ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230265

SpArX: Sparse Argumentative Explanations for Neural Networks

Hamed Ayoobi[a;*], **Nico Potyka**[b,a] **and Francesca Toni**[a]

[a]Department of Computing, Imperial College London, United Kingdom
[b]Cardiff University, United Kingdom
ORCiD ID: Hamed Ayoobi https://orcid.org/0000-0002-5418-6352

Abstract. Neural networks (NNs) have various applications in AI, but explaining their decisions remains challenging. Existing approaches often focus on explaining how changing individual inputs affects NNs' outputs. However, an explanation that is consistent with the input-output behaviour of an NN is not necessarily faithful to the actual mechanics thereof. In this paper, we exploit relationships between *multi-layer perceptrons* (MLPs) and *quantitative argumentation frameworks* (QAFs) to create argumentative explanations for the mechanics of MLPs. Our *SpArX* method first sparsifies the MLP while maintaining as much of the original structure as possible. It then translates the sparse MLP into an equivalent QAF to shed light on the underlying decision process of the MLP, producing *global* and/or *local explanations*. We demonstrate experimentally that SpArX can give more faithful explanations than existing approaches, while simultaneously providing deeper insights into the actual reasoning process of MLPs.

1 Introduction

The increasing use of black-box models like neural networks (NNs) in autonomous intelligent systems raises concerns about their fairness, reliability and safety. To address these concerns, the literature puts forward various explainable AI approaches to render NNs more transparent, including model-agnostic approaches [43, 33], and approaches tailored to the structure of NNs [14, 50]. However, they fail to capture the actual mechanics of the NNs and thus it is hard to evaluate how *faithful* these approaches are to the NNs [27, 44, 42].

Some works advocate the use of formal, interpretable approaches for explainability [35]. Specifically for NNs, recent work [48] proposes regularizing the training procedure of NNs so that they can be well approximated by interpretable decision trees. While this is an interesting direction, evaluating the faithfulness of the decision trees to the NN remains a challenge. Other recent work unearths formal relationships between NNs in the form of multi-layer perceptrons (MLPs) and symbolic reasoning with *quantitative argumentation frameworks (QAFs)* [39, 10, 9, 11, 8, 12] or weighted conditional knowledge bases [24]. The formal relationships indicate that these approaches may pave the way towards potentially more faithful explanations than approximate abstractions such as decision trees.

In this paper, we provide explanations for MLPs leveraging on their formal relationships with QAFs in [39]. Intuitively, QAFs represent arguments and relations of attack or support between them as a graph, where nodes represent arguments and edges relations. Various QAF formalisms have been studied over the years, e.g. by [18, 6, 16, 41, 5, 37, 36, 38]. As it turns out, every MLP corresponds to a QAF of a particular form under a particular semantics [39]. This formal relationship between MLPs and QAFs suggests that QAFs are well suited to create faithful explanations for MLPs. However, just reinterpreting an MLP as a QAF would not give us a comprehensible explanation because the QAF has the same size and density as the original MLP. In order to create faithful and comprehensible argumentative explanations, we propose a two-step method. We first *sparsify* the MLP, while maintaining as much of its mechanics as possible. Then, we translate the sparse MLP into a QAF. We call our method *SpArX* (*Sparse Argumentative eXplanations* for MLPs). In principle, any existing compression method for NNs can be used for sparsification (e.g. [49]). However, existing methods are not designed for maintaining the mechanics of NNs towards explainability. We thus make the following contributions:

- We propose a novel *clustering method* for summarizing neurons based on their output-similarity. The clustered neurons' parameters result from aggregating the original parameters so that their output is similar to the outputs of neurons that they summarize.
- We propose two families of *aggregation functions* for aggregating the neurons in a cluster: the first gives *global explanations* (explaining the MLP for all inputs) and the second gives *local explanations* (explaining the MLP for a target input).
- We conduct several experiments demonstrating the viability of our SpArX method for MLPs and its competitiveness with respect to other methods in terms of (i) conventional notions of *input-output faithfulness* of explanations and (ii) novel notions of *structural faithfulness*, while (iii) shedding some light on the tradeoff between faithfulness and comprehensibility understood in terms of a notion of *cognitive complexity*, important towards human usability of explanations with SpArX. The code is publicly available[1].

