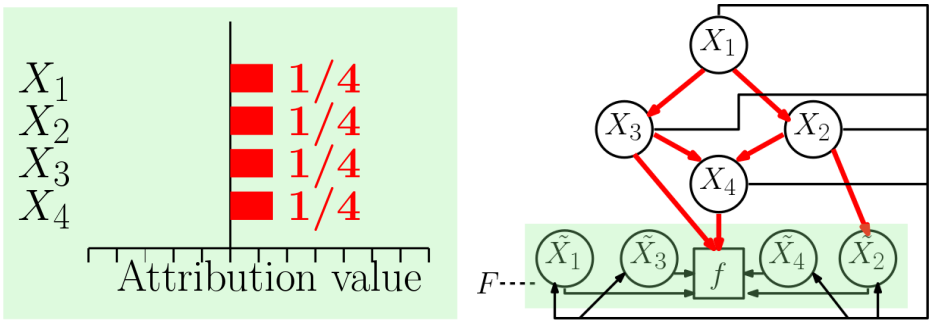


(a) On-manifold attribution (b) On-manifold boundary

Figure 8: On manifold perturbation methods can be computed using Shapley Flow with a specific explanation boundary.

6 Explanation boundary for on-manifold methods without a causal graph

On-manifold perturbation using conditional expectations can be unified with Shapley Flow using explanation boundaries (**Figure 8a**). Here we introduce $\tilde{X}_i$ as an auxiliary variable that represent the imputed version of X_i. Perturbing any feature X_i affects all input to the model ($\tilde{X}_1$, $\tilde{X}_2$, $\tilde{X}_3$, $\tilde{X}_4$) so that they respect the correlation in the data after the perturbation. When X_i has not been perturbed, $\tilde{X}_j$ treats it as missing for $i, j \in [1, 2, 3, 4]$ and would sample $\tilde{X}_j$ from the conditional distribution of X_j given non-missing predecessors. The red edges contain causal links from **Figure 1**, whereas the black edges are the causal structure used by the on-manifold perturbation method. The credit is equally split among the features because they are all correlated. Again, although giving X_1 and X_2 credit is not true to f, it is true to the model defined by F.

7 The Shapley Flow algorithm

A pseudo code implementation highlighting the main ideas for Shapley Flow is included in **Algorithm 1**. For approximations, instead of trying all edge orderings in line 15 of **Algorithm 1**, one can try random orderings and average over the number of orderings tried.

8 Shapley Flow's uniqueness proof

Without loss of generality, we can assume $\mathcal{G}$ has a single source node s. We can do this because every node in a causal graph is associated with an independent noise node (Peters et al., 2017, Chapter 6). For deterministic relationships, the function for a node doesn't depend on its noise. Treating those noise nodes as a single node, s, wouldn't have changed any boundaries that already exist in the original graph. Therefore we can assume there is a single source node s.

8.1 At most one solution satisfies the axioms

Assuming that a solution exists, we show that it must be unique.

Proof. We adapt the argument from the Shapley value uniqueness proof [3], by defining basis payoff functions as carrier games. Choose any boundary $\mathcal{B}$, we show here that any game defined on the boundary has a unique attribution. We also drop the subscript $\mathcal{B}$ in the proof as there is no ambiguity. Note that since every edge will appear in some boundary, if all boundary edges are uniquely attributed to, all edges have unique attributions. A carrier game associated with coalition (ordered list) O is a game with payoff function v^O such that $v^O(S) = 1(0)$ if coalition S starts with O (otherwise 0). By dummy player, we know that only the last edge e in O gets credit and all other edges in the cut set are dummy because a coalition is constructed in order (only adding e changes the payoff from 0 to 1). Note that in contrast with the traditional symmetry axiom (Shapley, 1953) defined

[3] `https://ocw.mit.edu/courses/economics/14-126-game-theory-spring-2016/lecture-notes/MIT14_126S16_cooperative.pdf`

Algorithm 1 Shapley Flow pseudo code

Input: A computational graph $\mathcal{G}$ (each node i has a function f_i), foreground sample $\mathbf{x}$, background sample $\mathbf{x}'$
Output: Edge attribution $\phi : E \to \mathbb{R}$
Initialization:
$\mathcal{G}$: add an new source node pointing to original source nodes.

function ShapleyFlow($\mathcal{G}$, $\mathbf{x}'$, $\mathbf{x}$)
  Initialize($\mathcal{G}$, $\mathbf{x}'$, $\mathbf{x}$) ▷ Set up game ν for any boundary in $\mathcal{G}$
  $s \leftarrow$ source($\mathcal{G}$) ▷ Obtain the source node
  return DFS(s, {}, [])
end function

function DFS(s, D, S)
  ▷ s is a node, D is the data side of the current boundary, S is coalition
  ▷ Using Python list slice notation
  Initialize ϕ to output 0 for all edges
  if IsSinkNode(s) **then**
    ▷ Here we overload D to refer to its boundary
    $\phi(S[-1]) \leftarrow \nu_D(S) - \nu_D(S[:-1])$ ▷ Difference in output is attributed to the edge
    return ϕ
  end if

  for $p \leftarrow$ AllOrderings(Children(s)) **do** ▷ Try all orderings/permutations of the node's children
    for $c \leftarrow p$ **do** ▷ Follow the permutation to get the node one by one
      edgeCredit $\leftarrow$ DFS(c, $D \cup \{s\}$, $S + [(s, c)]$) ▷ Recurse downward
      $\phi \leftarrow \phi + \frac{\text{edgeCredit}}{\text{NumChildren}(s)!}$ ▷ Average attribution over number of runs
      $\phi(S[-1]) \leftarrow \phi(S[-1]) + \frac{\text{edgeCredit}(s,c)}{\text{NumChildren}(s)!}$ ▷ Propagate upward
    end for
  end for
  return ϕ
end function

on a set of players, the symmetry axiom is not explicit in our case (it is made implicitly) because not all edges in the carrier game are symmetric with each other (observe that e is different from all other edges, which are dummy), thus we do not need an explicit symmetry axiom to argue for unique attribution in the carrier game. Furthermore, e must be an edge in the boundary to form a valid game because boundary edges are the only edges that are connected to the model defined by the boundary. Therefore we give 0 credit to edges in the cut set other than e (because they are *dummy players*). By the *efficiency axiom*, we give $\sum_{h\in\tilde{\mathcal{H}}} \frac{\nu_{\mathcal{B}}(h)}{|\tilde{\mathcal{H}}|} - \nu_{\mathcal{B}}([])$ credit to e where $\tilde{\mathcal{H}}$ is the set of all possible boundary consistent histories as defined in **Section 3.3**. This uniquely attributed the boundary edges for this game.

