Can Information Flows Suggest Targets for Interventions in Neural Circuits?

Anonymous Author(s)
Affiliation
Address
email

Abstract

In neuroscientific and clinical settings, we often need to examine the flows of information about different stimuli or behaviors, determine which flows are desirable and undesirable, and then identify specific targets for intervention, e.g., through electrical stimulation or surgery, to remove undesirable flows. In such contexts, interventions are difficult to perform and often irreversible, so iterating through candidate interventions is infeasible; however, observation of the system through neural recording modalities is relatively easy. In this paper, we empirically examine whether *observational* measures of information flow can predict the outcomes of *interventions*, using artificial neural networks. We use the context of fairness in machine learning as a model system, and examine information flow about the true label (which is responsible for accuracy, and is hence desirable) as well as information flow about a protected attribute (e.g., gender, which is responsible for bias at the output, and is hence undesirable). Using the recently developed M*-information flow* framework, we measure these two information flows on the edges of a trained neural network, and compare the flow magnitudes against the effect of intervening on those edges by pruning. We show that pruning edges that carry larger information flows about the protected attribute reduces the bias at the output to a greater extent, demonstrating that M-information flow can meaningfully suggest interventions. We also evaluate the bias-accuracy tradeoff for different intervention strategies to analyze how one might use information flows about accuracy and bias jointly, to preserve accuracy while reducing bias. Our results answer the title's question in the affirmative, showing that although the information flow measure we use is *observational*, it allows us to predict the effects of certain *interventions*.

1 Introduction

Consider a neuroscientific experiment to study addiction, in which a mouse is performing a reward-based learning task. To understand and treat the addiction, neuroscientists are trying to determine how information about different signals flow within the "reward circuit" of the mouse's brain [1, 2], e.g., when it is rewarded with water, or when it is rewarded with an addictive substance like cocaine [3]. In this context, one might want to identify two different flows of information in the reward circuit: flows responsible for normal reward-based learning, i.e., improving the "accuracy" of the mouse in this task, and flows responsible for addictive behaviours, which could be said to "bias" the mouse towards certain stimuli.[1] Then, one might envision intervening (using drugs, electrical stimulation, and/or

[1] Evidence for such information flows exists in the neuroscientific literature: for instance, signals responsible for associative learning and "positively-valenced" emotions are useful for reward-based learning; on the other hand, unregulated assignment of "incentive salience" to specific stimuli could result in addiction [4].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

surgery) on targeted brain regions or their interconnections to reduce addiction, while preserving other normal behaviors.

Such examples are common in neuroscience and clinical practice, where we often seek to understand information flow in the brain to identify targets for intervention. Indeed, our current simplistic understanding of such information flows in the reward circuitry (e.g. [5]) is already informing causal interventions in animals [6] and humans [7]. Furthermore, we often care about the information flows of more than one "*message*", in order to make our interventions more specific: we might want to interrupt flows that produce undesirable behaviors, while preserving those that are responsible for desirable behaviors.

One reason we might want to use information flows to inform interventions is that, in both neuroscience and clinical practice, interventions are very expensive: they are difficult to perform in practical terms, and they may also be irreversible—the latter is especially true in clinical settings. As a result, sequentially testing various interventions to identify the optimal one is usually infeasible. On the other hand, there are several well-established neural recording modalities for making *observations*: in fact, there have been rapid advances in non-invasive measurement modalities over the last several decades (such as high density EEG [8], fMRI [9] and NIRS [10]). Therefore, ideally, we would like to use "*observational*" (rather than *interventional*) information flow measures to obtain a clear picture of where we can intervene in a neural circuit to produce a desired effect on behavior.

In this paper, we examine whether such an approach is possible, by evaluating the extent to which information flows can predict the effects of interventions in a trained artificial neural network (ANN). To stay true to our neuroscientific motivations, we seek a machine learning task that has analogous characteristics: desirable and undesirable flows of information, as well as a need for interventions to reduce or remove undesirable behaviors at the output. Coincidentally, we find that the context of *fairness in Machine Learning* [11, 12] shares many of these aspects: we have information flows responsible for improving the accuracy of the network, as well as information flows that produce bias against a protected class (e.g., members of one gender or race). We are also interested in methods to *edit* the network, so as to decrease bias while minimally affecting accuracy. Furthermore, we can easily identify the precise variables (or "*messages*") that are responsible for accuracy and bias, namely, the true label and the protected attribute (respectively). Since information flow about the true label is responsible for accuracy at the output, while information flow about the protected attribute is responsible for bias, the fairness context lends itself perfectly to examining how we can use information flow to inform structured interventions that reduce only bias, without affecting accuracy.

For measuring information flow, we leverage the recently proposed M-information flow framework of Venkatesh et al. [13], which has the distinct advantage of being able to track *multiple* messages as they flow through a neural circuit. Our paper adapts their information flow measure to ANNs, and evaluates this measure in an empirical setting involving multiple messages (to our knowledge, ours is the first paper of this nature). We perform interventions using different strategies to prune nodes or edges of the ANN, and measure the effect of each intervention by quantifying the change in accuracy (Δ_{acc}) or bias (Δ_{bias}) at the output of the network.