Overall, we show that formal relationships between black-box machine learning (with NNs) and interpretable symbolic reasoning in QAFs can provide faithful and comprehensible explanations.

2 Related Work

While MLPs are most commonly used in their fully connected form, there has been increasing interest in learning sparse NNs in recent

[*] Corresponding Author. Email: h.ayoobi@imperial.ac.uk

[1] https://github.com/H-Ayoobi/SpArX

years. However, the focus is usually not on finding an easily interpretable network structure, but rather on decreasing the risk for overfitting, memory and runtime complexity and the associated power consumption. Existing approaches include regularization to encourage neurons with weight 0 to be deleted [32], pruning of edges [49], compression [31] and low rank approximation [47]. Interval NNs [40] summarize neurons in clusters based on their parameters and consider interval outputs for the clustered neurons to give lower and upper bounds on the outputs. We also summarize neurons in clusters, but cluster neurons based on their output and return an aggregated output instead of an interval for cluster neurons.

Several approaches exist for obtaining argumentative explanations for a variety of models [21]. Some approaches use argumentation to explain models directly [10], others use argumentative counterparts of the models. Some of them focus on NN explanations [1, 46], but they are based on approximations of NNs (e.g. using Layerwise Relevance Propagation [14]), rather than summarization as in our method, and their faithfulness is difficult to ascertain.

Several existing methods make use of symbolic reasoning for providing explanations [35]. The explanations resulting from these methods (e.g. abduction-based explanations [28], prime implicants [45], sufficient reasons [22], and majority reasons [7]) faithfully capture the input-output behaviour of the explained models rather than their mechanics. Other methods extract logical rules as explanations for machine learning models, including NNs [26, 23], but again focus on explanations that are only input-output faithful.

3 Preliminaries

Intuitively, a multi-layer perceptron (MLP) is a layered acyclic graph that processes its input by propagating it through the layers. Formally, we describe MLPs as follows.

Definition 1 (Multi-Layer Perceptron (MLP))**.** An *MLP* $\mathcal{M}$ is a tuple $(V, E, \mathcal{B}, \mathcal{W}, \varphi)$. (V, E) is a directed graph. $V = \biguplus_{l=0}^{d+1} V_l$ consists of (ordered) layers of neurons; for $0 \leq l \leq d+1$, $V_l = \{v_{l,i} \mid 1 \leq i \leq |V_l|\}$: we call V_0 the *input layer*, V_{d+1} the *output layer* and V_l, for $1 \leq l \leq d$, the l-th *hidden layer*; d is the *depth* of the MLP. $E \subseteq \bigcup_{l=0}^{d} \big(V_l \times V_{l+1}\big)$ is a set of edges between adjacent layers; if $E = \bigcup_{l=0}^{d} \big(V_l \times V_{l+1}\big)$, then the MLP is called *fully connected*. $\mathcal{B} = \{b^1, \ldots, b^{d+1}\}$ is a set of *bias* vectors, where, for $1 \leq l \leq d+1$, $b^l \in \mathbb{R}^{|V_l|}$. $\mathcal{W} = \{W^0, \ldots, W^d\}$ is a set of *weight* matrices, where, for $1 \leq l \leq d$, $W^l \in \mathbb{R}^{|V_{l+1}| \times |V_l|}$ such that $W^l_{i,j} = 0$ when $(v_{l,j}, v_{l+1,i}) \notin E$. $\varphi : \mathbb{R} \to \mathbb{R}$ is an *activation function*.

