Flow-based Attribution in Graphical Models: A Recursive Shapley Approach

Raghav Singal [1] **George Michailidis** [1 2] **Hoiyi Ng** [1]

Abstract

We study the attribution problem in a graphical model, wherein the objective is to quantify how the effect of changes at the source nodes propagates through the graph. We develop a model-agnostic flow-based attribution method, called *recursive Shapley value* (RSV). RSV generalizes a number of existing node-based methods and uniquely satisfies a set of flow-based axioms. In addition to admitting a natural characterization for linear models and facilitating mediation analysis for non-linear models, RSV satisfies a mix of desirable properties discussed in the recent literature, including implementation invariance, sensitivity, monotonicity, and affine scale invariance.

1. Introduction

Quantifying effect propagation is a fundamental problem, related to various concepts on causality: direct / indirect effects (Pearl, 2001), responsibility (Chockler & Halpern, 2004), explanations (Halpern & Pearl, 2005), mediation analysis (MacKinnon et al., 2007), and causal influence (Janzing et al., 2013). A recent application of effect propagation is on interpretable ML / explainable AI (XAI) (Adadi & Berrada, 2018). For example, in a neural network, quantifying how the effect of changes at the input layer propagates through the network aids in explaining its behavior, thus turning a complex and opaque AI system into a "glass-box" (Sokol & Flach, 2018; Biecek, 2018; Turek, 2020). In addition to being desirable, model interpretability is now enforced as a "right to explanation" (Selbst & Powles, 2018), highlighting the need to understand effect propagation.

In this work, we consider a graphical model where the underlying graph is a DAG with arbitrary relations between the variables ("model agnostic"). This is an adequately general framework capturing a wide array of causal and AI systems (e.g., non-linear structural equations and neural networks).

[1]Amazon [2]University of Florida. Correspondence to: RS <rs3566@columbia.edu>, GM <gmichail@ufl.edu>, HN <nghoiyi@amazon.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

For instance, as we illustrate in §6, our framework can be used to understand whether sensitive attributes exert an "unfair" influence on the output (e.g. school admission or credit approval). The key research question addressed is: "Given a change in the "source" nodes, how does the effect (change in the output) flow through the graph?"

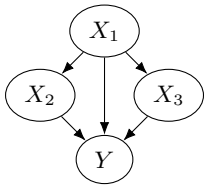

Figure 1. Example of a linear graphical model. X_1 is set exogenously, $X_2 = 2X_1$, $X_3 = 3X_1$, and $Y = 5X_1 + X_2 + X_3$.

As an example, consider the deterministic graphical model in Figure 1, specified by linear structural equations. Suppose the value at source node (without parents) X_1 changes from 0 to 1. As a result, the output Y changes from 0 to 10. How does the effect (10) of this change at X_1 flow through the graph? For this linear model, a natural answer is to use the edge weights: path $X_1 \to X_2 \to Y$ propagates 20% of the effect, $X_1 \to Y$ propagates 50%, and $X_1 \to X_3 \to Y$ propagates 30%. It is of interest to address the following questions regarding how changes at the "source" nodes flow through the graph: (a) what if the model is *non-linear* and (b) how the *graph structure* impacts the propagation.

The main contribution is to develop a flow-based attribution method that quantifies effect propagation, coined *recursive Shapley value* (RSV). RSV operates on a top-down principle by first attributing to the "source" nodes and then flowing it down the DAG, as illustrated in Figure 2. RSV generalizes a number of existing methods. Further, we establish RSV's uniqueness to a set of flow-based axioms, characterize it under a linear model, and illustrate how it facilitates mediation analysis in non-linear models. Finally, we demonstrate its adherence to various desirable properties discussed in the recent literature, including implementation invariance, sensitivity, monotonicity, and affine scale invariance.

Outline. The attribution problem is stated in §2, limitations of existing approaches[1] in §3, and the proposed approach in

[1]We focus on the XAI literature and plan to explore connections with the causality literature in a future work.

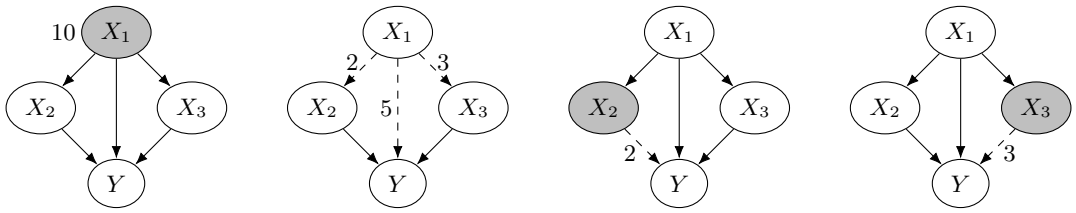

Figure 2. Illustration of RSV on the model from Figure 1. First, RSV attributes all the value of 10 (i.e., change in Y) to node 1. Second, RSV splits node 1's value (10) to its outgoing edges by evaluating their "contributions" via counterfactual questions of the following form: how much attribution would node 1 have received if edge $(1, 2)$ had not propagated the change at node 1? Third, RSV flows down the value received by node 2 to its outgoing edge by evaluating the attribution node 2 would have received if edge $(2, Y)$ had not propagated the change at node 2. Fourth, RSV repeats the procedure at node 3.

§4, followed by its properties and an application in §5 and §6, respectively. Concluding remarks are drawn in §7.

2. Problem Definition

We start by defining the underlying graphical model (§2.1) followed by the attribution problem (§2.2).

2.1. Graphical Model Setting

Consider a graph $\mathsf{G} = (\mathsf{N}^+, \mathsf{E})$ with node set $\mathsf{N}^+ := \{1, \ldots, n, n+1\}$ and edge set E. There may be multiple *source nodes* (nodes without "parents") and we denote their collection by N_0. The output of interest (e.g., model prediction) is captured by the *sink node* (node without "children") that corresponds to node $n+1$. For ease of exposition, the focus is on a *single sink* node, but generalizations are straightforward. Define $\mathsf{N} := \mathsf{N}^+ \setminus \{n+1\} = \{1, \ldots, n\}$. We assume G to be *directed acyclic* (DAG). Node i is a *parent* of node j (and node j is a *child* of node i), if there exists an edge (i, j) in E. Denote by P_j the set of parents of node j, i.e., $\mathsf{P}_j := \{i \in \mathsf{N} : (i, j) \in \mathsf{E}\}\ \forall j \in \mathsf{N}^+$. By definition, source nodes do not have any parents, i.e., $\mathsf{P}_j = \emptyset$ for all $j \in \mathsf{N}_0$. Denote by C_i the set of children of node i, i.e., $\mathsf{C}_i := \{j \in \mathsf{N}^+ : (i, j) \in \mathsf{E}\}\ \forall i \in \mathsf{N}^+$. By definition, the sink node does not have any children, i.e., $\mathsf{C}_{n+1} = \emptyset$. Node $i \in \mathsf{N}$ corresponds to a (possibly multi-dimensional) variable X_i and the sink node $n+1$ to variable $Y \in \mathbb{R}$. Occasionally, we will use X_{n+1} to denote Y. Define $\mathbf{X} := (X_1, \ldots, X_n)$ and $\mathbf{X}_N := (X_i)_{i \in N}$ for all $N \subseteq \mathsf{N}$. For ease of exposition, we assume each node to be a deterministic function of its parents: $X_i = f_i(\mathbf{X}_{\mathsf{P}_i})$ for all $i \in \mathsf{N}^+ \setminus \mathsf{N}_0$. Extension to stochastic functions is possible via structural equations with errors (Pearl, 2009), as illustrated with an example in §6 (Example 3). The values of the source nodes N_0 are set *exogenously and independently* of each other. We denote the output as a function of all the variables in the DAG by $Y = f(\mathbf{X})$, which equals $f_{n+1}(\mathbf{X}_{\mathsf{P}_{n+1}})$. Given the deterministic setup along with the DAG structure, the source variables $\mathbf{X}_{\mathsf{N}_0}$ are sufficient to determine the output Y. The graphical model is denoted by $\mathsf{M} := (\mathsf{G}, \mathsf{F})$, where $\mathsf{F} := [f_i(\cdot)]_{i \in \mathsf{N}^+ \setminus \mathsf{N}_0}$.