Goals and Contributions. Concretely, the main goals and contributions of this paper are as follows:

1. Our primary goal is to study *whether* measuring information flows about a message can inform where we might intervene in an ANN to change how its output is affected by that message. Towards this, our first contribution is an adaptation of the M-information flow framework [13] to ANNs, and developing a quantitative notion of information flow which is meaningful in the ANN context.
2. Secondly, we wish to understand the correlation between the *magnitude* of information flow and the degree to which the intervention affects the output. For this, our contribution is an empirical examination of information flows in an ANN, showing that they can predict the effect of interventions in a fairness context.
3. Lastly, we wish to understand how we can use information flows corresponding to two different messages, to remove undesirable behaviors while preserving desirable ones. Towards this, we compare different intervention strategies that are informed by information flows, and compare the bias-accuracy tradeoffs for each of these strategies.

While the results are presented in a fairness setting, our goal is *not* to find *optimal* tradeoffs between fairness and accuracy, which is the focus of many recent works [14–21]. Instead, the fairness setting

is just a concrete setting for us to understand whether and how an observational measure (namely, M-information flow) can inform interventions in a network with complex flows, including multiple messages.

The rest of the paper is organized as follows: we discuss related work below, following which we summarize the fundamentals of the M-information flow framework in Section 2. We also show how M-information flow can be adapted to ANNs, and discuss the setup of the fairness context. In Section 3, we describe how we empirically evaluate our goals: specifically, we discuss the process of estimating M-information flow and the design of different intervention strategies. Finally, we present our results on synthetic and real datasets in Section 4, and conclude with a discussion of the implications of our results in Section 5.

Related work. The idea of using information flow to understand an ANN (and neural circuits more broadly), could be interpreted as a new type of "explanation" of ANNs. A large number of works have dealt with problems in the fields of explainability, transparency and fairness in machine learning broadly, as well as for artificial neural networks specifically. Molnar [22] provides a good summary of the different approaches taken by many of these methods for explainability. Most of these approaches seek to understand the contribution of individual features [23–26] or individual data points [27, 28] to the output. For ANNs specifically, these can also take the form of visualizations to describe what features or what abstract concepts an ANN has learned [29, 30]. There have also been a number of information theoretic approaches for measuring bias and promoting fairness in AI systems [31–34].

Our approach in this paper is quite different from these prior works: we want to understand what it is about the *network structure itself* that leads to a certain output. We want to understand which edges carry information relevant to classification, as well as information resulting in bias, to the output. We also want to know which edges need to be changed in order to produce a desired output, e.g., fairness towards a protected group with minimal loss of accuracy.

2 Background and Problem Statement

In this section, we provide a brief introduction to the M-information flow framework [13], and show how it can be adapted and reinterpreted for ANNs. We also explain the setup of the fairness problem, and describe how the adapted information flow measure is related to commonly used measures of bias against a protected class.

2.1 Adapting and Reinterpreting M-Information Flow for ANNs

The M-information flow framework provides a concrete way to *define* information flow about a random variable M—called the message—in a general computational system. By changing what M represents, we can understand the flows of any number of different messages (e.g., if Y and Z are two different variables, we can examine Y-information flows and Z-information flows in the same system). The computational system, which is modeled after the brain, is a graph consisting of nodes that compute functions, and edges that transmit the results of these computations between nodes. The proposed definition of M-information flow satisfies an important property: it guarantees the ability to *track* how information about the message M flows from the input to the output of a feedforward neural circuit.[2] Such a model is completely compatible with a feedforward neural network, where the neurons of the ANN act as the nodes, and the weights connecting neurons from different layers act as the edges of the computational system.

A key feature of the computational system model is how it accounts for the inherent stochasticity of the brain as well as its inputs: nodes can generate noise intrinsically, and the transmissions on the edges are considered to be random variables. But in a trained ANN, the computation at every node is deterministic, specified completely by the weights on the incoming edges and the neuron's activation function.[3] However, we can still continue to think of the edges' transmissions as being random variables because the *input data* for the neural network comes from a distribution (which

[2] In the original framework designed for the neuroscientific context, the authors think of the neural circuit as being "feedforward in *time*": i.e., the computational graph is *time-unrolled* in such a way that edges send transmissions from nodes at time t to nodes at time $t+1$.

[3] We avoid constructions that introduce stochasticity within an ANN for the sake of simplicity.

could also be an *empirical* distribution, i.e., a dataset). We are now in a position to state what the original M-information flow definition of Venkatesh et al. [13] looks like in the context of ANNs, before proceeding to adapt it for our purposes.

Definition 1 (Original M-information flow [13])**.** *Let an arbitrary edge of the neural network at layer t be denoted E_t and let the "transmission" on this edge be denoted $X(E_t)$. Similarly, let a subset of edges at layer t be denoted $\mathscr{E}_t'$ and the set of transmissions on this subset be denoted $X(\mathscr{E}_t')$. Then we say that information about M flows on the edge E_t, i.e., edge E_t has M-information flow, if*

$$\exists\, \mathscr{E}_t' \subseteq \mathscr{E}_t \quad s.t. \quad I\big(M; X(E_t) \,\big|\, X(\mathscr{E}_t')\big) > 0, \tag{1}$$

where $\mathscr{E}_t$ is the set of all edges in layer t and $X(\mathscr{E}_t')$ represents the set of transmissions on $X(\mathscr{E}_t')$.

