)	Visit to Asia	0.46±0.05	0.38±0.11	0.00±0.00	0.00±0.00	0.05±0.06
	Tuberculosis	0.62±0.04	1.13±0.04	0.99±0.00	1.00±0.00	0.58±0.29
	Smoking	1.07±0.07	1.01±0.00	0.99± 0.00	1.00±0.00	0.56±0.33
	Lung Cancer	0.40±0.07	0.62±0.02	0.48±0.04	0.49±0.00	0.77±0.75
	Bronchitis	1.55±0.14	1.48±0.06	0.93±0.00	1.08±0.01	1.11±0.51
	Either	0.87±0.18	0.78±0.06	0.53±0.03	0.55±0.00	0.65±0.23
	X-ray	0.11±0.05	0.09±0.04	0.03±0.00	0.00±0.00	0.08±0.12
	Average	0.72±0.09	0.78±0.05	0.56±0.00	0.59±0.00	**0.54±0.33**
			Sachs			
RMSE (↓)	PKC	0.08±0.07	0.10±0.09	0.19±0.00	0.08±0.02	0.12±0.06
	PKA	2.29±1.40	2.19±0.90	0.46±0.00	3.81±0.02	0.65±0.17
	Raf	0.15±0.03	0.11±0.05	0.24±0.00	0.02±0.02	0.12±0.03
	Mek	0.20±0.04	0.21±0.13	0.23±0.00	0.42±0.02	0.14±0.01
	Erk	4.33±3.25	0.63±2.23	0.53±0.00	2.87±0.05	0.51±0.34
	Jnk	0.08±0.04	0.07±0.04	0.25±0.00	0.13±0.05	0.01±0.01
	P38	0.26±0.18	0.09±0.06	0.31±0.00	0.04±0.05	0.02±0.01
	Average	1.05±0.71	0.71±0.27	0.32±0.00	1.05±0.00	**0.22±0.09**
Frechet (↓)	PKC	0.14±0.12	0.13±0.12	0.30±0.00	0.11±0.00	0.17±0.09
	PKA	2.89±1.62	2.97±1.14	0.29±0.00	5.02±0.02	0.91±0.23
	Raf	0.21±0.05	0.16±0.08	0.27±0.00	0.03±0.02	0.17±0.05
	Mek	0.33±0.08	0.27±0.18	0.37±0.00	0.56±0.02	0.17±0.01
	Erk	5.63±4.04	3.12±0.90	0.36±0.00	4.04±0.05	0.70±0.45
	Jnk	0.12±0.06	0.09±0.05	0.36±0.00	0.16±0.05	0.02±0.02
	P38	0.41±0.30	0.12±0.09	0.47±0.00	0.06±0.05	0.02±0.02
	Average	1.39±0.90	0.98±0.36	0.34±0.00	1.43±0.00	**0.31±0.12**

Table 2: Results on Synthetic, Auto-MPG, Lung Cancer, and Sachs Datasets.

Memory Requirements: Storing offline interventional statistics for every sample on the dataset (and corresponding intervention values) quickly becomes impractical, especially for high-dimensional data. To reduce this memory overhead, we use clustering/hashing techniques (KD Tree, DBSCAN) to cluster training data samples, and store interventional statistics for only cluster centers (see Appendix §F for more details of this strategy). From the results shown in Appendix §F, we observe 3 to 10-fold improvements in run time using the proposed binning approach for a slight reduction in the precision of estimated causal effects.

5 Experiments and Results

We conduct experiments on a synthetic dataset, three well-known real-world benchmark datasets, and three industry-based simulated datasets. We compare the causal explanations of AHCE with a post-hoc gradient-based explanation method: Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017), a post-hoc causal explanations (CA) method (Chattopadhyay et al. 2019), the causal Shapley values (CSHAP) (Heskes et al. 2020), and a causal regularization method in (Kancheti et al. 2022). We compare against IG since its explanations can be viewed as individual causal effects (Imbens and Rubin 2015). Ground truth causal effects are computed using the adjustment formula (Eqn 3). Following (Kancheti et al. 2022), we use the Root Mean Squared Error (RMSE) and Frechet distance between true causal effects and the learned explanations. We present our results on total causal effects for a fair comparison with all methods (indirect causal effects do not exist for IG, CA, CREDO). Additional results, including a comparison of CSHAP with our method on indirect causal effects and experimental setup, are presented in the Appendix. Code is available at https://github.com/gautam0707/Learning-and-Explaining-Indirect-Causal-Effects.