Remark 1. *The relationships in* F *are not necessarily causal; in case they are, then the posited graphical model is also causal. Hence, the proposed general framework captures causal graphs as a special case, but is not restricted to them, thus reflecting a wide array of ML / AI systems.*

2.2. The Attribution Problem

Next, we state the attribution problem. We are interested in two values of the input variables $\mathbf{X}$: $\mathbf{X}^{(1)} := (X_1^{(1)}, \ldots, X_n^{(1)})$ (*background*) and $\mathbf{X}^{(2)} := (X_1^{(2)}, \ldots, X_n^{(2)})$ (*foreground*). Both $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ satisfy the equations in F. Since the system is deterministic, changes at source nodes dictate changes in the entire graph. The background $\mathbf{X}^{(1)}$ defines the baseline output $Y^{(1)} := f(\mathbf{X}^{(1)})$, and analogously $Y^{(2)} := f(\mathbf{X}^{(2)})$. Hence, the change in the output Y equals $Y^{(2)} - Y^{(1)}$. Since source nodes N_0 capture all the exogenous sources of variation, the change in Y is *solely driven* by the magnitude of the change at N_0 and its propagation through the DAG.

The question of interest is how to attribute the change in Y to the changes at N_0 and their propagation through edges. In a DAG with a single node, i.e., $n = 1$, attribution is trivial since the single node is solely responsible for the change and hence, receives 100% attribution, which flows through the only edge. However, in a DAG with $n > 1$ nodes, there can be non-trivial interactions. How does one decouple such interactions? Is it possible to leverage the structure of the graphical model and compute a flow-based attribution?

To address these questions, some additional notation is introduced: $[\pi_i]_{i \in \mathsf{N}}$ denotes attribution to nodes and $[\pi_{ij}]_{(i,j) \in \mathsf{E}}$ attribution to edges (*attribution flow*). Their scale is expressed in terms of $Y^{(2)} - Y^{(1)}$. In our deterministic setup, since the source nodes capture all the exogenous source of variation, a desirable property is $\sum_{i \in \mathsf{N}_0} \pi_i = Y^{(2)} - Y^{(1)}$, which we call *source efficiency*. Further, we require that flow be *conserved* at each node, i.e., flow in equals flow out.

Definition 1 (Flow conservation)**.** *At internal node* $j \in \mathsf{N} \setminus \mathsf{N}_0$, $\sum_{i \in \mathsf{P}_j} \pi_{ij} = \sum_{k \in \mathsf{C}_j} \pi_{jk}$. *For source nodes,* $\sum_{i \in \mathsf{N}_0} \sum_{j \in \mathsf{C}_i} \pi_{ij} = Y^{(2)} - Y^{(1)}$. *At sink node,*

$\sum_{i \in \mathsf{P}_{n+1}} \pi_{i,n+1} = Y^{(2)} - Y^{(1)}$.

Wang et al. (2021) advocate for a similar property: "cut efficiency". It is easy to verify that flow conservation holds iff cut efficiency holds. Moreover, flow conservation generalizes the conservation properties in Bach et al. (2015) as we do not assume G to be layered. Though flow conservation is desirable, it is not sufficient to output a unique flow.

Given our focus on attribution, we assume M to be given. Graph estimation / causal discovery (Peters et al., 2017; Glymour et al., 2019) is an active area of research and we emphasize that even with a known graphical model / causal graph, attribution is challenging. As noted in Remark 1, M need not be causal and hence, we can "incorporate known causal relationships without the prohibitive requirement of a full causal graph", as advocated in Frye et al. (2020b).

3. Existing Approaches and Their Limitations

Next, we use the proposed framework to discuss existing approaches in the XAI literature, including node-based (§3.1) and edge- / flow-based (§3.2).

3.1. Node-based Approaches

Node-based approaches typically use the Shapley value (SV) (Shapley, 1953)[2] for attribution to the nodes in a graph. They consider the nodes as the players in the underlying game and define an appropriate coalition and a characteristic function. The attribution received by a node is its SV in the constructed game. Depending on how the coalition and the characteristic function are defined, the attribution can be different. Independent SV (ISV) (Strumbelj & Kononenko, 2010; Sun & Sundararajan, 2011; Sundararajan et al., 2020; Janzing et al., 2020; Sundararajan & Najmi, 2020) attributes all the value to the parents of the output node, whereas conditional SV (CSV) (Štrumbelj & Kononenko, 2014; Datta et al., 2016; Lundberg & Lee, 2017; Aas et al., 2019; Frye et al., 2020a) attributes to all the nodes. Both violate source efficiency. Asymmetric SV (ASV) (Frye et al., 2020b) satisfies source efficiency, but does not inform how the effect flows through the DAG. We formally define these approaches using our graphical model language in Appendix A and focus here on illustrating their key limitations via a simple example (Example 1).

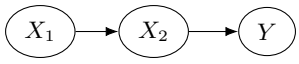

Figure 3. The graphical model for Example 1. The source variable X_1 is set exogenously, $X_2 = X_1$, and $Y = X_2$. We consider the background value of $X_1^{(1)} = 0$ and the foreground value of $X_1^{(2)} = 1$. Hence, $X_2^{(1)} = Y^{(1)} = 0$ and $X_2^{(2)} = Y^{(2)} = 1$.

[2]See Chapter 8 (Peleg & Sudhölter, 2007) for a primer on SV.

Example 1 (Chain)**.** *Consider the model in Figure 3. ISV attributes all the value to node 2:* $(\pi_1^{ISV}, \pi_2^{ISV}) = (0, 1)$*. CSV splits the value:* $(\pi_1^{CSV}, \pi_2^{CSV}) = (1/2, 1/2)$*. Thus, ISV and CSV violate source efficiency. ASV attributes all the value to node 1:* $(\pi_1^{ASV}, \pi_2^{ASV}) = (1, 0)$*. Though ASV obeys source efficiency, it does not identify how the effect flows through the DAG. (Detailed computations are given in Appendix A.)*

We note that such node-based approaches are better suited for regression settings, wherein all the nodes in the DAG except the sink node are source nodes. A flow-based view is most useful in "hierarchical" graphs. In fact, as we discuss in §5, our flow-based approach recovers such node-based approaches for regression settings.

3.2. Edge-based / Flow-based Approaches

Approaches such as LRP (Bach et al., 2015; Binder et al., 2016), DeepLIFT (Shrikumar et al., 2017), conductance (Dhamdhere et al., 2018), and internal influence (Leino et al., 2018) are flow-based but *model-specific*, since they impose assumptions on the model M (e.g., layered structure in G or continuity / linearity in F). As Frye et al. (2020b) note, "model-specific approaches are bespoke in nature and do not solve the problem of explainability in general". The approach we propose in §4 is *model-agnostic* (Ribeiro et al., 2016), wherein we allow for an arbitrary M (as long as G is a DAG). Further, backpropagation-based approaches such as LRP (Binder et al., 2016), DeepLIFT (Shrikumar et al., 2017), and DeepSHAP (Lundberg & Lee, 2017) do not satisfy *implementation invariance* (see §8.2 of Dhamdhere et al. (2018)), whereas our approach does (§5).

The closest idea to our approach is Shapley Flow (SF) (Wang et al., 2021). At a high-level (details in Appendix B), in contrast to our *edge-based* approach, SF is *path-based* as it considers each source-to-sink path as a player. Furthermore, SF modifies the original definition of SV by only considering orderings that are consistent with a depth first search (discussed in Appendix B). Given this modification, it is unclear what connection SF exhibits with SV (if any). To the best of our knowledge, there is no underlying "game" for which SF is the SV of [3]. On the other hand, our edge-based proposal comes out naturally from a well-defined sequence of games, without any ad hoc modifications.