The motivation behind this definition is two-fold: (i) One intuitively expects an edge E_t to carry information flow about M if its transmission $X(E_t)$ *depends* on M, i.e., if $I(M; X(E_t)) > 0$. Definition 1 satisfies this requirement. (ii) However, it is possible for two edges E_t^1 and E_t^2 to carry information about M jointly in such a way that $I(M; X(E_t^1)) = I(M; X(E_t^2)) = 0$, but $I(M; \{X(E_t^1), X(E_t^2)\}) > 0$. Therefore, Definition 1 is designed to assign M-information flow to both E_t^1 and E_t^2 in this case: Venkatesh et al. [13] showed that such an assignment is imperative for consistently *tracking* the information flow about M in a computational system.

To adapt this definition of information flow to ANNs, we start by recognizing that the computational system of Venkatesh et al. [13] allowed each outgoing edge of a given node to carry a different transmission. In an ANN, however, the outgoing edges of a given neuron all carry the *same activation*, but with different weights. Consequently, the random variables representing the transmissions are all scaled versions of each other, and have precisely the same information content. Therefore, we can define information flows for the *activations of every node*, rather than for the transmissions of every edge. We then use the properties of edges (specifically, their weights) to construct a definition of information flow for edges, so as to identify the most important outgoing edges of each node and intervene in a more selective manner. Furthermore, Definition 1 only specifies *whether or not* a given edge has M-information flow. However, we require a *quantification* of M-information flow that will let us compare different nodes or edges, and decide which ones to intervene upon.

Keeping these aspects in mind, we propose M-information flow for the nodes of an ANN, followed by a quantification of M-information flow, and finally a weighted version that assigns different flows to each outgoing edge of a given node:

Definition 2 (M-information flow for ANNs)**.** *Let an arbitrary node of the neural network at layer t be denoted V_t, and let the activations of this node by represented by the random variable $X(V_t)$. Similarly, let an arbitrary subset of nodes at layer t be denoted $\mathscr{V}_t'$ and the set of activations of this subset be denoted $X(\mathscr{V}_t')$. Then, we say that the node V_t has M-information flow if*

$$\exists\, \mathscr{V}_t' \quad s.t. \quad I\big(M; X(V_t) \,\big|\, X(\mathscr{V}_t')\big) > 0. \tag{2}$$

We quantify M-information flow by taking a maximum over all subsets of nodes $\mathscr{V}_t'$ in layer t:

$$\mathscr{F}_M(V_t) \coloneqq \max_{\mathscr{V}_t'} I\big(M; X(V_t) \,\big|\, X(\mathscr{V}_t')\big). \tag{3}$$

Finally, if E_t is an outgoing edge of the node V_t that has weight $w(E_t)$, then we define the weighted M-information flow on that edge as[4]

$$\mathscr{F}_M(E_t) \coloneqq |w(E_t)|\, \mathscr{F}_M(V_t). \tag{4}$$

2.2 Fairness Problem Setup

The central problem in the field of fair machine learning is to understand how we can train models for classification or regression without learning biases present in training data. Recent examples in the literature have shown why algorithms biased against a protected group can be of great concern [12, 35], with the rise of automated algorithms in application domains such as hiring [12], criminal recidivism prediction [36] and predictive policing [37].

[4] Our definition of weighted M-information flow for the edges of the ANN is admittedly heuristic, however it has a simple rationale: an edge with a larger weight has a greater influence on the activation of a target node as compared to an edge with a smaller weight, given the same input to both.

We consider the problem of training artificial neural networks for classification using datasets that have bias in their features and/or labels. The dependencies between the protected attribute (e.g., race, gender, nationality, etc.), the true labels, and the features may be described using a graphical model as shown in Figure 1. We assume that the protected attribute Z influences the features $\{X_i\}$, and possibly the label Y, along with some other latent factors encoded in U. We then train an ANN using the labels and features to acquire the predicted label $\hat{Y}$.

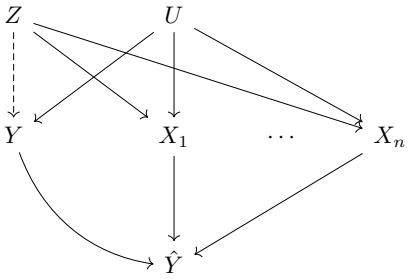

Figure 1: A graphical model representing the causal relationships assumed between the variables used in the fairness setup. The ANN's output $\hat{Y}$ depends on the features $\{X_i\}$ and the true labels Y, which in turn are influenced by the protected attribute Z and some latent variables U. The dashed line from Z to Y indicates that the true labels may or may not be biased.

Recall that our goal is to measure the information flows of two different messages: (i) information flow about the protected attribute, i.e., Z-information flows, which we also refer to as *bias flow*; and (ii) information flow about the true label, i.e., Y-information flows, which we also refer to as *accuracy flow*, as these are responsible for accuracy at the output. The measure of bias we consider at the output is an information theoretic version of statistical (or demographic) parity [38], which has also been used in many previous works [19]. This is because Z-information flow at the output is simply $I(Z;\hat{Y})$, since there are no other edges to condition upon.

3 Estimating Information Flow and Implementing Interventions

In this section, we discuss how we estimate the information flow measure described earlier, and propose a few different intervention strategies for "editing" a trained neural network.