Synthetic Data: We create a synthetic dataset using the structural equations: $W \leftarrow Uniform(0,1), Z \leftarrow W/2 + \mathcal{N}(0,0.1), X \leftarrow -W - Z + \mathcal{N}(0,0.1), Y \leftarrow X^3 + \log(Z^2) + \mathcal{N}(0,0.1)$ where W has only indirect causal effect on Y via the paths: $W \rightarrow X \rightarrow Y, W \rightarrow Z \rightarrow X \rightarrow Y$, and $W \rightarrow Z \rightarrow Y$. Z has a direct causal effect on Y via the path $Z \rightarrow Y$ and an indirect causal effect on Y via the path $Z \rightarrow X \rightarrow Y$, and X has only a direct causal effect on Y via the path $X \rightarrow Y$. This dataset has linear equations with additive Gaussian noise among input features W, Z, X, and the output Y is a non-linear function of its inputs with additive Gaussian noise. Hence, for purposes of modeling causal effects, the lateral connections among inputs in $\mathcal{N}^{Ind}$ are obtained using simple linear regressors (for real-world datasets, we replace simple linear regressors with multi-layer perceptrons to account for non-linear relationships among inputs). Tab 2 shows the results. The total causal effects given by our method are closer to ground truth causal effects than baselines. That is, the training algorithm for our ante-hoc causal explanation model can better learn both direct and indirect causal effects.

Auto-MPG: In this experiment, we work on Auto-MPG dataset (Dua and Graff 2017) where the task is to predict *miles per gallon* (MPG) based on various parameters such as *acceleration, horsepower*, etc. We do not know the ground truth causal graph in this case. Hence, we first construct a causal graph based on pertinent domain knowledge (see Appendix). Subsequently, we verify the correctness of this constructed causal graph through interaction with the popular large language model GPT-3.5 (Brown et al. 2020), questioning the correctness of each causal edge within the constructed graph. We use this constructed graph as the available knowledge in our experiments. Tab 2 shows these results. Since the underlying structural equations are unavailable for this dataset, we cannot evaluate indirect causal effects. However, we can compare the performance with respect to total causal effects, which is the sum of direct and indirect causal effects. From the results, our method outperforms baselines in capturing true total causal effects.

Lung Cancer: In Lung Cancer dataset (Scutari and Denis 2014), whose causal graph is known (see Appendix), we consider *Dyspnea* is the output variable with the remaining features such as *smoking, bronchitis*, etc. as inputs. From the results shown in Tab 2, our model is better at learning the true total causal effects when compared to the baselines. The lateral connections among input features are implemented using simple multi-layer perceptrons with non-linear activation functions. Since the underlying causal graph of the Lung Cancer dataset is a discrete Bayesian network with binary-valued features, the Frechet score is not relevant, and so we report only RMSE values for this dataset. Similar to Auto-MPG dataset, we present results on total causal effects.

Sachs: Sachs dataset consists of 11 protein types and their causal relationships. We consider the variable *Akt* as output and the remaining variables as inputs. The results in Tab 2 show that our model is better at learning the true total causal effects than the baselines.

Flight Simulation Datasets: To study the value of our efficient implementation strategies discussed in Sec. 4.3, we consider flight simulation datasets that benefit from such strategies. We consider three different time series-based datasets: *Parking Brake Dataset*, *Flap Dataset* and *Multiple Anomaly Dataset* which simulate the application of parking brakes during the takeoff, the deployment of a wrong flap during takeoff and the multiple brake anomalies (*left-brake, right-brake, and auto-brake*) respectively. These datasets are captured on an industry-grade flight simulator. In all these datasets, we train an RNN to predict whether a given sequence is anomalous. We compare our method with CA and an approximation to the second-order term in Eqn 4 proposed in (Chattopadhyay et al. 2019). Tab A4 of Appendix § E shows the results, highlighting the improvements in time needed to compute ACE in our method.