There exists a growing body of literature on interpretable ML in addition to the approaches discussed above. We refer the reader to Molnar (2020) and Molnar et al. (2020) for a

[3]Given a cut, Wang et al. (2021) define a game with the cut-set being the players. A coalition is defined to be a "partial history from $t = 0$ to any $t \in [0 \dots T]$", where a "*history* is a list of edges detailing the event from $t = 0$ (values being $X^{(1)}$) to $t = T$ (values being $X^{(2)}$)". The characteristic function is "the evaluation of f following the coalition". We are unable to map this to a formal game and verify its connection to the path-based formula of SF.

comprehensive overview. It suffices to note that this stream of literature does not focus on flow-based approaches and hence, is not directly related to our goal.

4. The Recursive Shapley Value Approach

We start by defining necessary quantities, followed by intuition (§4.1) and then a formal definition (§4.2). Finally, we establish related axioms (§4.3).

For convenience, we insert a *super-source* (node 0) to G with edges directed to source nodes: $(0, j)$ for $j \in \mathsf{N}_0$. We re-define $\mathsf{E} \leftarrow \mathsf{E} \cup \{(0, j) : j \in \mathsf{N}_0\}$ to include the super-source edges, and $\mathsf{N} \leftarrow \mathsf{N} \cup \{0\}$ and $\mathsf{N}^+ \leftarrow \mathsf{N}^+ \cup \{0\}$ to include the super-source node. We still use N_0 to denote the original set of source nodes (and not the super-source). $\mathsf{E}_i := \{(i, j) : j \in \mathsf{C}_i\}$ denotes outgoing edges of node $i \in \mathsf{N}$ and hence, $(\mathsf{E}_0, \ldots, \mathsf{E}_n)$ is a partition of E. Our flow-based approach outputs edge attributions $[\pi_{ij}]_{(i,j)\in\mathsf{E}}$ and we define node attributions as the sum of all incoming flows, i.e., $\pi_j := \sum_{i\in\mathsf{P}_j} \pi_{ij}$ for all $j \in \mathsf{N} \setminus \{0\}$.

The notion of an edge being *active* or *inactive*, defined next, proves helpful in the sequel. In the posited framework, an edge communicates information. Thus, motivated by Pearl (2001), an edge $(i, j) \in \mathsf{E}$ being active means it transmits the "updated" value X_i from node i to j, whereas an inactive edge $(i, j) \in \mathsf{E}$ is unable to communicate the "update" and thus, node j receives the background value $X_i^{(1)}$, as illustrated in Figure 4.

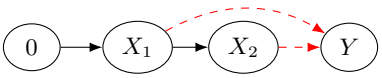

Figure 4. Illustration of active and inactive edges. Set of edges $\mathsf{E} = \{(0, 1), (1, 2), (1, 3), (2, 3)\}$. Node "3" corresponds to Y. Edges $(1, 3)$ and $(2, 3)$ are inactive (red dashed lines). Node 1 is set to the foreground $X_1^{(2)}$ since edge $(0, 1)$ is active. Node 2 receives $X_1^{(2)}$ (since edge $(1, 2)$ is active *and* X_1 is set to $X_1^{(2)}$) but node 3 ("Y") receives $X_1^{(1)}$ (since edge is $(1, 3)$ inactive) and $X_2^{(1)}$ (since edge $(2, 3)$ is inactive). Hence, $X_1 = X_1^{(2)}$, $X_2 = f_2(X_1^{(2)})$, and $Y = f_3(X_1^{(1)}, X_2^{(1)})$. If edge $(0, 1)$ were inactive, then $X_1 = X_1^{(1)}$, $X_2 = f_2(X_1^{(1)})$, and $Y = f_3(X_1^{(1)}, X_2^{(1)})$.

In general, consider any subset of active edges $E \subseteq \mathsf{E}$. Then, each source node $i \in \mathsf{N}_0$ is set as follows:

$$X_i(E) := \begin{cases} X_i^{(1)} & \text{if } (0, i) \notin E \\ X_i^{(2)} & \text{if } (0, i) \in E. \end{cases} \tag{1}$$

Each non-source node $j \in \mathsf{N}^+ \setminus \{\mathsf{N}_0 \cup \{0\}\}$ obeys $X_j(E) = f_j((X_{ij}(E))_{i\in\mathsf{P}_j})$, where for all $i \in \mathsf{P}_j$,

$$X_{ij}(E) := \begin{cases} X_i^{(1)} & \text{if } (i, j) \notin E \\ X_i(E) & \text{if } (i, j) \in E. \end{cases} \tag{2}$$

Notation X_{ij} is new. $X_{n+1}(E)$ denotes the output $Y(E)$. Hence, the notation $Y(E)$ is well-defined for all $E \subseteq \mathsf{E}$. With this notation at hand, we proceed to providing intuition.

4.1. Intuition Guiding RSV

The RSV approach first attributes to the source nodes. Then, it flows down the DAG the attributions by computing the "contribution" of each outgoing edge ("top-down"). As a concrete example, consider the graph in Figure 5 and an arbitrary F. Recall that $\mathbf{X}$ changes from $\mathbf{X}^{(1)}$ to $\mathbf{X}^{(2)}$, resulting in Y changing from $Y^{(1)}$ to $Y^{(2)}$. In the deterministic setup, source nodes $\mathsf{N}_0 = \{1, 2\}$ capture all the exogenous source of variation and hence, the change in Y is solely driven by how the changes at (X_1, X_2) propagate through the DAG. Thus, in agreement with distal ASV (Frye et al., 2020b), we first attribute to the source nodes. To do so, we consider the following game at node 0. The set of players is $\mathsf{E}_0 = \{(0, 1), (0, 2)\}$. Given a coalition $E_0 \subseteq \mathsf{E}_0$, the characteristic function is defined as $v_0(E_0) := Y(E_0, \mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4)$, where the notation $Y(\cdot)$ follows (1) and (2). All downstream edges $(\mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4)$ are active. Then, edges $(0, 1)$ and $(0, 2)$ receive attributions π_{01} and π_{02} equal to the SVs of this game (Figure 6). Source efficiency is satisfied, since $\pi_1 + \pi_2 = \pi_{01} + \pi_{02} = v_0(\mathsf{E}_0) - v_0(\emptyset) = Y^{(2)} - Y^{(1)}$. The second equality follows SV efficiency and the third equality is by construction. For clarity, we use $\pi_{01}(\mathsf{E})$, $\pi_{02}(\mathsf{E})$, $\pi_1(\mathsf{E})$, and $\pi_2(\mathsf{E})$ to capture the dependence on $\mathsf{E} = (\mathsf{E}_0, \mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4)$.

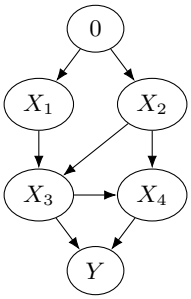

Figure 5. Graph for illustrating RSV's intuition.