3.1 Estimating Information Flow

Conditional mutual information is a notoriously difficult quantity to estimate [39]. However, in our empirical study, we only consider datasets where the true label Y and the protected attribute Z are binary, which makes estimation considerably easier. We use a classification-based method to estimate the conditional mutual information that appears in Definition 2.

First, we construct separate classifiers for predicting Y and Z from the intermediate activations of a node (or a subset of nodes) in a chosen layer, which we call X^{int}. The generalization accuracy of this classifier indirectly tells us the extent to which information about Z (say) is present in X^{int}. More precisely, if the generalization accuracy of classifying Z from X^{int} is a, that means the expected probability of error in correctly guessing Z from X^{int} is $P_e \coloneqq 1-a$. Then, from Fano's inequality [40, Ch. 2], we have:

$$H(Z \mid X) \leq H_b(P_e) + P_e \log(|\mathcal{Z}| - 1), \tag{5}$$

where H_b is the binary entropy function, and $\mathcal{Z}$ is the sample space of Z. Since Z is binary, $\mathcal{Z} = \{0, 1\}$, hence $|\mathcal{Z}| = 2$. If we further assume $Z \sim \text{Ber}(1/2)$, then $H(Z) = 1$ bit. This simplifies the above equation to:

$$H(Z \mid X^{\text{int}}) \leq H_b(P_e) = H_b(1-a) \tag{6}$$

$$\Rightarrow\ I(Z; X^{\text{int}}) = H(Z) - H(Z \mid X^{\text{int}}) = 1 - H(Z \mid X^{\text{int}}) \geq 1 - H_b(1-a) \tag{7}$$

Therefore, given any classifier that can predict Z from X^{int} with generalization accuracy a, we can compute a lower bound on the mutual information between Z and X^{int}, with a better classifier providing a tighter lower bound. This allows us to compute all conditional mutual information quantities required by Definition 2 using the chain rule [40, Ch. 2]:

$$I(Z; X_1^{\text{int}} \mid X_2^{\text{int}}) = I(Z; X_1^{\text{int}}, X_2^{\text{int}}) - I(Z; X_2^{\text{int}}), \tag{8}$$

where X_1^{int} and X_2^{int} are the activations of two intermediate nodes (or subsets of intermediate nodes) in the ANN. We use both linear and kernel Support Vector Machines for our classifiers. Further details on estimating the mutual information can be found in Appendix A in the supplementary material.

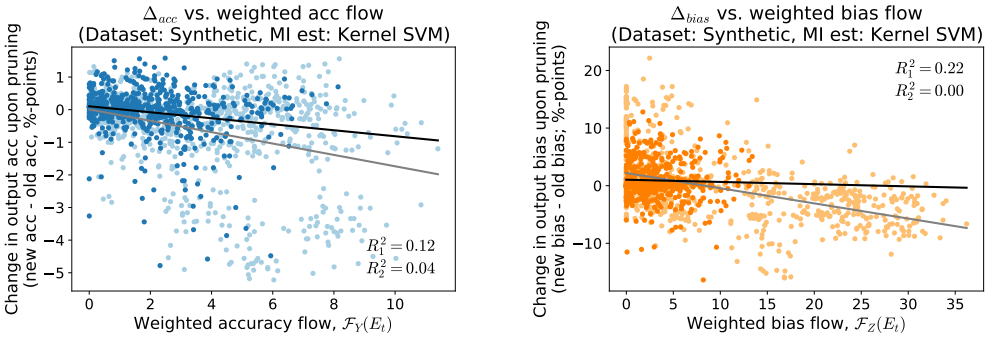

Figure 2: (Left) A plot showing the dependence between the weighted Y-information flow of every edge and the change in accuracy (Δ_{acc}) at the output of the ANN upon pruning that edge, for the synthetic dataset. (Right) The same for weighted Z-information flow and change in bias (Δ_{bias}). In both figures, as the information flow of an edge increases, there is a greater decrease in the accuracy or bias at the output upon pruning that edge. The lighter data points indicate edges in the first layer, while the darker points represent edges in the second (final) layer, with grey and black lines representing linear fits. R_i^2 is the fraction of variance explained by the linear fit, for points in layer i.

3.2 Intervention strategies

We perform interventions on an ANN by pruning its edges. To evaluate bias-accuracy tradeoffs, we use "soft-pruning" of edges, i.e, gradually reducing their weights rather than removing them outright. We compare a number of pruning strategies, which differ based on the following factors:

1. **Pruning method.** We consider three methods: (i) pruning entire *nodes*; (ii) pruning individual *edges*; and (iii) pruning *paths* from the input layer to the output layer. Pruning nodes, i.e., simultaneously decreasing the weights of all outputs of a particular node, can be interpreted as decreasing the dependence of future computations on the derived feature that this node represents. Pruning individual edges has the advantage of being extremely specific, potentially allowing for similar results with minimal changes to the network. Pruning paths can be interpreted as stopping the largest net flow of information from the input features all the way to the output.
2. **Pruning metric.** For each of the above methods, we use two different metrics: the primary metric is based on the ratio of bias flow to accuracy flow, and a secondary metric is the ratio of accuracy flow to bias flow, which we use as a "control" to show that our tradeoff results are not merely a product of chance. When soft-pruning nodes, the primary metric we use is the ratio of flows at the node, i.e. $\mathcal{F}_Z(V_t)/\mathcal{F}_Y(V_t)$. When soft-pruning edges, our primary metric is the *weighted* ratio of flows on the edge, i.e., $|w(E_t)|\,\mathcal{F}_Z(E_t)/\mathcal{F}_Y(E_t)$ (note that we need to re-weight the ratio since weights in $\mathcal{F}_Z(E_t)$ and $\mathcal{F}_Y(E_t)$ would otherwise cancel). Finally, when soft-pruning paths, we score each path using the smallest edge-metric, and choose the path P having the largest score:
$$P^* = \operatorname*{argmax}_{P} \min_{E_t \in P} |w(E_t)| \cdot \mathcal{F}_Z(E_t) \,/\, \mathcal{F}_Y(E_t). \tag{9}$$
3. **Pruning level.** In each of the above cases, we soft-prune the top "k" nodes, edges or paths, which exhibit the largest value of the respective pruning metric. Pruning too much will likely hinder all flows, so we test a few different values of k to find optimal "level" of pruning.

4 Results

4.1 Synthetic Dataset

First, we examine information flows for a small neural network trained on a synthetic dataset. The synthetic dataset is generated in a manner similar to Figure 1. Details of the probability distributions and functions used are provided in Appendix B. The dataset has three continuous-valued features X_1, X_2 and X_3, a binary label Y, and a binary protected attribute Z. Two of the three features, X_1 and X_2, are designed to receive a large causal influence from Z, while X_3 and the labels Y have no influence from Z. Lastly, all three features independently provide information about Y, i.e., they are noisily correlated with Y, with independent noise terms.

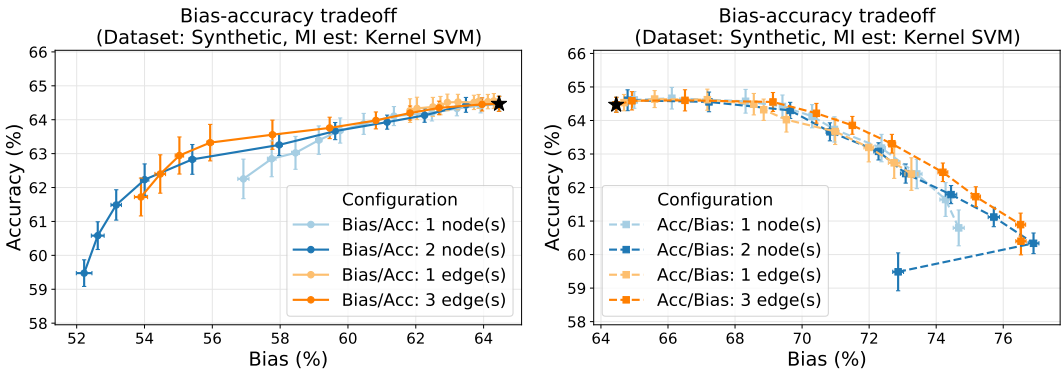

Figure 3: Figures showing the tradeoff between fairness and accuracy when gradually pruning nodes or edges of an ANN trained on the synthetic dataset. ★ denotes the accuracy and bias after training but before pruning. The legend indicates the pruning strategy used: pruning based on bias-to-accuracy flow ratio (left) or accuracy-to-bias flow ratio (right); pruning nodes or edges; and the number of nodes or edges pruned. Pruning based on the bias-to-accuracy ratios causes bias to fall faster than accuracy, while pruning based on accuracy-to-bias ratios causes bias to *increase* while accuracy falls.

For simplicity, the neural network was taken to have just one hidden layer with three neurons, with leaky ReLU activations. The output layer is a one-hot encoding of the binary $\hat{Y}$ and cross-entropy loss was used for training. The data used for training the neural network was completely separate from the data used for estimating information flows on the trained network (we use a 50% split of the initial data for each component of the analysis). We used a kernel SVM to fit classifiers for the estimation of information measures (as described in Section 3.1), and nested cross validation to fit SVM hyperparameters and estimate generalization accuracy. Finally, all analyses are repeated across 100 neural networks trained on the same data but with different random weight initializations. Further details are in Appendix C.

We first analyzed whether the extent of change in the bias (Δ_{bias}) or accuracy (Δ_{acc}) at the output upon pruning an edge was related to the magnitude of the respective information flow on that edge. To do this, we completely pruned each edge of the trained network, keeping all other edges intact, and examined the change in accuracy and bias at the output. The results of this analysis are shown in Figure 2. The plots show that pruning edges with larger magnitudes of weighted accuracy or bias flow tends to produce greater decreases in accuracy or bias at the output, respectively. We also see *significantly* larger weighted bias flows than accuracy flows, which might be reasonable considering the presence of two highly biased features. There is also a marked difference in the slopes of the black and grey lines, indicating that edges from the second layer are less likely to change accuracy or bias upon being pruned: this is discussed in the following subsection.

Next, we analyzed how bias and accuracy evolve upon "soft-pruning" the edges gradually, for different pruning strategies outlined in Section 3.2. The results of this analysis are presented in Figure 3, from which it is clear that, when pruning on the basis of bias-to-accuracy flow ratio, bias falls faster than accuracy initially. The tradeoff curves produced by different pruning methods and levels (Section 3.2) were similar, but interestingly, pruning on the basis of the accuracy-to-bias flow ratio causes accuracy to fall, while bias *increases*. This may suggest the presence of redundancy: when edges most responsible for accuracy are pruned, other edges (presumably from nodes that have larger bias) are able to compensate, but at the cost of increasing bias.