We show that the set of carrier games associated with every coalition that ends in a boundary edge (denoted as $\hat{C}$) form basis functions for all payoff functions associated with the system. Recall from **Section 3.2** that $\tilde{C}$ is the set of *boundary consistent coalitions*. We show here that payoff value on coalitions from $\tilde{C}$ is redundant given $\hat{C}$. Note that $\tilde{C} \setminus \hat{C}$ represents all the coalitions that do not end in a boundary edge. For $c \in \tilde{C} \setminus \hat{C}$, $v^O(c) = v^O(c[:-1])$ (using Python's slice notation on list) because only boundary edges are connected to the model defined by the boundary. Therefore it suffices to show that v^O is linearly independent for $O \in \hat{C}$. For a contradiction, assume for all $c \in \hat{C}$, $\sum_{O\subseteq\hat{C}} \alpha^O v^O(c) = 0$, with some non zero $\alpha^O \in \mathbb{R}$ (definition of linear dependence). Let S be a coalition with minimal length such that $\alpha^S \neq 0$. We have $\sum_{O\subseteq\hat{C}} \alpha^O v^O(S) = \alpha^S$, a contradiction.

Therefore for any ν we have unique α's such that $\nu = \sum_{O\subseteq\hat{C}} \alpha^O v^O$. Using the *linearity axiom*, we have

$$\phi_\nu = \phi_{\sum_{O\subseteq\hat{C}} \alpha^O \nu^O} = \sum_{O\subseteq\hat{C}} \alpha^O \phi_{\nu^O}$$

The uniqueness of α and ϕ_{ν^O} makes the attribution unique if a solution exists. Axioms used in the proof are italicized.

□

8.2 Shapley Flow satisfies the axioms

Proof. We first demonstrate how to generate all boundaries. Then we show that Shapley Flow gives boundary consistent attributions. Following that, we look at the set of histories that can be generated by DFS in boundary $\mathcal{B}$, denoted as $\Pi^{\text{dfs}}_{\mathcal{B}}$. We show that $\Pi^{\text{dfs}}_{\mathcal{B}} = \tilde{\mathcal{H}}_{\mathcal{B}}$. Using this fact, we check the axioms one by one.

- Every boundary can be "grown" one node at a time from $D = \{s\}$ where s is the source node: Since the computational graph $\mathcal{G}$ is a directed acyclic graph (DAG), we can obtain a topological ordering of the nodes in $\mathcal{G}$. Starting by including the first node in the ordering (the source node s), which defines a boundary as $(D = \{s\}, F = \text{Nodes}(\mathcal{G})\backslash D)$, we grow the boundary by adding nodes to D (removing nodes from F) one by one following the topological ordering. This ordering ensures the corresponding explanation boundary is valid because the cut set only flows from D to F (if that's not true, then one of the dependency nodes is not in D, which violates topological ordering).

  Now we show every boundary can be "grown" in this fashion. In other words, starting from an arbitrary boundary $\mathcal{B}_1 = (D_1, F_1)$, we can "shrink" one node at a time to $D = \{s\}$ by reversing the growing procedure. First note that, D_1 must have a node with outgoing edges only pointing to nodes in F_1 (if that's not the case, we have a cycle in this graph because we can always choose to go through edges internal to D_1 and loop indefinitely). Therefore we can just remove that node to arrive at a new boundary (now its incoming edges are in the cut set). By the same argument, we can keep removing nodes until $D = \{s\}$, completing the proof.

- Shapley Flow gives boundary consistent attributions: We show that every boundary grown has edge attribution consistent with the previous boundary. Therefore all boundaries have consistent edge attribution because the boundary formed by any two boundary's common set of nodes can be grown into those two boundaries using the property above. Let's focus on the newly added node c from one boundary to the next. Note that a property of depth first search is that every time c's value is updated, its outgoing edges are activated in an atomic way (no other activation of edges occur between the activation of c's outgoing

edges). Therefore, the change in output due to the activation of new edges occur together in the view of edges upstream of c, thus not changing their attributions. Also, since c's outgoing edges must point to the model defined by the current boundary (otherwise it cannot be a valid topological ordering), they don't have down stream edges, concluding the proof.

- $\Pi_{\mathcal{B}}^{\text{dfs}} = \tilde{\mathcal{H}}_{\mathcal{B}}$: Since attribution is boundary consistent, we can treat the model as a blackbox and only look at the DFS ordering on the data side. Observe that the edge traversal ordering in DFS is a valid history because a) every edge traversal can be understood as a message received through edge , b) when every message is received, the node's value is updated, and c) the new node's value is sent out through every outgoing edge by the recursive call in DFS. Therefore the two side of the equation are at least holding the same type of object.

  We first show that $\Pi_{\mathcal{B}}^{\text{dfs}} \subseteq \tilde{\mathcal{H}}_{\mathcal{B}}$. Take $h \in \Pi_{\mathcal{B}}^{\text{dfs}}$, we need to find a history h^* in $\mathcal{B}^*$ such that a) h can be expanded into h^* and b) for any boundary, there is a history in that boundary that can be expanded into h^*. Let h^* be any history expanded using DFS that is aligned with h. To show that every boundary can expand into h^*, we just need to show that the boundaries generated through the growing process introduced in the first bullet point can be expanded into h^*. The base case is $D = \{s\}$. There must have an ordering to expand into h^* because h^* is generated by DFS, and that DFS ensures that every edge's impact on the boundary is propagated to the end of computation before another edge in D is traversed. Similarly, for the inductive step, when a new node c is added, we just follow the expansion of its previous boundary to reach h^*.