An example of MLP is given later in Fig. 1a. In order to process an *input* $x \in \mathbb{R}^{|V_0|}$, the input layer of $\mathcal{M}$ is initialized with x. The input is then propagated forward through $\mathcal{M}$ to generate values at each subsequent layer and finally an *output* in the output layer. Formally, if the values at layer l are $x_l \in \mathbb{R}^{|V_l|}$, then the values $x_{l+1} \in \mathbb{R}^{|V_{l+1}|}$ at the next layer are given by $x_{l+1} = \varphi(W^l x_l + b^l)$, with the activation function φ applied component-wise. We let $\mathcal{O}^{\mathcal{M}}_x : V \to \mathbb{R}$ denote the *output function* of $\mathcal{M}$, assigning to each neuron its value when the input x is given. That is, for $v_{0,i} \in V_0$, we let $\mathcal{O}^{\mathcal{M}}_x(v_{0,i}) = x_i$ and, for $l > 0$, we let the *activation value* of neuron $v_{l,i}$ be $\mathcal{O}^{\mathcal{M}}_x(v_{l,i}) = \varphi(W^l\, \mathcal{O}^{\mathcal{M}}_x(V_{l-1}) + b^l)_i$, where $\mathcal{O}^{\mathcal{M}}_x(V_{l-1})$ denotes the vector obtained from V_{l-1} by applying $\mathcal{O}^{\mathcal{M}}_x$ component-wise.

Every MLP can be seen as a quantitative argumentation framework (QAF) [39]. Intuitively, QAFs are *edge-weighted* directed graphs, where nodes represent *arguments* and, similarly to [36], edges with negative weight represent *attack* and edges with positive weight represent *support* relations between arguments. Each argument is initialized with a *base score* that assigns an apriori *strength* to the argument. The strength of arguments is then updated iteratively based on the strength values of attackers and supporters until the values converge. In acyclic graphs corresponding to MLPs, this iterative process is equivalent to the forward propagation process in the MLPs [39]. Conceptually, strength values are from some *domain* $\mathcal{D}$ [15]. As we focus on (real-valued) MLPs, we will assume $\mathcal{D} \subseteq \mathbb{R}$. The exact domain depends on the activation function, e.g. the logistic function results in $\mathcal{D} = [0, 1]$, the hyperbolic tangent in $\mathcal{D} = [-1, 1]$ and ReLU in $\mathcal{D} = [0, \infty]$. Formally, we describe QAFs as follows.

Definition 2 (Quantitative Argumentation Framework (QAF))**.** A *QAF with domain* $\mathcal{D} \subseteq \mathbb{R}$ is a tuple $(\mathcal{A}, E, \beta, w)$ that consists of

- sets $\mathcal{A}$ of *arguments* and $E \subseteq \mathcal{A} \times \mathcal{A}$ of *edges* between arguments;
- a function $\beta : \mathcal{A} \to \mathcal{D}$ assigning *base scores* in $\mathcal{D}$ to all arguments;
- a function $w : E \to \mathbb{R}$ assigning *weights* in $\mathbb{R}$ to all edges.

Edges with negative/positive weights are called *attack*/*support* edges, denoted by Att/Sup, respectively.

The strength values of arguments are usually computed iteratively using a two-step update procedure [36]: first, an *aggregation function* α aggregates the strength values of attackers and supporters; then, an *influence function* ι adapts the base score. Examples of aggregation functions include product [16, 41], addition [3, 37] and maximum [36], with the influence function defined accordingly to guarantee that strength values fall in $\mathcal{D}$. Here, we focus on the aggregation and influence functions from [39], to obtain QAFs simulating MLPs with a logistic activation function [39]. The strength values of arguments are computed by the following iterative procedure: for every $a \in \mathcal{A}$, we let $s_a^{(0)} := \beta(a)$ be the initial strength value; the strength values are then updated by the next two steps repeatedly (where the auxiliary α_a^i carries the aggregate at iteration $i \geq 0$):

Aggregation: $\alpha_a^{(i+1)} := \sum_{(b,a)\in E} w((b,a)) \cdot s_b^{(i)}$.