4.2 Modified Adult Dataset

We also performed the same analyses on the Adult dataset (also known as the "Census Income" dataset) from the UCI machine learning repository [41]. The Adult dataset, which comes from the 1994 US census, consists of a mix of numerical and categorical features for classifying people with annual incomes less than and greater than $50k. We perform our experiments on two ANNs: a small neural network, identical to the one used for the synthetic dataset, and a slightly larger network having five input features, two hidden layers with three neurons each, and a one-hot-encoded output layer. For the smaller network, we use three numerical features—`education-num`, `hours-per-week` and `age`—and take `sex` to be the protected attribute. For the larger network, we additionally use

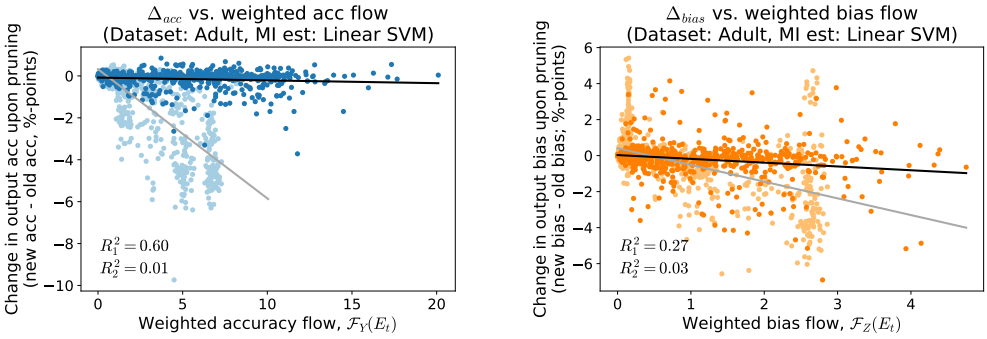

Figure 4: Plots analagous to those in Figure 2, for the Adult dataset trained on the smaller ANN.

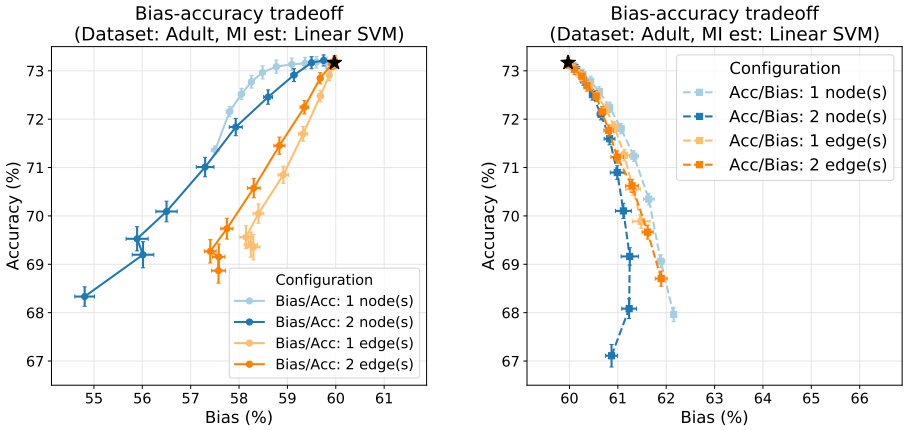

Figure 5: Tradeoff plots analogous to Figure 3, for the Adult dataset trained on the smaller ANN.

continuous-valued embeddings derived from two categorical features, `occupation` and `workclass`, to test that our observations are not limited to a specific topology. We also used only a subset of the records in order to equalize the number of individuals with high and low incomes and the number of male and female individuals. However, we introduced a bias in the true labels by skewing the dataset towards higher incomes among males and lower incomes among females (at a ratio of 2:1). In the figures, we simply use "Adult dataset" to refer to this modified version of the Adult dataset. Information estimates were performed using a linear SVM (refer to Section 3.1).

Results analogous to the synthetic dataset are shown in Figures 4 and 5 for the smaller ANN, and in Figure 6 for the larger ANN. These results reflect the trends seen in the synthetic dataset. Interestingly, Figures 2 and 4 indicate that the smaller ANN trained on the modified Adult dataset shows a stronger dependence between information flow and interventional effect than the one trained on the synthetic dataset. This may be a result of the fact that the synthetic data had two features that were *highly* biased, whereas the modified Adult dataset likely has much less bias in its input features. Figures 2 and 4 also show that interventions in the first layer (lighter points) tend to have a greater effect on output accuracy/bias than those in the second layer (darker points)—this is particularly pronounced in Figure 4 (left). We believe this is because pruning a single edge in the second layer will only affect *one* of the two output neurons: since the two outputs are a one-hot encoding of a binary random variable, they are fully redundant with respect to each other (i.e., they provide the same information). As a result, information estimates computed on the two output neurons are likely to be unaffected if pruning only affects the information content in *one* of the two: the classifiers used to estimate accuracy and bias will simply rely on the other neuron.