  Next we show that $\tilde{\mathcal{H}}_{\mathcal{B}} \subseteq \Pi_{\mathcal{B}}^{\text{dfs}}$. First observe that for history h_1 in $\mathcal{B}_1 = (D_1, F_1)$ and history h_2 in $\mathcal{B}_2 = (D_2, F_2)$ with $F_2 \subseteq F_1$, if h_1 cannot be expanded into h_2, then $HE(h_1) \cap HE(h_2) = \emptyset$ because they already have mismatches for histories that doesn't involve passing through $\mathcal{B}_1$. Assume we do have $h \in \tilde{\mathcal{H}}_{\mathcal{B}}$ but $h \notin \Pi_{\mathcal{B}}^{\text{dfs}}$. To derive a contradiction, we shrink the boundary one node at a time from $\mathcal{B}$, again using the procedure described in the first bullet point. We denote the resulting boundary formed by removing n nodes as $\mathcal{B}_{-n}$. Since h is assumed to be boundary consistent, there exist $h_{\mathcal{B}_{-1}} \in \mathcal{H}_{\mathcal{B}_{-1}}$ such that $h_{\mathcal{B}_{-1}}$ must be able to expand into h. Say the two boundaries differ in node c. Note that any update to c crosses $\mathcal{B}_{-1}$, therefore its impact must be reached by F before another event occurs in D_{-1}. Since all of c's outgoing edges crosses $\mathcal{B}$, any ordering of messages sent through those edges is a DFS ordering from c. This means that if $h_{\mathcal{B}_{-1}}$ can be reached by DFS, so can $h_{\mathcal{B}}$, violating the assumption. Therefore, $h_{\mathcal{B}_{-1}} \notin \Pi_{\mathcal{B}_{-1}}^{\text{dfs}}$ and $h_{\mathcal{B}_{-1}} \in \tilde{\mathcal{H}}_{\mathcal{B}_{-1}}$ (the latter because $h_{\mathcal{B}_{-1}}$ can expand into a history that is consistent with all boundaries by first expanding into h). We run the same argument until $D = \{s\}$. This gives a contradiction because in this boundary, all histories can be produced by DFS.

- Efficiency: Since we are attributing credit by the change in the target node's value following a history h given by DFS, the target for this particular DFS run is thus $\nu_{\mathcal{B}}(h) - \nu_{\mathcal{B}}([])$. Average over all DFS runs and noting that $\tilde{\mathcal{H}}_{\mathcal{B}} = \Pi_{\mathcal{B}}^{\text{dfs}}$ gives the target $\sum_{h \in \tilde{\mathcal{H}}_{\mathcal{B}}} \nu_{\mathcal{B}}(h)/|\tilde{\mathcal{H}}_{\mathcal{B}}| - \nu_{\mathcal{B}}([])$. Noting that each update in the target node's value must flow through one of the boundary edges. Therefore the sum of boundary edges' attribution equals to the target.

- Linearity: For two games of the same boundary v and u, following any history, the sum of output differences between the two games is the output difference of the sum of the two games, therefore ϕ_{v+u} would not differ from $\phi_v + \phi_u$. It's easy to see that extending addition to any linear combination wouldn't matter.

- Dummy player: Since Shapley Flow is boundary consistent, we can just run DFS up to the boundary (treat F as a blackbox). Since every step in DFS remains in the coalition $\tilde{\mathcal{C}}_{\mathcal{B}}$ because $\Pi_{\mathcal{B}}^{\text{dfs}} \subseteq \tilde{\mathcal{H}}_{\mathcal{B}}$, if an edge is dummy, every time it is traversed through by DFS, the output won't change by definition, thus giving it 0 credit.

□

Therefore Shapley Flow uniquely satisfies the axioms. We note that efficiency requirement simplifies to $f(\boldsymbol{x}) - f(\boldsymbol{x'})$ when applying it to an actual model because all histories from DFS would lead the target node to its target value. We can prove a stronger claim that actually all nodes would reach its target value when DFS finishes. To see that, we do an induction on a topological ordering of the nodes. The source nodes reaches its final value by definition. Assume this holds for the k^{th} node. For the $k+1^{\text{th}}$ node, its parents achieves target

value by induction. Therefore DFS would make the parents' final values visible to this node, thus updating it to the target value.

9 Causal graphs

While the nutrition dataset is introduced in the main text, we describe an additional dataset to further demonstrate the usefulness of Shapley Flow. Moreover, we describe in detail how the causal relationship is estimated. The resulting causal graphs for the nutrition dataset and the income dataset are visualized in **Figure 9**.

9.1 The Census Income dataset

The Census Income dataset consists of $32,561$ samples with 12 features. The task is to predict whether one's annual income exceeds $50k$. We assume a causal graph, similar to that used by Frye et al. (2019) (**Figure 9b**). Attributes determined at birth e.g., sex, native country, and race act as source nodes. The remaining features (marital status, education, relationship, occupation, capital gain, work hours per week, capital loss, work class) have fully connected edges pointing from their causal ancestors. All features have a directed edge pointing to the model.

9.2 Causal Effect Estimation

Given the causal structure described above, we estimate the relationship among variables using XGBoost. More specifically, using an 80/20 train test split, we use XGBoost to learn the function for each node. If the node value is categorical, we train to minimize cross entropy loss. Otherwise, we minimize mean squared error. Models are fitted by 100 XGBoost trees with a max depth of 3 for up to 1000 epochs. Since features are rarely perfectly determined by their dependency node, we add independent noise nodes to account for this effect. That is, each non-sink node is pointed to by a unique noise node that account for the residue effect of the prediction.

Depending on whether the variable is discrete or continuous, we handle the noise differently. For continuous variables, the noise node's value is the residue between the prediction and the actual value. For discrete variables, we assume the actual value is sampled from the categorical distribution specified by the prediction. Therefore the noise node's value is any possible random number that could result in the actual value. As a concrete example for handling discrete variable, consider a binary variable y, and assume the trained categorical function f gives $f(\boldsymbol{x}) = [0.3, 0.7]$ where $\boldsymbol{x}$ is the foreground value of the input to predict y. We view the data generation as the following. The noise term associated with y is treated as a uniform random variable between 0 and 1. If it lands within 0 to 0.3, y is sampled to be 0, otherwise 1 (matching the categorical function of 70% chance of sampling y to be 1). Now if we observe the foreground value of y to be 0, it means the foreground value of noise must be uniform between 0 to 0.3. Although we cannot infer the exact value of the noise, we can sample the noise from 0 to 0.3 multiple times and average the resulting attribution.

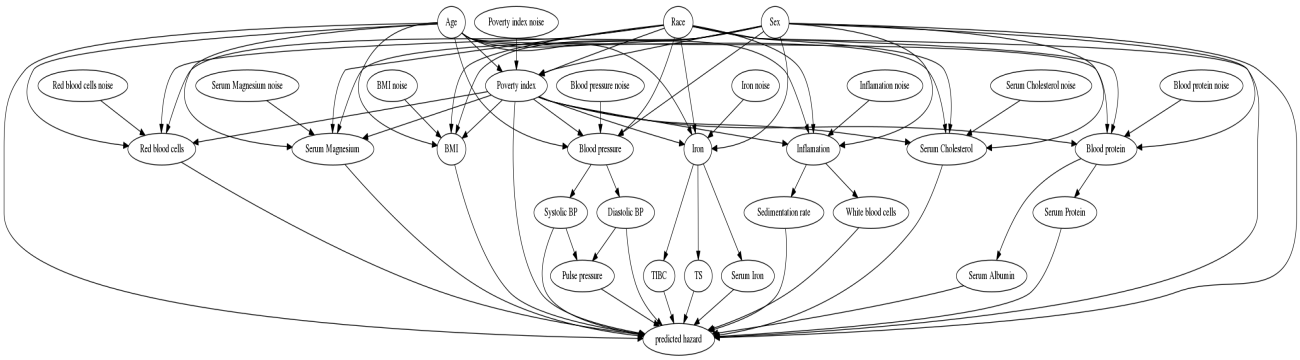
10 Additional Results

In this section, we first present additional sanity checks with synthetic data. Then we show additional examples from both the nutrition and income datasets to demonstrate how a complete view of boundaries should be preferable over single boundary approaches.