Next, we proceed down to nodes 1 and 2. Attribution through node 1 is trivial since it only has one outgoing edge. Hence, flow conservation implies $\pi_{13} = \pi_1(\mathsf{E})$. However, understanding how $\pi_2(\mathsf{E})$ flows through node 2 is non-trivial, since node 2 has two outgoing edges: $\mathsf{E}_2 = \{(2, 3), (2, 4)\}$. We wish to understand how much of the attribution received by node 2 is due to the "presence" of each of these two edges. Recall that in the game at node 0, we assumed the downstream edges to be active. Hence, *if* both $(2, 3)$ and $(2, 4)$ are active, *then* node 2 receives an attribution of $\pi_2(\mathsf{E}_0, \mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4)$. To determine π_{23} and π_{24}, we posit the following counterfactual questions. How much attribution would node 2 have received if both $(2, 3)$

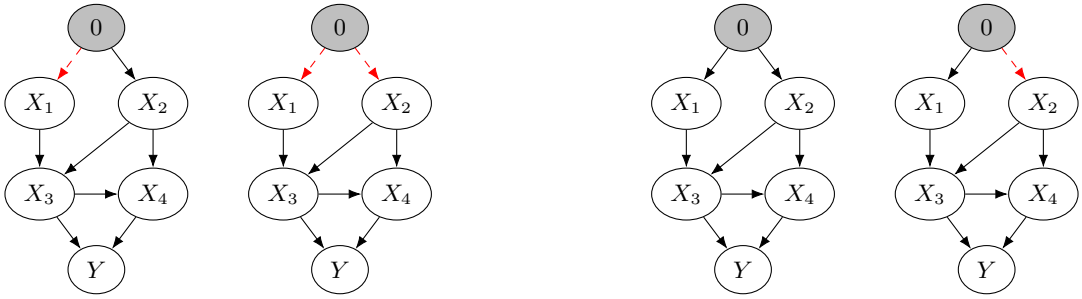

Figure 6. Visualizing the computation of π_{02}. Edge $(0, 2)$ receives an attribution equal to its "value-add", which is evaluated via the following counterfactual question: how much would have been the effect (change at node Y) had edge $(0, 2)$ been inactive? Note that there are two possibilities in such a counterfactual question: (a) edge $(0, 1)$ is inactive and (b) edge $(0, 1)$ is active. The difference between the first two subfigures corresponds to possibility (a), i.e., value-add of edge $(0, 2)$ when edge $(0, 1)$ is inactive. Similarly, the difference between the last two subfigures corresponds to possibility (b), i.e., value-add of edge $(0, 2)$ when edge $(0, 1)$ is active. The attribution received by edge $(0, 2)$ is the weighted average of these two value-adds, where the weights come from the classical SV (1/2 for this instance). Note that all downstream edges $(\mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4)$ are active. The computation of π_{01} can be visualized similarly.

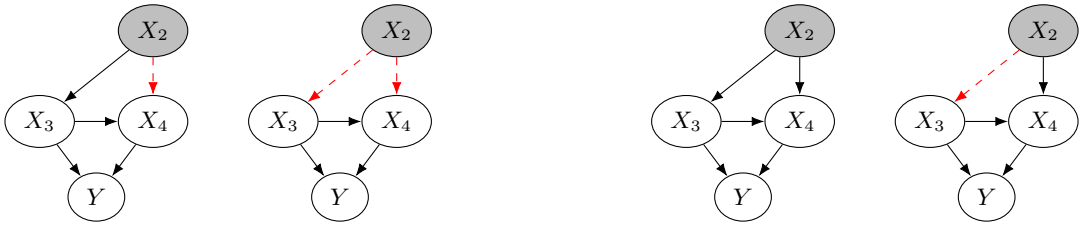

Figure 7. Visualizing the computation of π_{23}. Edge $(2, 3)$ receives an attribution equal to its "value-add", which is evaluated via the following counterfactual question: how much attribution would node 2 have received had edge $(2, 3)$ been inactive? Note that there are two possibilities in such a counterfactual question: (a) edge $(2, 4)$ is inactive and (b) edge $(2, 4)$ is active. The difference between the first two subfigures corresponds to possibility (a), i.e., value-add of edge $(2, 3)$ when edge $(2, 4)$ is inactive. Similarly, the difference between the last two subfigures corresponds to possibility (b), i.e., value-add of edge $(2, 3)$ when edge $(2, 4)$ is active. The attribution received by edge $(2, 3)$ is the weighted average of these two value-adds, where the weights come from the classical SV (1/2 for this instance). However, one still needs to compute the attribution node 2 receives in each of these four subfigures. In Figure 6, we did this computation for the third subfigure here, i.e., assuming both $(2, 3)$ and $(2, 4)$ to be active. Figure 8 below shows the computation corresponding to the first subfigure, i.e., $(2, 3)$ active but $(2, 4)$ inactive. Computations for subfigures 2 and 4 are similar and not shown for brevity. Further, the computation of π_{24} can be visualized similarly.

and $(2, 4)$ were inactive? What if one of them was inactive? Observe that in the game at node 0, if both $(2, 3)$ and $(2, 4)$ were inactive, then the corresponding characteristic function would have been $Y(E_0, \mathsf{E}_1, \emptyset, \mathsf{E}_3, \mathsf{E}_4)\ \forall E_0 \subseteq \mathsf{E}_0$ and edge $(0, 2)$ (and hence, node 2) would have received zero attribution, i.e., $\pi_2(\mathsf{E}_0, \mathsf{E}_1, \emptyset, \mathsf{E}_3, \mathsf{E}_4) = 0$. Hence, edges $(2, 3)$ and $(2, 4)$ are "responsible" for node 2 receiving $\pi_2(\mathsf{E})$ and it seems logical to conserve flow: $\pi_2(\mathsf{E}) = \pi_{23} + \pi_{24}$. However, this is not sufficient to uniquely determine (π_{23}, π_{24}).

An easy case corresponds to a setting, wherein an edge being active or inactive does not change the attribution received by node 2. Then, it seems natural to assign zero attribution to that edge ("nullity"). Another easy case is if both edges "exert same impact" on the attribution received by node 2. For example, if node 2 received an attribution of $\pi_2(\mathsf{E})/2$ if *either* of the edges were active, then splitting $\pi_2(\mathsf{E})$ equally between the two edges seems appropriate ("symmetry").

To operationalize this intuition, we consider the following game at node 2, with the set of players being E_2. Given a coalition $E_2 \subseteq \mathsf{E}_2$, the characteristic function is defined as the attribution received by node 2 from the game at node 0 (hence, "recursive"), i.e., $v_2(E_2) := \pi_2(\mathsf{E}_0, \mathsf{E}_1, E_2, \mathsf{E}_3, \mathsf{E}_4)$. Then, edges $(2, 3)$ and $(2, 4)$ receive attributions π_{23} and π_{24} equal to the SVs of this game (Figure 7). Flow conservation holds since $\pi_{23} + \pi_{24} = v_2(\mathsf{E}_2) - v_2(\emptyset) = \pi_2(\mathsf{E})$. We then move down to nodes 3 and 4, which use a similar logic. The set of players at the node 3 game is E_3 and given coalition $E_3 \subseteq \mathsf{E}_3$, the characteristic function is defined as the attribution received by node 3 from the upstream games, i.e., $v_3(E_3) := \pi_3(\mathsf{E}_0, \mathsf{E}_1, \mathsf{E}_2, E_3, \mathsf{E}_4)$.

Remark 2. *Our "top-down" philosophy is fundamentally different from backpropagation-based ("bottom-up") approaches that rely on an ad hoc "chain rule" and a "linear approximation" to conserve flow (see (15) and (16) in Lundberg & Lee (2017)). As shown in §4.3, RSV naturally satisfies flow conservation and three additional flow-based*

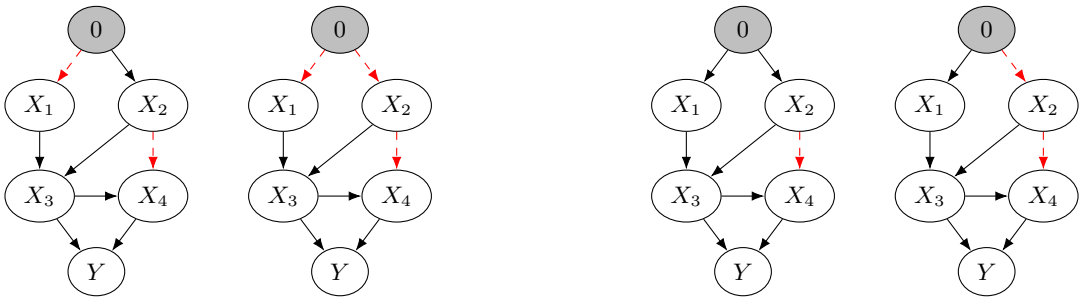

Figure 8. Visualizing the computation of π_{02} (attribution node 2 receives) when edge $(2,4)$ is inactive. The four subfigures here are identical to those in Figure 6 but with one difference: edge $(2,4)$ in inactive. The computational procedure is the same as discussed in the caption of Figure 6.

properties. Further, as we establish in Proposition 1, it obeys implementation invariance, which is a known issue with bottom-up approaches (Dhamdhere et al., 2018).