In Figures 5 and 6, edge pruning methods underperform node methods (e.g., compare 2-nodes with 6-edges in Figure 6, both of which prune 6 edges). This may suggest that the node-based method, which reduces the impact of an entire (derived) feature, may be a more robust intervention than a finer edge-based method. We also see that when pruning using accuracy-to-bias flow ratios (dashed lines), accuracy falls quickly and bias increases only modestly (in contrast to Figure 3 for the synthetic dataset). We believe this indicates that the modified Adult dataset is dominated more by

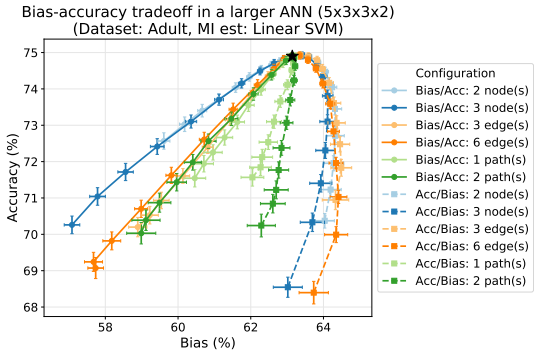

Figure 6: A tradeoff plot for the Adult dataset, trained on a slightly larger ANN, with five input features and two hidden layers with three neurons each. The green lines are for the path-based pruning method (described in Section 3.2).

accuracy flows while the synthetic dataset is dominated by bias flows (this is also substantiated by the magnitudes of information flow seen in Figures 2 and 4). Finally, the path-based pruning strategy (which makes more sense for deeper networks) is shown in Figure 6, and is similar to the edge-based methods in its tradeoff performance when using weighted bias-to-accuracy flow ratios. Interestingly however, when pruning based on weighted accuracy-to-bias flow ratios, there is no increase in bias, suggesting that path-based methods might have greater specificity to a particular message.

Further results, including a more extensive sampling of pruning strategies, as well as results from fast information flow estimates based on correlation, are provided in the supplementary material.

5 Discussion

Our results show that measures of information flow can indeed suggest targets for intervention in a trained artificial neural network, to change its behavior at the output. Furthermore, the bias-accuracy tradeoff curves indicate that we can use different messages to understand flows about different variables within the system, and edit the system to remove undesirable flows while preserving desirable ones. This has important implications for neuroscience, where such an information flow framework may prove useful for understanding how information is communicated between different parts of the brain. The result also shows that we can make use of such knowledge to then intervene and change flows in ways we desire, e.g., to treat various brain disorders. In such contexts, the sheer number of potential interventions, the amount of data required for each, and the inability to undo changes, all necessitate an approach that does not iterate through interventions. In general, the field of Causal Inference shows that purely observational measures cannot always yield results for interventions. Even so, our work provides compelling evidence that, in practice, the M-information flow framework can provide insights on where to intervene in a neural circuit.

Our work is intended to be an initial exploration for understanding how we might make interventions by studying information flows. Future studies may examine other intervention strategies, datasets, and more sophisticated networks. Extending these results to a neuroscientific setting may also be difficult. Firstly, the computational system model (nodes and edges) [13] may be too simplistic for biological neurons where even single cell has sophisticated dynamics. Secondly, measurements in real neural systems are frequently noisy, corrupted by background brain activity as well as sensor noise, while we assume here that noiseless measurements can be made. Similarly, although technologies for making interventions on specific cells are rapidly advancing [42], it is still difficult to perform precise neuromodulation, especially without genetic manipulation of cells.

Potential negative societal impacts. Although our paper uses the context of fairness to understand whether information flows can suggest interventions, we have not tested this approach rigorously enough for it to be employed to induce fairness in real systems—indeed, this is not our intent. While we believe that information flow analyses may, in future, be used to guide the systematic editing of neural networks for purposes such as bias reduction, that would require a more thorough analysis of optimal information measures and pruning strategies. In the meantime, the techniques developed here might be misused in fairness settings, despite having no formal guarantees.

References

[1] S. Cooper, A. Robison, and M. S. Mazei-Robison, "Reward circuitry in addiction," *Neurotherapeutics*, vol. 14, no. 3, pp. 687–697, 2017.

[2] N. D. Volkow, M. Michaelides, and R. Baler, "The neuroscience of drug reward and addiction," *Physiological Reviews*, vol. 99, no. 4, pp. 2115–2140, 2019.

[3] F. J. Muñoz-Cuevas, J. Athilingam, D. Piscopo, and L. Wilbrecht, "Cocaine-induced structural plasticity in frontal cortex correlates with conditioned place preference," *Nature Neuroscience*, vol. 16, no. 10, pp. 1367–1369, 2013.

[4] K. C. Berridge and T. E. Robinson, "What is the role of dopamine in reward: hedonic impact, reward learning, or incentive salience?" *Brain Research Reviews*, vol. 28, no. 3, pp. 309–369, 1998.

[5] S. J. Russo and E. J. Nestler, "The brain reward circuitry in mood disorders," *Nature Reviews Neuroscience*, vol. 14, no. 9, pp. 609–625, 2013.

[6] G. D. Stuber, D. R. Sparta, A. M. Stamatakis, W. A. van Leeuwen, J. E. Hardjoprajitno, S. Cho, K. M. Tye, K. A. Kempadoo, F. Zhang, K. Deisseroth *et al.*, "Excitatory transmission from the amygdala to nucleus accumbens facilitates reward seeking," *Nature*, vol. 475, no. 7356, pp. 377–380, 2011.