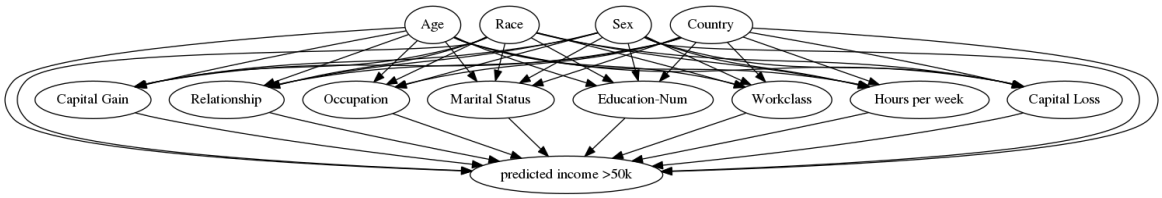
10.1 Additional Sanity Checks

We include further sanity check experiments in this section. The first sanity check consists of a chain with 4 variables. Each node along the chain is an identical copy of its predecessor and the function to explain only depends on X_4 (**Figure 10**). The dataset is created by sampling $X_1 \sim \mathcal{N}(0, 1)$, that is a standard normal distribution, with 1000 samples. We use the first sample as background, and explain the second sample (one can choose arbitrary samples to obtain the same insights). As shown in **Figure 10**, independent SHAP fails to show the indirect impact of X_1, X_2, and X_3, ASV fails to show the direct impact of X_4, on manifold SHAP fails to fully capture both the direct and indirect importance of any edge.

The second sanity check consists of linear models as described in **Section 4.3**. We include the full result with

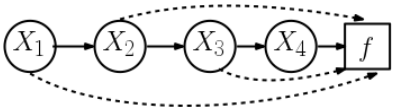


(a) Causal graph for the nutrition dataset



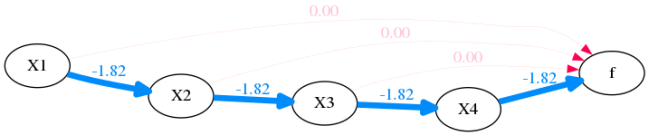
(b) Causal graph for the Census Income dataset

Figure 9: The causal graphs we used for the two real datasets. Note that each node in the causal graph for (a) is given a noise node to account for random effects. The noise nodes are omitted for better readability for (b). The resulting causal structures are over-simplifications of the true causal structure; the relationship between source nodes (e.g., race and sex) and other features is far more complex. They are used as a proof of concept to show both the direct and indirect effect of features on the prediction output.



(a) chain dataset

	Independent	On-manifold	ASV
X4	-1.82	-0.45	0.0
X1	0.0	-0.45	-1.82
X3	0.0	-0.45	0.0
X2	0.0	-0.45	0.0



(b) Shapley Flow

Figure 10: **(a)** The chain dataset contains exact copies of nodes. The dashed edges denotes dummy dependencies. **(b)** While Shapley Flow shows the entire path of influence, other baselines fails to capture either direct and indirect effects.

Methods	Income	Nutrition	Synthetic
Independent	**0.0** (± 0.0)	**0.0** (± 0.0)	**0.0** (± 0.0)
On-manifold	0.4 (± 0.3)	1.3 (± 2.5)	0.8 (± 0.7)
ASV	0.4 (± 0.6)	1.5 (± 3.3)	1.2 (± 1.4)
Shapley Flow	**0.0** (± 0.0)	**0.0** (± 0.0)	**0.0** (± 0.0)

Table 2: Shapley Flow and independent SHAP have lower mean absolute error (std) for direct effect of features on linear models.

Methods	Income	Nutrition	Synthetic
Independent	0.1 (± 0.2)	0.8 (± 2.7)	1.1 (± 1.4)
On-manifold	0.4 (± 0.3)	0.9 (± 1.6)	1.5 (± 1.5)
ASV	0.1 (± 0.1)	0.6 (± 1.9)	1.1 (± 1.5)
Flow	**0.0** (± 0.0)	**0.0** (± 0.0)	**0.0** (± 0.0)

Table 3: Shapley Flow and ASV have lower mean absolute error (std) for indirect effect on linear models.

the income dataset added in **Table 2** and **Table 3** for direct and indirect effects respectively. The trend for the income dataset algins with the nutrition and synthetic dataset: only Shapley Flow makes no mistake for estimating both direct and indirect impact. Independent Shap only does well for direct effect. ASV only does well for indirect effects (it only reaches zero error when evaluated on source nodes).

10.2 Additional examples

In this section, we analyze another example from the nutrition dataset (**Figure 11**) and 3 additional example from the adult censor dataset.

Independent SHAP ignores the indirect impact of features. Take an example from the nutrition dataset (**Figure 11**). The race feature is given low attribution with independent SHAP, but high importance in ASV. This happens because race, in addition to its direct impact, indirectly affects the output through blood pressure, serum magnesium, and blood protein, as shown by Shapley Flow (**Figure 11a**). In particular, race partially accounts for the impact of serum magnesium because changing race from Black to White on average increases serum magnesium by 0.07 meg/L in the dataset (thus partially explaining the increase in serum magnesium changing from the background sample to the foreground). Independent SHAP fails to account for the indirect impact of race, leaving the user with a potentially misleading impression that race is irrelevant for the prediction.