4.2. Recursive Shapley Value

RSV involves multiple recursions. A recursion is initialized via a game at each node $j \in \mathsf{N} \setminus \{0\}$, with set of players E_j. Given a coalition $E_j \subseteq \mathsf{E}_j$, the characteristic function is defined as the attribution received by node j from the upstream games assuming all other edges to be active, i.e., $v_j(E_j) := \pi_j(\mathsf{E}_0, \dots, E_j, \dots, \mathsf{E}_n) = \sum_{i \in \mathsf{P}_j} \pi_{ij}(\mathsf{E}_0, \dots, E_j, \dots, \mathsf{E}_n)$. Then, the attribution received by edge $(j,k) \in \mathsf{E}_j$ equals the corresponding SV of this game: $\pi_{jk}(\mathsf{E}) = \sum_{E_j \subseteq \mathsf{E}_j \setminus \{(j,k)\}} w_{\mathsf{E}_j}(E_j) \times \{v_j(E_j \cup \{(j,k)\}) - v_j(E_j)\}$, where $w_{\mathsf{E}_j}(E_j) := |E_j|!(|\mathsf{E}_j| - |E_j| - 1)!/|\mathsf{E}_j|!$ is the SV weight. Observe that $\pi_{jk}(\mathsf{E}_0, \dots, \mathsf{E}_n)$ depends on $\pi_{ij}(\mathsf{E}_0, \dots, E_j, \dots, \mathsf{E}_n)$ for $E_j \subseteq \mathsf{E}_j$, where i is a parent of j; the latter defines a recursion. When computing $\pi_{ij}(\mathsf{E}_0, \dots, E_j, \dots, \mathsf{E}_n)$ at any upstream node $i \in \mathsf{P}_j$, the characteristic function of the corresponding game at node i would be $\pi_i(\mathsf{E}_0, \dots, E_i, \dots, E_j, \dots, \mathsf{E}_n)$ for $E_i \subseteq \mathsf{E}_i$, as opposed to $\pi_i(\mathsf{E}_0, \dots, E_i, \dots, \mathsf{E}_j, \dots, \mathsf{E}_n)$, i.e., we do not assume all other edges to be active *within* a recursion, but only at its *initialization*. The pseudocode in Algorithm 1 below clarifies this. We use the notation $v_j(E_j \mid E_{-j})$ to emphasize the difference when needed, where $E_{-j} := (E_0, \dots, E_{j-1}, E_{j+1}, \dots, E_n)$ and $E_\ell \subseteq \mathsf{E}_\ell \ \forall \ell \in \mathsf{N}$. Unless otherwise stated, $v_j(\cdot) = v_j(\cdot \mid \mathsf{E}_{-j}) \ \forall j \in \mathsf{N}$.

Every recursion breaks via a game at node 0, which is defined as follows. Consider an arbitrary collection of downstream edges $(E_1, \dots, E_n)$, where $E_\ell \subseteq \mathsf{E}_\ell$, $\forall \ell = 1, \dots, n$. We define this game conditioned on $(E_1, \dots, E_n)$, with set of players E_0. Given a coalition $E_0 \subseteq \mathsf{E}_0$, the characteristic function is defined as the output value given edges E_{-0} (recall (1) and (2)), i.e., $v_0(E_0 \mid E_{-0}) := Y(E_0, E_{-0})$. Then, the attribution received by edge $(0,k) \in \mathsf{E}_0$ equals the corresponding SV of this game: $\pi_{0k}(\mathsf{E}_0, E_{-0}) = \sum_{E_0 \subseteq \mathsf{E}_0 \setminus \{(0,k)\}} w_{\mathsf{E}_0}(E_0) \times \{v_0(E_0 \cup \{(0,k)\} \mid E_{-0}) - v_0(E_0 \mid E_{-0})\}$, which is non-recursive. The attribution received by super-source edges themselves correspond to the game at node 0 with all downstream edges being active: $(\mathsf{E}_1, \dots, \mathsf{E}_n)$.

Each recursion is well-defined due to the DAG and boils down to evaluating $Y(E)$ for $E \subseteq \mathsf{E}$. Algorithm 1 along with the sub-routine Algorithm 2 formalize the recursive procedure to compute RSV. Algorithm 1 outputs $\pi^{\text{RSV}}_{jk} = \pi_{jk}(\mathsf{E}) \ \forall (j,k) \in \mathsf{E}$. The inputs N and E to Algorithm 1 are assumed to contain the super-source node and edges.

Algorithm 1 $\texttt{RSV}(\mathsf{N}, \mathsf{E})$

for $(j,k) \in \mathsf{E}$
  $\pi^{\text{RSV}}_{jk} = \texttt{RSV}_{jk}(\mathsf{E}_0, \dots, \mathsf{E}_n)$
end for
return $[\pi^{\text{RSV}}_{jk}]_{(j,k) \in \mathsf{E}}$

In Example 1, $(\pi^{\text{RSV}}_{01}, \pi^{\text{RSV}}_{12}, \pi^{\text{RSV}}_{23}) = (1,1,1)$ (see Appendix C), where π^{RSV}_{01} is the flow received by node 1. Thus, RSV gives a complete view of effect propagation. In fact, for Example 1, flow conservation suffices to uniquely determine this flow. However, this is not true in general. Hence, it is of interest to investigate what properties *uniquely* determine an attribution flow, a task resolved next.

4.3. Flow-based Axioms

The first property is flow conservation, as introduced in §2. The second and third properties are flow symmetry and nullity, as alluded to in §4.1. Informally, symmetry requires "equivalent" outgoing edges to receive the same flow, whereas nullity requires a "redundant" edge to receive zero flow. The fourth property is flow linearity, presented after the formal definition of these properties.

Definition 2. *The flow-based axioms are as follows:*

1. *Flow conservation:* $\sum_{k \in \mathsf{C}_0} \pi_{0k} = Y^{(2)} - Y^{(1)}$ *and* $\sum_{i \in \mathsf{P}_j} \pi_{ij} = \sum_{k \in \mathsf{C}_j} \pi_{jk} \ \forall j \in \mathsf{N} \setminus \{0\}$.

Algorithm 2 $\texttt{RSV}_{jk}(\mathcal{E}_0, \ldots, \mathcal{E}_n)$

if $j > 0$ % recursion
  return $\sum_{E_j \subseteq \mathcal{E}_j \setminus \{(j,k)\}} w_{\mathcal{E}_j}(E_j) \times \sum_{i \in \mathsf{P}_j} \{\texttt{RSV}_{ij}(\mathcal{E}_0, \ldots, E_j \cup \{(j,k)\}, \ldots, \mathcal{E}_n) - \texttt{RSV}_{ij}(\mathcal{E}_0, \ldots, E_j, \ldots, \mathcal{E}_n)\}$
else % base case
  return $\sum_{E_0 \subseteq \mathcal{E}_0 \setminus \{(0,k)\}} w_{\mathcal{E}_0}(E_0) \times \{Y(E_0 \cup \{(0,k)\}, \mathcal{E}_1, \ldots, \mathcal{E}_n) - Y(E_0, \mathcal{E}_1, \ldots, \mathcal{E}_n)\}$
end if