6 Conclusions

We present a new perspective to learn and quantify causal effects in NNs. Using available prior causal knowledge, we design an ante-hoc causal explanation method to study both direct and indirect causal effects in an NN. We also present effective approximation strategies to compute causal effects for high-dimensional data. Experiments show significant promise of the methodology to elicit direct and indirect causal effects in an NN model.

Acknowledgements

This work was partly supported by the Prime Minister's Research Fellowship (PMRF) program and support from Honeywell through the Govt of India IMPRINT program (earlier the Uchchatar Avishkar Yojana). We are grateful to the anonymous reviewers for their valuable feedback, which improved the presentation of the paper.

References

2019. *CXPlain: Causal Explanations for Model Interpretation under Uncertainty*.

Alvarez-Melis, D.; and Jaakkola, T. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

Chattopadhyay, A.; Manupriya, P.; Sarkar, A.; and Balasubramanian, V. N. 2019. Neural Network Attributions: A Causal Perspective. In *ICML*.

Dandl, S.; Molnar, C.; Binder, M.; and Bischl, B. 2020. Multi-Objective Counterfactual Explanations. In Bäck, T.; Preuss, M.; Deutz, A.; Wang, H.; Doerr, C.; Emmerich, M.; and Trautmann, H., eds., *Parallel Problem Solving from Nature – PPSN XVI*, 448–469. Springer International Publishing.

Datta, A.; Sen, S.; and Zick, Y. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. *2016 IEEE Symposium on Security and Privacy (SP)*, 598–617.

Došilović, F. K.; Brčić, M.; and Hlupić, N. 2018. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0210–0215.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.

Geva, M.; Caciularu, A.; Dar, G.; Roit, P.; Sadde, S.; Shlain, M.; Tamir, B.; and Goldberg, Y. 2022. LM-Debugger: An Interactive Tool for Inspection and Intervention in Transformer-Based Language Models.

Goyal, A.; Lamb, A.; Hoffmann, J.; Sodhani, S.; Levine, S.; Bengio, Y.; and Schölkopf, B. 2021. Recurrent Independent Mechanisms. In *ICLR*.

Goyal, Y.; Feder, A.; Shalit, U.; and Kim, B. 2019a. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*.

Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; and Lee, S. 2019b. Counterfactual Visual Explanations. In *ICML*.

Hendricks, L. A.; Hu, R.; Darrell, T.; and Akata, Z. 2018. Grounding Visual Explanations. *Lecture Notes in Computer Science*, 269–286.

Heskes, T.; Sijben, E.; Bucur, I. G.; and Claassen, T. 2020. Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models. In *NeurIPS*.

Imbens, G. W.; and Rubin, D. B. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press.

Janzing, D. 2019. Causal regularization. In *NeurIPS*.

Janzing, D.; Minorics, L.; and Bloebaum, P. 2020. Feature relevance quantification in explainable AI: A causal problem. In *AISTATS*.

Kancheti, S. S.; Reddy, A. G.; Balasubramanian, V. N.; and Sharma, A. 2022. Matching Learned Causal Effects of Neural Networks with Domain Priors. In *ICML*.

Khademi, A.; and Honavar, V. 2020. A Causal Lens for Peeking into Black Box Predictive Models: Predictive Model Interpretation via Causal Attribution. *arXiv 2008.00357*.

Kocaoglu, M.; Snyder, C.; Dimakis, A. G.; and Vishwanath, S. 2018. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. In *ICLR*.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept Bottleneck Models. *International Conference on Machine Learning*, 5338–5348.

Kyono, T.; Zhang, Y.; and van der Schaar, M. 2020. CASTLE: Regularization via Auxiliary Causal Graph Discovery. In *NeurIPS*.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *NIPS*.

Mahajan, D.; Tan, C.; and Sharma, A. 2019. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. In *CausalML: Machine Learning and Causal Inference for Improved Decision Making Workshop, NeurIPS 2019*.

Montavon, G.; Lapuschkin, S.; Binder, A.; Samek, W.; and Müller, K.-R. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65: 211–222.

Mothilal, R. K.; Mahajan, D.; Tan, C.; and Sharma, A. 2021. Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End. In *AIES*.