On-manifold SHAP provides a misleading interpretation. With the same example (**Figure 11**), we observe that on-manifold SHAP strongly disagrees with independent SHAP, ASV, and Shapley Flow on the importance of age. Not only does it assign more credit to age, it also flips the sign, suggesting that age is protective. However, **Figure 12a** shows that age and earlier mortality are positively correlated; then how could age be protective? **Figure 12b** provides an explanation. Since SHAP considers all partial histories regardless of the causal structure, when we focus on serum magnesium and age, there are two cases: serum magnesium updates before or after age. We focus on the first case because it is where on-manifold SHAP differs from other baselines (all baselines already consider the second case as it satisfies the causal ordering). When serum magnesium updates before age, the expected age given serum magnesium is higher than the foreground age (yellow line above the black marker). Therefore when age updates to its foreground value, we observe a decrease in age, leading to a decrease in the output (so age appears to be protective). Serum magnesium is just one variable from which age steals credit. Similar logic applies to TIBC, red blood cells, serum iron, serum protein, serum cholesterol, and diastolic BP. From both an in/direct impact perspective, on-manifold perturbation can be misleading since it is based not on causal but on observational relationships.

ASV ignores the direct impact of features. As shown in **Figure 11**, serum magnesium appears to be more important in independent SHAP compared to ASV. From Shapley Flow (**Figure 11a**), this difference is explained by race as its edge to serum magnesium has a negative impact. However, looking at ASV alone, one fails to understand that intervening on serum magnesium could have a larger impact on the output.

Shapley Flow shows both direct and indirect impacts of features. Focusing on the attribution given by Shapley Flow (**Figure 11a**). We not only observe similar direct impacts in variables compared to inde-

pendent SHAP, but also can trace those impacts to their source nodes, similar to ASV. Furthermore, Shapley Flow provides more detail compared to other approaches. For example, using Shapley Flow we gain a better understanding of the ways in which race impacts survival. The same goes for all other features. This is useful because causal links can change (or break) over time. Our method provides a way to reason through the impact of such a change.

Figure 13 gives an example of applying Shapley Flow and baselines on the income dataset. Note that the attribution to capital gain drops from independent SHAP to on-manifold SHAP and ASV. From Shapley Flow, we know the decreased attribution is due to age and race. More examples are shown in **Figure 14** and **15**.

10.3 A global understanding with Shapley Flow

In addition to explaining a particular example, one can explain an entire dataset with Shapley Flow. Specifically, for multi-class classification problems, we take the average of attributions for the probability predicted for the actual class, in accordance with (Frye et al., 2019). A demonstration on the income dataset using 1000 randomly selected examples is included in **Figure 16**. As before, we use a single shared background sample for explanation. Here, we observe that although the relative importance across independent SHAP, on-manifold SHAP, and ASV are similar, age and sex have opposite direct versus indirect impact as shown by Shapley Flow.

10.4 Example with multiple background samples

An example with 100 background samples is shown in **Figure 17**. Shapley Flow shows a holistic picture of feature importance, while other baselines only show part of the picture.

Independent SHAP ignores the indirect impact of features. Take an example from the nutrition dataset (**Figure 17**). Independent SHAP only considers the direct impact of systolic blood pressure, and ignores its potential impact on pulse pressure (as shown by Shapley Flow in **Figure 17a**). If the causal graph is correct, independent SHAP would underestimate the effect of intervening on Systolic BP.

On-manifold SHAP provides a misleading interpretation. With the same example (**Figure 17**), we observe that on-manifold SHAP strongly disagrees with independent SHAP, ASV, and Shapley Flow on the importance of age. In particular, it flips the sign on the importance of age. Since the background age (50) is very close to the foreground age (51), we would not expected age to significantly affect the prediction. **Figure 18b** provides an explanation. Since SHAP considers all partial histories regardless of the causal structure, when we focus on systolic blood pressure and age, there are two cases: systolic blood pressure updates before or after age. We focus on the first case because it is where on-manifold SHAP differs from other baselines (all baselines already consider the second case as it satisfies the causal ordering). When systolic blood pressure updates before age, the expected age given systolic blood pressure is lower than the foreground age (yellow line below the black marker). Therefore when age updates to its foreground value, we observe a large increase in age, leading to a increase in the output (so age appears to be riskier). from both an in/direct impact perspective, on-manifold perturbation can be misleading since it is based not on causal but on observational relationships.

ASV ignores the direct impact of features. As shown in **Figure 17**, ASV gives no credit systolic blood pressure because it is an intermediate node. However, it is clear from Shapley Flow that intervening on systolic blood pressure has a large impact on the outcome.

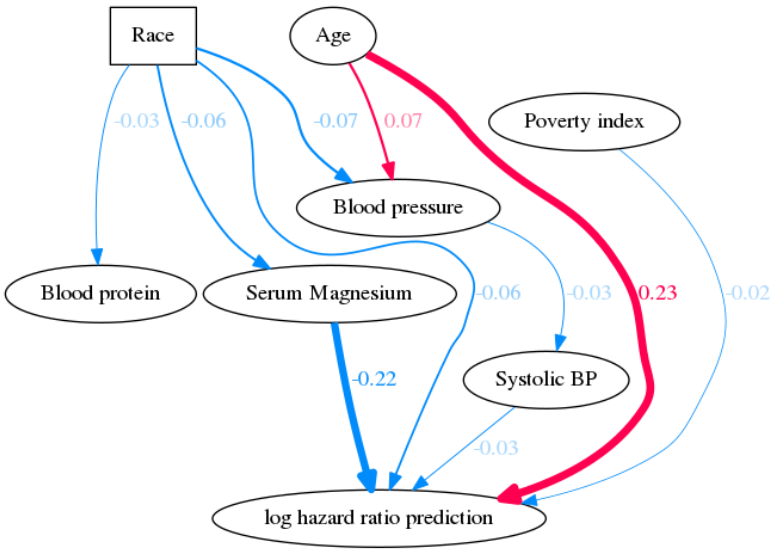
Shapley Flow shows both direct and indirect impacts of features. Focusing on the attribution given by Shapley Flow (**Figure 17a**). We not only observe similar direct impacts in variables compared to independent SHAP, but also can trace those impacts to their source nodes, similar to ASV.