[7] D. A. Formolo, J. M. Gaspar, H. M. Melo, T. Eichwald, R. J. Zepeda, A. Latini, M. S. Okun, and R. Walz, "Deep brain stimulation for obesity: a review and future directions," *Frontiers in Neuroscience*, vol. 13, p. 323, 2019.

[8] A. K. Robinson, P. Venkatesh, M. J. Boring, M. J. Tarr, P. Grover, and M. Behrmann, "Very high density eeg elucidates spatiotemporal aspects of early visual processing," *Scientific Reports*, vol. 7, no. 1, pp. 1–11, 2017.

[9] J. S. Phillips, A. S. Greenberg, J. A. Pyles, S. K. Pathak, M. Behrmann, W. Schneider, and M. J. Tarr, "Co-analysis of brain structure and function using fmri and diffusion-weighted imaging," *Journal of Visualized Experiments: JoVE*, no. 69, 2012.

[10] A. K. Fishell, T. M. Burns-Yocum, K. M. Bergonzi, A. T. Eggebrecht, and J. P. Culver, "Mapping brain function during naturalistic viewing using high-density diffuse optical tomography," *Scientific Reports*, vol. 9, no. 1, pp. 1–11, 2019.

[11] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning*. fairmlbook.org, 2019. [Online]. Available: http://www.fairmlbook.org

[12] T. W. House, "Big data: A report on algorithmic systems, opportunity, and civil rights," 2016. [Online]. Available: https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/2016_0504_data_discrimination.pdf

[13] P. Venkatesh, S. Dutta, and P. Grover, "Information flow in computational systems," *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5456–5491, September 2020. [Online]. Available: https://doi.org/10.1109/TIT.2020.2987806

[14] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Proceedings of the Conference on Fairness, Accountability and Transparency*, 2018, pp. 107–118.

[15] H. Zhao and G. J. Gordon, "Inherent tradeoffs in learning fair representation," *arXiv preprint arXiv:1906.08386*, 2019.

[16] I. Y. Chen, F. D. Johansson, and D. Sontag, "Why is my classifier discriminatory?" in *Advances in Neural Information Processing Systems*, 2018, pp. 3539–3550.

[17] S. Sabato and E. Yom-Tov, "Bounding the fairness and accuracy of classifiers from population statistics," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8316–8325.

[18] J. S. Kim, J. Chen, and A. Talwalkar, "Model-agnostic characterization of fairness trade-offs," *arXiv preprint arXiv:2004.03424*, 2020.

[19] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 60–69.

[20] A. Blum and K. Stangl, "Recovering from biased data: Can fairness constraints improve accuracy?" 2019.

[21] S. Dutta, D. Wei, H. Yueksel, P.-Y. Chen, S. Liu, and K. Varshney, "Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2803–2813.

[22] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[24] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 6970–6979. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/0060ef47b12160b9198302ebdb144dcf-Paper.pdf

[25] U. Bhatt, A. Weller, and J. M. F. Moura, "Evaluating and aggregating feature-based model explanations," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3016–3022, main track.

[26] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 598–617.

[27] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1885–1894.

[28] B. Kim, O. Koyejo, R. Khanna *et al.*, "Examples are not enough, learn to criticize! Criticism for interpretability." in *NIPS*, 2016, pp. 2280–2288.

[29] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018. [Online]. Available: https://distill.pub/2018/building-blocks

[30] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017. [Online]. Available: https://distill.pub/2017/feature-visualization

[31] S. Dutta, P. Venkatesh, P. Mardziel, A. Datta, and P. Grover, "An information-theoretic quantification of discrimination with exempt features," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3825–3833.

[32] H. Wang, B. Ustun, and F. Calmon, "Repairing without retraining: Avoiding disparate impact with counterfactual distributions," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6618–6627.

[33] H. Wang, H. Hsu, M. Diaz, and F. P. Calmon, "To split or not to split: The impact of disparate treatment in classification," *arXiv preprint arXiv:2002.04788*, 2020.

[34] A. Ghassami, S. Khodadadian, and N. Kiyavash, "Fairness in supervised learning: An information theoretic approach," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 176–180.

[35] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[36] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks." *ProPublica*, 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[37] A. G. Ferguson, "Policing predictive policing," *Washington University Law Review*, vol. 94, p. 1109, 2016.

[38] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 214–226.

[39] W. P. Bergsma, "Testing conditional independence for continuous random variables," *EURANDOM report*, no. 049, 2004.

[40] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[41] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[42] E. S. Boyden, "A history of optogenetics: the development of tools for controlling brain circuits with light," *F1000 biology reports*, vol. 3, 2011.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The contributions are concretely listed in the Introduction.
   (b) Did you describe the limitations of your work? [Yes] See the Discussion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Discussion section.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] These details are provided in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] These details are provided in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figures 4, 5 and 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] These details are provided in the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used the Adult dataset from the UCI machine learning repository, and cited it appropriately.
   (b) Did you mention the license of the assets? [N/A] The UCI machine learning repository does not appear to explicitly specify a license, but it has been widely used by the research community; we assume it is public domain.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code to generate our synthetic dataset has been included in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]