2. *Flow symmetry: For node $j \in \mathsf{N}$, if $(j,k) \in \mathsf{E}_j$ and $(j,\ell) \in \mathsf{E}_j$ are such that $v_j(E_j \cup \{(j,k)\}) = v_j(E_j \cup \{(j,\ell)\})\ \forall E_j \subseteq \mathsf{E}_j \setminus \{(j,k),(j,\ell)\}$, then $\pi_{jk} = \pi_{j\ell}$.*
3. *Flow nullity: For node $j \in \mathsf{N}$, if $v_j(E_j \cup \{(j,k)\}) = v_j(E_j)\ \forall E_j \subseteq \mathsf{E}_j \setminus \{(j,k)\}$, then $\pi_{jk} = 0$.*
4. *Flow linearity: For node $j \in \mathsf{N}$, consider characteristic functions $v_j(\cdot)$ and $u_j(\cdot)$. Linearity requires $\pi_{jk}(v_j+u_j) = \pi_{jk}(v_j)+\pi_{jk}(u_j)\ \forall (j,k) \in \mathsf{E}_j$. (Notation $\pi_{jk}(v_j)$ captures the dependence of π_{jk} on $v_j(\cdot)$.)*

Note that flow conservation is stated differently than in Definition 1, but the two are equivalent. To interpret linearity, observe that $[v_j(\cdot)]_{j \in \mathsf{N}}$, as defined in §4.2, ultimately corresponds to F since for all $j \in \mathsf{N}$, $v_j(\cdot)$ boils down to evaluating $v_0(\cdot)$, which depends on F. Think of $[u_j(\cdot)]_{j \in \mathsf{N}}$ corresponding to a different population of $\mathbf{X}$, such that the underlying equations are different (say F'). Linearity requires the attribution to be robust to mixing heterogeneous populations. That is, mixing the two populations first ($v_j + u_j$) and using the mixture population to compute attribution ($\pi_{jk}(v_j + u_j)$) should be the same as computing attributions on individual populations first ($\pi_{jk}(v_j)$ and $\pi_{jk}(u_j)$) and mixing them later ($\pi_{jk}(v_j) + \pi_{jk}(u_j)$).

To interpret symmetry in terms of the model primitive $Y(\cdot)$, it is easy to verify that for node $j \in \mathsf{N}$, if $(j,k) \in \mathsf{E}_j$ and $(j,\ell) \in \mathsf{E}_j$ obey $Y(E \cup \{(j,k)\}) = Y(E \cup \{(j,\ell)\})\ \forall E \subseteq \mathsf{E} \setminus \{(j,k),(j,\ell)\}$, then flow symmetry implies $\pi_{jk} = \pi_{j\ell}$. Similarly, if $Y(E \cup \{(j,k)\}) = Y(E)\ \forall E \subseteq \mathsf{E} \setminus \{(j,k)\}$, then flow nullity implies $\pi_{jk} = 0$. Having defined the axioms, we establish in Theorem 1 that RSV is their unique solution (proof given in Appendix D).

Theorem 1. $[\pi^{RSV}_{jk}]_{(j,k) \in \mathsf{E}}$ *is the unique solution to the flow-based axioms.*

Theorem 1 follows when the uniqueness of SV is applied to the game at each node. Thus, the novelty is primarily in constructing the overall flow-based recursive framework, which facilitates such axioms in the first place. Further, in the original SV setting (Shapley, 1953), uniqueness holds for a given game (defined via the set of players and the characteristic function). Thus, uniqueness of RSV holds given our choice of $n+1$ games (one game for each node in N). For the game at node 0, we defined the set of players as E_0 and the characteristic function as $Y(E_0, \mathsf{E}_1, \ldots, \mathsf{E}_n)\ \forall E_0 \subseteq \mathsf{E}_0$. For the game at node $j \in \mathsf{N} \setminus \{0\}$, we defined the set of players as E_j and the characteristic function as the attribution received by node j from the upstream games, i.e., $\sum_{i \in \mathsf{P}_j} \pi_{ij}(\mathsf{E}_0, \ldots, E_j, \ldots, \mathsf{E}_n)\ \forall E_j \subseteq \mathsf{E}_j$. Such a setup seems apt as it naturally results in flow conservation. Further, as we show in §5, RSV satisfies additional properties that have been deemed desirable in the ML / AI literature. To conclude this section, we illustrate the flow-based axioms via a DAG equipped with linear equations F (Example 2).

Example 2 (Linear model)**.** *Consider the model in Figure 9. RSV attributes $a_{12}a_{24}$ to edges $(1,2)$ and $(2,4)$, $a_{13}a_{34}$ to edges $(1,3)$ and $(3,4)$, and a_{14} to edge $(1,4)$.*

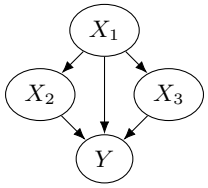

Figure 9. The model for Example 2. The source variable X_1 is set exogenously, $X_2 = a_{12}X_1$, $X_3 = a_{13}X_1$, and $Y = a_{14}X_1 + a_{24}X_2 + a_{34}X_3$. We consider the background $X_1^{(1)} = 0$ and the foreground $X_1^{(2)} = 1$. Hence, $(X_2^{(1)}, X_3^{(1)}, Y^{(1)}) = (0,0,0)$ and $(X_2^{(2)}, X_3^{(2)}, Y^{(2)}) = (a_{12}, a_{13}, a_{14} + a_{12}a_{24} + a_{13}a_{34})$. (Node 0 is not included for brevity.)

Flow conservation is straightforward. To understand symmetry, suppose $a_{12} = a_{13}$ and $a_{24} = a_{34}$, so that the node 1 change communicated to the output node through edge $(1,2)$ is identical to that through $(1,3)$. RSV attributes the same flow to both these edges. For nullity, suppose $a_{jk} = 0$ for some $(j,k) \in \mathsf{E}$. RSV attributes zero flow to such an edge. Finally, for linearity, consider a different model with the same G, but different coefficients a'_{jk} for all $(j,k) \in \mathsf{E}$. RSV in the "mixture" model ($\tilde{a} = (a + a')/2$) equals the RSV under the individual models "averaged" together. For example, mixture model attributes $(a_{14} + a'_{14})/2$ to edge $(1,4)$ and individual models attribute a_{14} and a'_{14}.

Example 2 also illustrates how RSV distinguishes *direct / indirect effects* (Pearl, 2001; Basu, 2020; Schamberg et al., 2020). The direct effect of X_1 on Y is captured via the flow to edge $(1,4)$, i.e., a_{14}, and the indirect effect is captured through edges $(1,2)$ and $(1,3)$: $a_{12}a_{24} + a_{13}a_{34}$. Further, RSV decomposes the indirect effects and hence, facilitates

mediation analysis (MacKinnon et al., 2007; Heskes et al., 2020). From MacKinnon et al. (2007), "a mediating variable transmits the effect of an independent variable on a dependent variable". In Figure 9, X_1 is the independent variable and it changes from 0 to 1, causing the dependent variable Y to change from 0 to $a_{14} + a_{12}a_{24} + a_{13}a_{34}$. RSV precisely quantifies how much of the effect is transmitted through the mediating variables X_2 ($a_{12}a_{24}$) and X_3 ($a_{13}a_{34}$).

It is possible to generalize such analysis to an arbitrary DAG G with linear equations F: $X_j = \sum_{i \in \mathsf{P}_j} a_{ij} X_i \; \forall j \in \mathsf{N}^+ \setminus \{\mathsf{N}_0 \cup 0\}$. For brevity, we defer it to Appendix E (Proposition A). Such a flow seems natural for a linear model and the node-based approaches fail this sanity check. Though some edge-based approaches (e.g. DeepSHAP) obey Proposition A (easy to verify), they violate implementation invariance, as we discuss next.

5. Additional Properties of RSV

We now establish RSV obeys four additional properties that have been deemed desirable in the recent literature.

5.1. Implementation Invariance and Sensitivity

Sundararajan et al. (2017) propose "two axioms that every attribution method must satisfy": *implementation invariance* and *sensitivity*. Implementation invariance refers to the robustness of attribution w.r.t. "internal" changes in the model. We define it in our langauage next, where $g(\cdot)$ denotes the mapping from source nodes to the output, i.e., $Y = g(\mathbf{X}_{\mathsf{N}_0})$ (with $(\mathsf{E}_1, \ldots, \mathsf{E}_n)$ active).