11 Considering all histories could lead to boundary inconsistency

In this section, we give an example of how considering all history $\mathcal{H}$ in the axioms (as opposed to $\tilde{\mathcal{H}}$) could lead to inconsistent attributions across boundaries. Consider two cuts for the same causal graph shown in **Figure 19**. Note that both the green and the red cut share the edge "a". We have 8 possible message transmission histories ('c', 'b' can be transmitted only after 'd' has been transmitted): $\{[a,d,c,b],[a,d,b,c],[d,a,c,b],[d,a,b,c],[d,c,a,b],[d,c,b,a],[d,b,a,c],[d,b,c,a]\}$. We use the same notation for carrier games (defined in **Section 8**) and construct a game as the following:

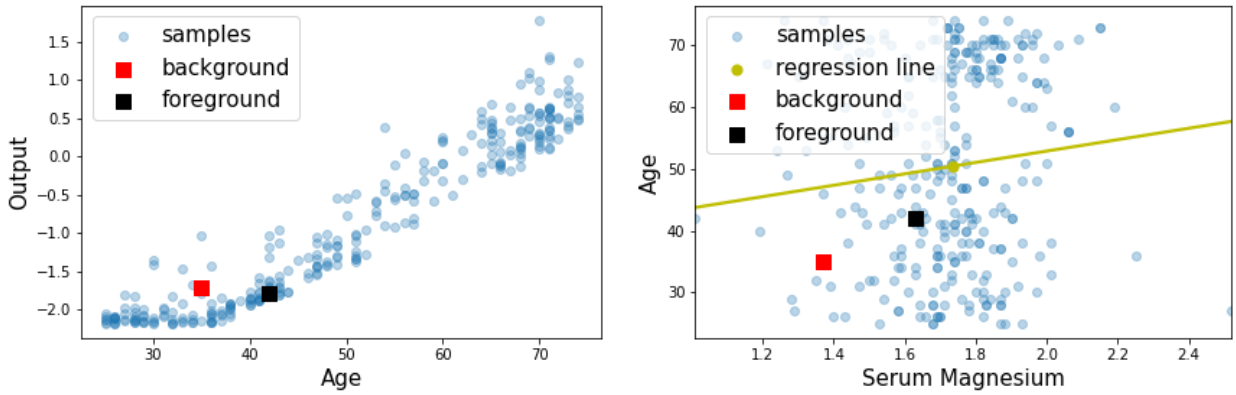
Top features	Age	Serum Magnesium	Race
Background sample	35.0	1.37	Black
Foreground sample	42.0	1.63	white

Attributions	Independent	On-manifold	ASV
Age	0.23	-0.38	0.3
Serum Magnesium	-0.21	-0.02	-0.15
Race	-0.06	0.04	-0.24
Pulse pressure	0.0	-0.08	0.0
Diastolic BP	0.0	0.08	0.0
Serum Cholesterol	0.0	0.07	0.0
Serum Protein	0.01	0.06	0.0
Serum Iron	0.0	0.05	0.0
Poverty index	-0.02	0.01	-0.01
Systolic BP	-0.03	-0.01	0.0
Red blood cells	0.0	0.05	0.0
Blood protein	0.0	0.0	0.04
TIBC	0.0	0.04	0.0
Blood pressure	0.0	0.0	-0.03
TS	0.0	0.03	0.0
BMI	-0.0	-0.03	-0.0
Sex	0.0	0.02	0.0
Serum Albumin	0.0	-0.01	0.0
White blood cells	0.01	-0.01	0.0
Sedimentation rate	0.0	0.01	0.0
Inflamation	0.0	0.0	0.01
Iron	0.0	0.0	0.0



(a) Shapley Flow

Figure 11: Comparison among baselines on a sample (top table) from the nutrition dataset, showing top 10 features/edges. As noted in the main text this graph is an oversimplification and is not necessarily representative of the true underlying causal relationship.



(a) Age vs. output

(b) Age vs. magnesium

Figure 12: Age appears to be protective in on-manifold SHAP because it steals credit from other variables.

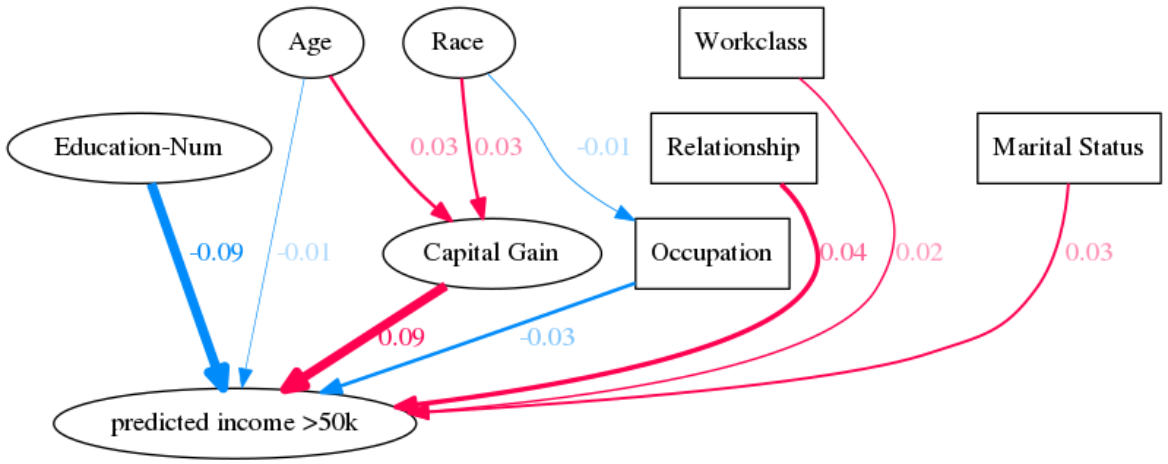
$v_{red} = v^{dca}_{red} - v^{dcab}_{red} + v^{dba}_{red} - v^{dbac}_{red}$

Because of the linearity axiom, we have $\phi_{v_{red}}(a) > 0, \phi_{v_{red}}(b) < 0, \phi_{v_{red}}(c) < 0, \phi_{v_{red}}(d) = 0$.

However, when we consider the green boundary, the ordering $dcab$ and $dbac$ does not exist because in the green boundary A and Y are assumed to be a black-box. Therefore, $v_{green} = \mathbf{0}$, which means a is now a dummy edge: $\phi_{v_{green}}(a) = 0 \neq \phi_{v_{red}}(a)$. This demonstrate that we cannot consider all histories in $\mathcal{H}$ and being boundary consistent.