O' Shaughnessy, M.; Canal, G.; Connor, M.; Rozell, C.; and Davenport, M. 2020. Generative causal explanations of black-box classifiers. In *NeurIPS*.

Parascandolo, G.; Kilbertus, N.; Rojas-Carulla, M.; and Schölkopf, B. 2018. Learning independent causal mechanisms. In *ICML*.

Pawlowski, N.; Coelho de Castro, D.; and Glocker, B. 2020. Deep Structural Causal Models for Tractable Counterfactual Inference. In *NeurIPS*.

Pearl, J. 2001. Direct and indirect effects. In *UAI*.

Pearl, J. 2009. *Causality*. Cambridge university press.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. ACM.

Rieger, L.; Singh, C.; Murdoch, W.; and Yu, B. 2020. Interpretations are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge. In *ICML*.

Ross, A. S.; Hughes, M. C.; and Doshi-Velez, F. 2017. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In *IJCAI*, 2662–2670.

Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; and Zhong, C. 2021. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. *Statistics Surveys*.

Sayres, R.; Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; and Viegas, F. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, 2668–2677. PMLR.

Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N. R.; Kalchbrenner, N.; Goyal, A.; and Bengio, Y. 2021. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634.

Schwab, P.; Linhardt, L.; Bauer, S.; Buhmann, J. M.; and Karlen, W. 2020. Learning counterfactual representations for estimating individual dose-response curves. In *AAAI*.

Scutari, M.; and Denis, J. 2014. *Bayesian Networks: With Examples in R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

Shalit, U.; Johansson, F. D.; and Sontag, D. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *ICML*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *ICML*.

Van Looveren, A.; and Klaise, J. 2021. Interpretable Counterfactual Explanations Guided by Prototypes. In Oliver, N.; Pérez-Cruz, F.; Kramer, S.; Read, J.; and Lozano, J. A., eds., *Machine Learning and Knowledge Discovery in Databases. Research Track*, 650–665. Cham: Springer International Publishing.

Verma, S.; Boonsanong, V.; Hoang, M.; Hines, K. E.; Dickerson, J. P.; and Shah, C. 2020. Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review.

Vig, J.; Gehrmann, S.; Belinkov, Y.; Qian, S.; Nevo, D.; Singer, Y.; and Shieber, S. 2020. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *NeurIPS*.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2018. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard journal of law and technology*, 31: 841–887.

Wang, Y.; Liu, F.; Chen, Z.; Wu, Y.-C.; Hao, J.; Chen, G.; and Heng, P.-A. 2022. Contrastive-ACE: Domain Generalization Through Alignment of Causal Mechanisms. *IEEE Transactions on Image Processing*, 32: 235–250.

Wickramanayake, S.; Hsu, W.; and Lee, M. L. 2019. FLEX: Faithful Linguistic Explanations for Neural Net Based Model Decisions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2539–2546. AAAI.

Wickramanayake, S.; Hsu, W.; and Lee, M. L. 2021. Comprehensible Convolutional Neural Networks via Guided Concept Learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Xia, K.; Lee, K.-Z.; Bengio, Y.; and Bareinboim, E. 2021. The causal-neural connection: Expressiveness, learnability, and inference. *NeurIPS*.

Yadu, A.; Suhas, P. K.; and Sinha, N. 2021. Class Specific Interpretability in CNN Using Causal Analysis. In *2021 IEEE International Conference on Image Processing (ICIP)*, 3702–3706.

Zarlenga, M. E.; Barbiero, P.; Ciravegna, G.; Marra, G.; Giannini, F.; Diligenti, M.; Shams, Z.; Precioso, F.; Melacci, S.; Weller, A.; Lio, P.; and Jamnik, M. 2022. Concept Embedding Models: Beyond the Accuracy-Explainability Trade-Off. In *Advances in Neural Information Processing Systems*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*, 818–833. Springer.

Zhang, W.; Liu, L.; and Li, J. 2021. Treatment Effect Estimation with Disentangled Latent Factors. In *AAAI*.

Zhou, B.; Sun, Y.; Bau, D.; and Torralba, A. 2018. Interpretable Basis Decomposition for Visual Explanation. In *ECCV*. Springer.