Definition 3 (Implementation invariance)**.** *Consider models $\mathsf{M}_1 = (\mathsf{G}_1, \mathsf{F}_1)$ and $\mathsf{M}_2 = (\mathsf{G}_2, \mathsf{F}_2)$ with the same source nodes, i.e., $\mathsf{N}_0(\mathsf{G}_1) = \mathsf{N}_0(\mathsf{G}_2)$, and the same input-output mapping, i.e., $g(\cdot \mid \mathsf{F}_1) = g(\cdot \mid \mathsf{F}_2)$. An attribution method obeys implementation invariance if the source nodes receive the same attribution under the two models, i.e., $\pi_j(\mathsf{M}_1) = \pi_j(\mathsf{M}_2) \; \forall j \in \mathsf{N}_0$.*

Sundararajan et al. (2017) define sensitivity in two parts, which we formalize in our language next.

Definition 4 (Sensitivity(a))**.** *Consider background $\mathbf{X}^{(1)}_{\mathsf{N}_0}$ and foreground $\mathbf{X}^{(2)}_{\mathsf{N}_0}$ s.t. $X^{(1)}_j = X^{(2)}_j \; \forall j \in \mathsf{N}_0 \setminus \{i\}$ and $X^{(1)}_i \neq X^{(2)}_i$, where $i \in \mathsf{N}_0$. Further, suppose $\mathbf{X}^{(1)}_{\mathsf{N}_0}$ and $\mathbf{X}^{(2)}_{\mathsf{N}_0}$ result in different output values, i.e., $Y^{(1)} \neq Y^{(2)}$, where $Y^{(1)} = g(\mathbf{X}^{(1)}_{\mathsf{N}_0})$ and $Y^{(2)} = g(\mathbf{X}^{(2)}_{\mathsf{N}_0})$. Then, an attribution method obeys sensitivity(a) if $\pi_i \neq 0$.*

Definition 5 (Sensitivity(b))**.** *Consider a source node $i \in \mathsf{N}_0$. Suppose the output $g(\mathbf{X}_{\mathsf{N}_0})$ is independent of X_i. Then, an attribution method obeys sensitivity(b) if $\pi_i = 0$.*

Proposition 1. *RSV obeys implementation invariance, sensitivity(a), and sensitivity(b).*

See Appendix F.1 for a proof. As shown using an example by Dhamdhere et al. (2018), backpropagation-based approaches (e.g. DeepSHAP) lack such robustness. We show RSV's robustness using the same example in Appendix F.2.

5.2. Demand Monotonicity and Affine Scale Invariance

We now establish RSV obeys the "proper uniqueness result" of Sundararajan & Najmi (2020) by satisfying *demand monotonicity* (DM) and *affine scale invariance* (ASI).

Definition 6 (DM)**.** *Suppose $g(\cdot)$ is monotone in X_i for some $i \in \mathsf{N}_0$. Consider background $\mathbf{X}^{(1)}_{\mathsf{N}_0}$ and foreground $(\mathbf{X}^{(2)}_{\mathsf{N}_0 \setminus \{i\}}, x_i)$. An attribution method obeys DM if π_i increases as x_i increases, holding $\mathbf{X}^{(1)}_{\mathsf{N}_0}$ and $\mathbf{X}^{(2)}_{\mathsf{N}_0 \setminus \{i\}}$ fixed.*

Definition 7 (ASI)**.** *Suppose G is topologically sorted so that nodes 1 to $|\mathsf{N}_0|$ denote source nodes. Consider an alternate input-output function $g'(\cdot)$ s.t. $\forall c \in \mathbb{R}$ and $d \in \mathbb{R}$, $g(\mathbf{X}_{\mathsf{N}_0}) = g'(X_1, \ldots, (X_j - d)/c, \ldots, X_{|\mathsf{N}_0|})$, for $j \in \mathsf{N}_0$ arbitrary. An attribution method obeys ASI if $\pi_i(\mathbf{X}^{(2)}_{\mathsf{N}_0}, \mathbf{X}^{(1)}_{\mathsf{N}_0}, g) = \pi_i((X^{(2)}_1, \ldots, cX^{(2)}_j + d, \ldots, X^{(2)}_{|\mathsf{N}_0|}), (X^{(1)}_1, \ldots, cX^{(1)}_j + d, \ldots, X^{(1)}_{|\mathsf{N}_0|}), g') \; \forall i \in \mathsf{N}_0$.*

Proposition 2. *RSV obeys DM and ASI.*

See Appendix F.3 for a proof. Before concluding this section, we note that a key reason RSV obeys such properties is its top-down nature. In fact, distal ASV (Frye et al., 2020b) operates on a similar principle and satisfies such properties too. Interestingly, RSV generalizes the node-based distal ASV to a flow-based attribution. In particular, RSV attributes the same value to the source nodes as distal ASV, but in addition, gives a breakdown of how the effect flows through the graph. Proposition 3 formalizes this connection (proof in Appendix G.1).

Proposition 3. *Source nodes receive the same attribution under RSV and distal ASV, i.e., $\pi^{RSV}_j = \pi^{ASV}_j \; \forall j \in \mathsf{N}_0$.*

The notation $[\pi^{\text{ASV}}_j]_{j \in \mathsf{N}_0}$ is defined in Appendix A.3. Hence, RSV generalizes distal ASV, which only attributes to the source nodes. On the other extreme, ISV and proximate ASV attribute all the value to the parent nodes of the output. However, such an attribution is apt only if the graph is "flat" since otherwise, it violates source efficiency. As we show in Appendix G.2, RSV recovers ISV / proximate ASV under such graphs, highlighting another desirable feature of RSV.

6. An Application on Mediation Analysis

To illustrate how RSV facilitates non-linear mediation analysis, we use the "causal unfairness" example (Frye et al., 2020b). We wish to understand *unresolved discrimination* (Kilbertus et al., 2017); i.e., if *sensitive attributes* influence output without being mediated by *resolving variables*.

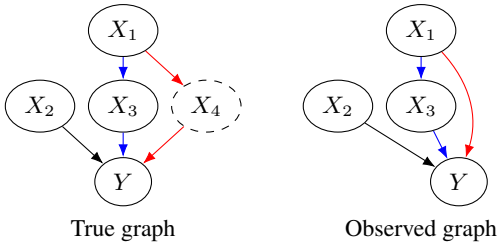

Figure 10. True (left) and observed (right) graphs for Example 3.

Example 3 (Causal unfairness)**.** *Suppose the causal graph is as in Figure 10 (left panel).* $X_1 \in [0,1]$, $X_2 \in \mathbb{R}$, $X_3 \in \{0,1\}$, *and* $X_4 \in \{0,1\}$ *represent the sensitive attribute, test score, department choice, and* unreported *referral;* $Y \in \{0,1\}$ *represents admission outcome.* X_1 *and* X_2 *are exogenous. In the data generating process (Appendix H), an applicant with a higher value of* X_1 *is more likely to apply to department 1 and is more likely to have a referral. The likelihood of admission is increasing in each of its inputs:* Y *is a Bernoulli variable with mean* $\Phi(a_2X_2 + a_3X_3 + a_4X_4)$*, where* $\Phi(\cdot)$ *is the standard normal CDF and* $(a_2, a_3, a_4) \in \mathbb{R}^3_+$ *are the probit weights. As* $a_3 \geq 0$*, department 1 is less competitive. Thus, applicants with a higher* X_1 *have a higher chance of admission due to the following two reasons: (1) they apply to the less competitive department more often ("fair channel") and (2) they have a referral more often ("unfair channel"). As noted by Frye et al. (2020b), a challenge here is that the referral variable* X_4 *is unobserved and the observed graph is as in Figure 10 (right panel). To understand unresolved discrimination, we generate multiple datasets by varying* a_4 *while fixing* (a_2, a_3) *at* $(1,1)$*. For each dataset, we fit a probabilistic model at nodes 3 (as a function of* X_1*) and* Y *(as a function of* X_1*,* X_2*,* X_3*); details in Appendix H. For each estimated model, we compute RSV by considering two applicants:* $(X_1^{(1)}, X_2^{(1)}) = (0,0)$ *and* $(X_1^{(2)}, X_2^{(2)}) = (1,0)$*; i.e., different value of the sensitive attribute, but same score. This enables us to understand if the sensitive attribute has an unfair influence on the outcome. In Figure 11, we show attributions to the fair* $(X_1 \to X_3 \to Y)$ *and the unfair* $(X_1 \to Y)$ *channels as a function of* a_4*. Attribution to the unfair channel increases with* a_4*. When* $a_4 = 1$*, the predicted admit probabilities of applicants 1 and 2 are 0.55 and 0.93. Given symmetric weights* $(a_3 = a_4)$*, RSV splits the difference equally and attributes 0.19 to each channel. When there is no unfairness* $(a_4 = 0)$*, the unfair channel receives zero attribution (nullity). Since the test scores are identical, channel* $X_2 \to Y$ *receives zero attribution (another instance of nullity).*