	Background sample	Foreground sample
Age	39	35
Workclass	State-gov	Federal-gov
Education-Num	13	5
Marital Status	Never-married	Married-civ-spouse
Occupation	Adm-clerical	Farming-fishing
Relationship	Not-in-family	Husband
Race	White	Black
Sex	Male	Male
Capital Gain	2174	0
Capital Loss	0	0
Hours per week	40	40
Country	United-States	United-States

	Independent	On-manifold	ASV
Education-Num	-0.12	-0.11	-0.09
Relationship	0.05	0.06	0.04
Capital Gain	0.09	0.01	0.03
Occupation	-0.03	-0.07	-0.02
Marital Status	0.04	0.05	0.03
Workclass	0.02	0.03	0.02
Race	-0.01	-0.03	0.01
Age	-0.01	-0.01	0.02
Capital Loss	0.0	0.03	0.0
Country	0.0	0.03	0.0
Sex	0.0	0.03	0.0
Hours per week	0.0	0.0	0.0

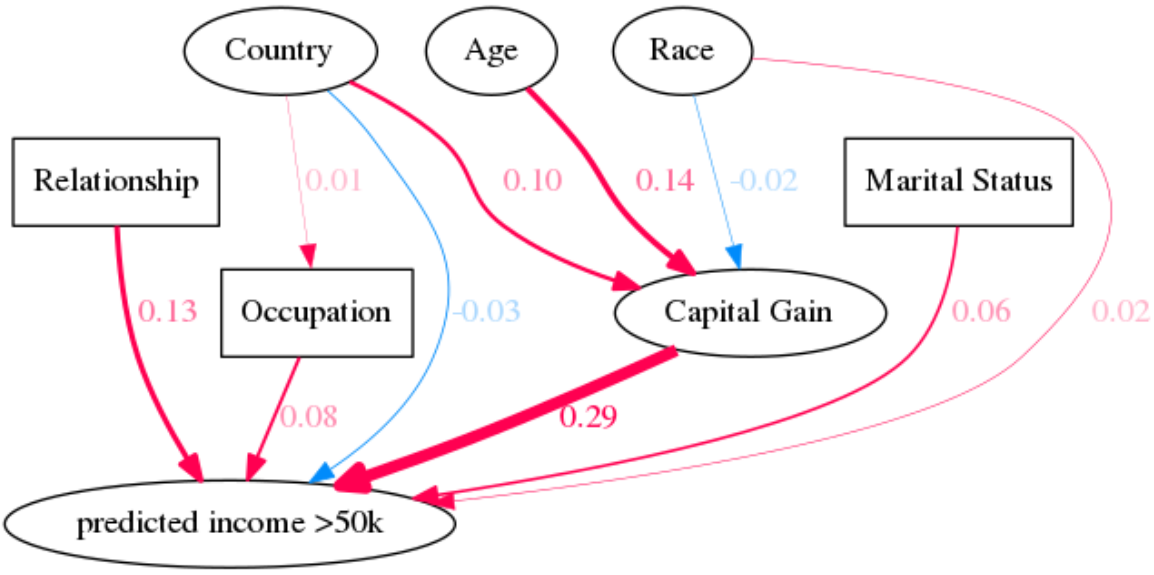


(a) Shapley Flow

Figure 13: Comparison between independent SHAP, on-manifold SHAP, ASV, and Shapley Flow on a sample from the income dataset. Shapley flow shows the top 10 links. The direct impact of capital gain is not represented by on-manifold SHAP. As noted in the text this graph is based on previous work and is not necessarily representative of the true underlying causal relationship.

	Background sample	foreground sample
Age	39	30
Workclass	State-gov	State-gov
Education-Num	13	13
Marital Status	Never-married	Married-civ-spouse
Occupation	Adm-clerical	Prof-specialty
Relationship	Not-in-family	Husband
Race	White	Asian-Pac-Islander
Sex	Male	Male
Capital Gain	2174	0
Capital Loss	0	0
Hours per week	40	40
Country	United-States	India

	Independent	On-manifold	ASV
Relationship	0.17	0.04	0.13
Capital Gain	0.22	0.01	0.07
Occupation	0.1	0.06	0.07
Marital Status	0.08	0.06	0.07
Country	-0.04	0.07	0.07
Age	-0.0	-0.02	0.13
Education-Num	0.0	0.12	0.0
Race	0.02	0.07	0.0
Workclass	0.0	0.06	0.0
Hours per week	0.0	0.03	0.0
Sex	0.0	0.03	0.0
Capital Loss	0.0	0.01	0.0

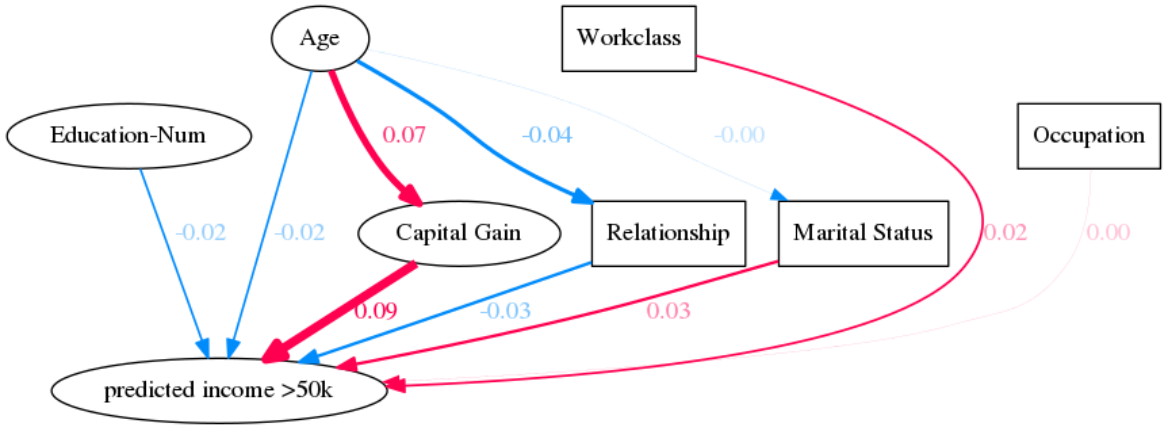


(a) Shapley Flow

Figure 14: Comparison between independent SHAP, on-manifold SHAP, ASV, and Shapley Flow on a sample from the income dataset. Shapley flow shows the top 10 links. The indirect impact of age is only highlighted by Shapley Flow and ASV. As noted in the text this graph is based on previous work and is not necessarily representative of the true underlying causal relationship.