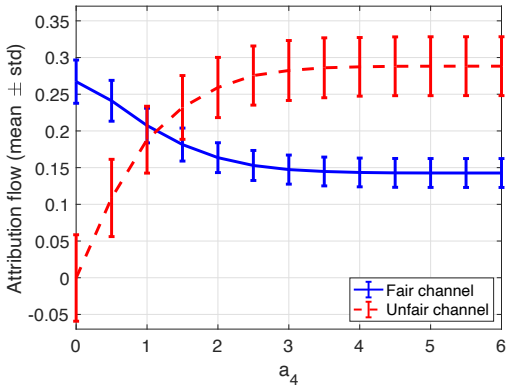

Figure 11. Numerics for Example 3. Attribution to the unfair channel increases with the level of unfair influence a_4.

Node-based approaches do not facilitate such mediation analysis; Frye et al. (2020b) used "a variant" of ASV to capture such discrimination. RSV needs no such tweak, highlighting the significance of flow-based attribution. Though backpropagation-based approaches might work well for the graph in Example 3, they violate implementation invariance.

7. Concluding Remarks

We develop a model agnostic flow-based attribution method (RSV) for a graphical model, using a sequence of recursive games. RSV generalizes existing node-based methods and uniquely satisfies a set of flow-based axioms. In addition to admitting a clean characterization for linear models and facilitating mediation analysis for non-linear models, it satisfies multiple properties deemed desirable in the literature.

This work opens up a few research directions worth pursuing. It is of interest to extend our deterministic framework to allow for stochasticity, which we believe is possible via structural equations with errors (Pearl, 2009). RSV uniquely satisfies *one* set of axioms and as with any axiomatic approach, it is of interest to explore alternative axioms. Given our foundational focus, a few practical questions such as runtime of RSV and robustness to model estimation need further investigation. Finally, connecting RSV to the causality literature (see §1) is worthwhile. In fact, Chockler & Halpern (2004) mention using SV to quantify the "degree of responsibility" in structural equations models.

Acknowledgements

We thank the ICML reviewers and the following colleagues for useful comments: David Afshartous, Patrick Bloebaum, Kailash Budhathoki, Antoine Desir, Philipp Michael Faller, Kumar Goutam, James Hensman, Garud Iyengar, Ryan McNellis, Jay Sharma, and Luke Smith.

References

Aas, K., Jullum, M., and Løland, A. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *arXiv preprint arXiv:1903.10464*, 2019.

Adadi, A. and Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):e0130140, 2015.

Basu, D. On Shapley Credit Allocation for Interpretability. *arXiv preprint arXiv:2012.05506*, 2020.

Biecek, P. DALEX: Explainers for Complex Predictive Models in R. *The Journal of Machine Learning Research*, 19(1):3245–3249, 2018.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W. Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers. In *International Conference on Artificial Neural Networks*, pp. 63–71. Springer, 2016.

Chockler, H. and Halpern, J. Y. Responsibility and Blame: A Structural-Model Approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

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

Dhamdhere, K., Sundararajan, M., and Yan, Q. How Important Is a Neuron? *arXiv preprint arXiv:1805.12233*, 2018.

Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., and Feige, I. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020a.

Frye, C., Rowat, C., and Feige, I. Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33, 2020b.

Glymour, C., Zhang, K., and Spirtes, P. Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics*, 10:524, 2019.

Halpern, J. Y. and Pearl, J. Causes and Explanations: A Structural-Model Approach. Part II: Explanations. *The British Journal for the Philosophy of Science*, 56(4):889–911, 2005.

Heskes, T., Sijben, E., Bucur, I. G., and Claassen, T. Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models. *Advances in Neural Information Processing Systems*, 33, 2020.

Janzing, D., Balduzzi, D., Grosse-Wentrup, M., Schölkopf, B., et al. Quantifying causal influences. *The Annals of Statistics*, 41(5):2324–2358, 2013.

Janzing, D., Minorics, L., and Blöbaum, P. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR, 2020.

Kilbertus, N., Carulla, M. R., Parascandolo, G., Hardt, M., Janzing, D., and Schölkopf, B. Avoiding Discrimination through Causal Reasoning. In *Advances in Neural Information Processing Systems*, pp. 656–666, 2017.

Leino, K., Sen, S., Datta, A., Fredrikson, M., and Li, L. Influence-Directed Explanations for Deep Convolutional Networks. In *2018 IEEE International Test Conference (ITC)*, pp. 1–8. IEEE, 2018.

Lundberg, S. M. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

MacKinnon, D. P., Fairchild, A. J., and Fritz, M. S. Mediation Analysis. *Annual Review of Psychology*, 58:593–614, 2007.

Molnar, C. *Interpretable Machine Learning*. Lulu.com, 2020.

Molnar, C., Casalicchio, G., and Bischl, B. Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. *arXiv preprint arXiv:2010.09337*, 2020.

Pearl, J. Direct and Indirect Effects. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 32:411–420, 2001.

Pearl, J. *Causality*. Cambridge University Press, 2009.

Peleg, B. and Sudhölter, P. *Introduction to the Theory of Cooperative Games*, volume 34. Springer Science & Business Media, 2007.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of Causal Inference*. The MIT Press, 2017.

Ribeiro, M. T., Singh, S., and Guestrin, C. Model-Agnostic Interpretability of Machine Learning. *arXiv preprint arXiv:1606.05386*, 2016.

Schamberg, G., Chapman, W., Xie, S.-P., and Coleman, T. P. Direct and Indirect Effects – An Information Theoretic Perspective. *Entropy*, 22(8):854, 2020.

Selbst, A. and Powles, J. "Meaningful Information" and the Right to Explanation. In *Conference on Fairness, Accountability and Transparency*, pp. 48–48. PMLR, 2018.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3145–3153. PMLR, 2017.

Sokol, K. and Flach, P. A. Glass-Box: Explaining AI Decisions With Counterfactual Statements Through Conversation With a Voice-enabled Virtual Assistant. In *IJCAI*, pp. 5868–5870, 2018.

Strumbelj, E. and Kononenko, I. An Efficient Explanation of Individual Classifications using Game Theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3): 647–665, 2014.

Sun, Y. and Sundararajan, M. Axiomatic Attribution for Multilinear Functions. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pp. 177–178, 2011.

Sundararajan, M. and Najmi, A. The Many Shapley Values for Model Explanation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9269–9278. PMLR, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3319–3328. PMLR, 2017.

Sundararajan, M., Dhamdhere, K., and Agarwal, A. The Shapley Taylor Interaction Index. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9259–9268. PMLR, 2020.

Turek, M. Explainable Artificial Intelligence (XAI). 2020. URL https://www.darpa.mil/program/explainable-artificial-intelligence.

Wang, J., Wiens, J., and Lundberg, S. Shapley Flow: A Graph-based Approach to Interpreting Model Predictions. In *International Conference on Artificial Intelligence and Statistics*, pp. 721–729. PMLR, 2021.