	Background sample	Foreground sample
Age	39	30
Workclass	State-gov	Federal-gov
Education-Num	13	10
Marital Status	Never-married	Married-civ-spouse
Occupation	Adm-clerical	Adm-clerical
Relationship	Not-in-family	Own-child
Race	White	White
Sex	Male	Male
Capital Gain	2174	0
Capital Loss	0	0
Hours per week	40	40
Country	United-States	United-States

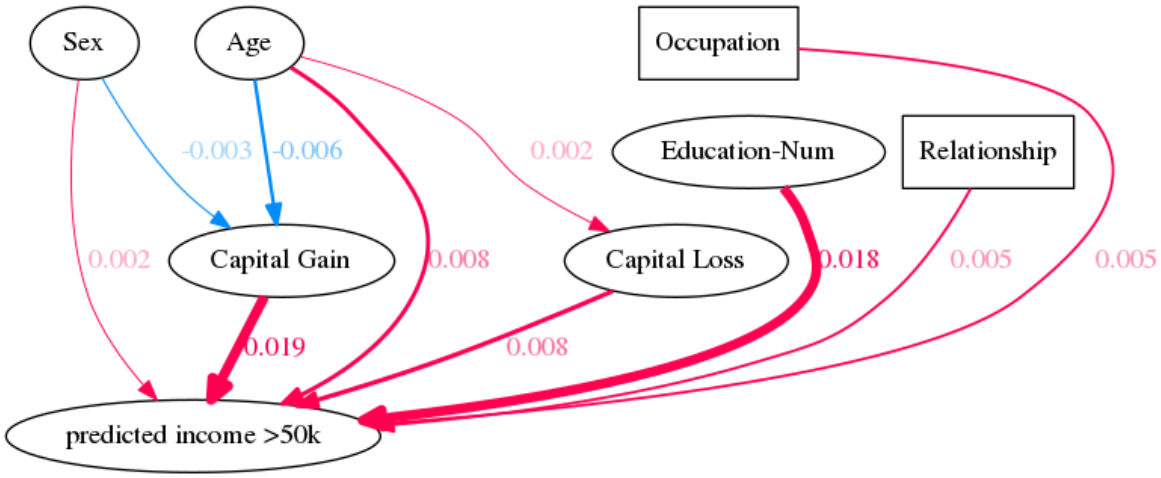
Attributions	Independent	On-manifold	ASV
Marital Status	0.03	0.08	0.03
Capital Gain	0.06	0.02	0.02
Workclass	0.03	0.03	0.02
Relationship	-0.01	-0.11	0.01
Education-Num	-0.02	0.01	-0.02
Age	-0.02	-0.03	0.01
Country	0.0	0.03	0.0
Capital Loss	0.0	0.03	0.0
Occupation	0.0	-0.03	0.0
Sex	0.0	0.03	0.0
Race	0.0	0.02	0.0
Hours per week	0.0	-0.0	0.0



(a) Shapley Flow

Figure 15: Comparison between independent SHAP, on-manifold SHAP, ASV, and Shapley Flow on a sample from the income dataset. Shapley flow shows the top 10 links. Note that although age appears to be not important for all baselines, its impact through different causal edges are opposite as shown by Shapley Flow.

	Independent	On-manifold	ASV
Capital Gain	0.02	0.02	0.03
Education-Num	0.02	0.03	0.02
Age	0.01	0.01	0.01
Occupation	0.0	0.01	0.0
Capital Loss	0.01	-0.0	0.01
Relationship	0.01	0.0	0.0
Hours per week	0.0	0.01	-0.0
Sex	0.0	-0.01	0.0
Country	0.0	-0.01	0.0
Marital Status	-0.0	0.0	-0.0
Race	0.0	-0.01	-0.0
Workclass	0.0	-0.0	-0.0

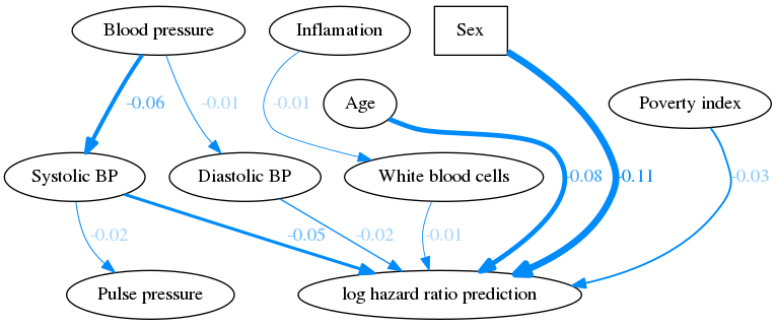


(a) Shapley Flow

Figure 16: Comparison of global understanding between independent SHAP, on-manifold SHAP, ASV, and Shapley Flow on the income dataset. Showing only the top 10 attributions for Shapley Flow for visual clarity.

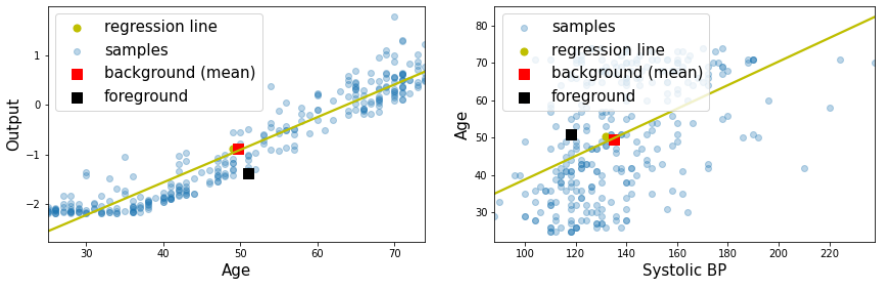
Top features	Sex	Age	Systolic BP
Background mean	NaN	50	135
Foreground sample	Female	51	118

Attributions	Independent	On-manifold	ASV
Sex	-0.11	-0.16	-0.1
Age	-0.07	0.23	-0.08
Systolic BP	-0.05	-0.22	0.0
Poverty index	-0.03	0.09	-0.02
Blood pressure	0.0	0.0	-0.08
TIBC	0.0	-0.16	0.0
Diastolic BP	-0.02	-0.08	0.0
Pulse pressure	-0.01	-0.11	0.0
Serum Iron	0.01	0.07	0.0
BMI	-0.0	-0.05	-0.0
White blood cells	-0.01	0.03	0.0
Serum Protein	-0.0	0.05	0.0
Serum Albumin	-0.0	-0.04	0.0
Inflamation	0.0	0.0	-0.02
Serum Cholesterol	-0.0	0.04	-0.0
Iron	0.0	0.0	0.02
Sedimentation rate	-0.01	-0.01	0.0
Race	-0.0	0.0	-0.01
TS	0.01	0.01	0.0
Serum Magnesium	-0.0	-0.01	-0.0
Blood protein	0.0	0.0	-0.01
Red blood cells	-0.0	0.01	-0.0



(a) Shapley Flow

Figure 17: Comparison among methods on 100 background samples from the nutrition dataset, showing top 10 features/edges.



(a) Age vs. output (b) Age vs. systolic blood pressure

Figure 18: Age appears to be highly risky in on-manifold SHAP because it steals credit from other variables.

(a) Red cut

(b) Green cut

Figure 19: Two cuts that represent two boundaries for the same causal graph.