Influence: $s_a^{(i+1)} := \varphi_l\big(\ln(\frac{\beta(a)}{1-\beta(a)}) + \alpha_a^{(i+1)}\big)$, where $\varphi_l(z) = \frac{1}{1+\exp(-z)}$ is the logistic function.

The *final strength* of argument a is defined via the limit of $s_a^{(i)}$, for i towards infinity. Notably, the semantics given by this notion of final strength satisfies almost all desiderata for QAF semantics [39].

4 From General MLPs to QAFs

Here we generalize the connection between MLPs and QAFs beyond MLPs with logistic activation functions, as follows. Assume that $\varphi : \mathbb{R} \to \mathcal{D}$ is an activation function that is strictly monotonically increasing. Examples include logistic, hyperbolic tangent and parametric ReLU activation functions. Then φ is invertible and $\varphi^{-1} : \mathcal{D} \to \mathbb{R}$ is defined. We can then define the update function for an MLP with such activation function φ by using the same aggregation function as before and using the following influence function:

Influence: $s_a^{(i+1)} := \varphi\big(\varphi^{-1}(\beta(a)) + \alpha_a^{(i+1)}\big)$.

Note that the previous definition of influence in Section 3, from [39], is a special case because $\ln(\frac{1}{1-x})$ is the inverse function of the logistic function $\varphi_l(x)$. Note also that the popular ReLU activation function $\varphi_{ReLU}(x) = \max(0, x)$ is not invertible because all non-positive numbers are mapped to 0. However, for our purpose of translating MLPs to QAFs, we can define

$$\varphi^{-1}_{ReLU}(x) = \begin{cases} x, & \text{if } x > 0; \\ 0, & \text{otherwise.} \end{cases}$$

In order to translate an MLP $\mathcal{M}$ with activation function φ and input x into a QAF $Q_{\mathcal{M},x}$, we interpret every neuron $v_{l,i}$ as an abstract argument $A_{l,i}$. Edges in $\mathcal{M}$ with positive/negative weights are interpreted as supports/attacks, respectively, in $Q_{\mathcal{M},x}$. The base score of an argument $A_{0,i}$ associated with input neuron $v_{0,i}$ is just the corresponding input value x_i. The base score of the remaining arguments $A_{l,i}$ is $\varphi(b_i^l)$, where b_i^l is the bias of the associated neuron $v_{l,i}$.

Proposition 1. *Let $\mathcal{M}$ be an MLP with an invertible activation function φ or ReLU. Then, for every input x, the QAF $Q_{\mathcal{M},x}$ satisfies $\mathcal{O}_x^{\mathcal{M}}(v_{l,i}) = \sigma(A_{l,i})$, where $\sigma(A_{l,i})$ denotes the final strength of $A_{l,i}$ in $Q_{\mathcal{M},x}$.* (See the Supplementary Material (SM) in [13] for a proof).

5 SpArX: Explaining MLPs with QAFs

Just translating an MLP into a QAF may not give a comprehensible explanation because the QAF has the same size and density as the original MLP. Thus, we first sparsify the MLP and then translate it into a QAF. The sparsification should maintain as much of the original MLP as possible to give faithful explanations. To achieve this, we exploit redundancies in the MLP by replacing neurons giving similar outputs with a single neuron that summarizes their joint effect.

Summarizing neurons in this way is a clustering problem. Formally, a clustering problem is defined by a set of inputs from an abstract space $\mathcal{S}$ and a distance measure $\delta : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$. The goal is to partition $\mathcal{S}$ into clusters $C_1, \ldots, C_K$ (where $\mathcal{S} = \uplus_{i=1}^{K} C_i$) such that the distance between points within a cluster is 'small' and the distance between points in different clusters is 'large'. Finding an optimal clustering is NP-complete in many cases [25]. Thus, we cannot expect to find an efficient algorithm that computes an optimal clustering, but we can apply standard algorithms e.g. K-means [34] to find a good (but not necessarily optimal) clustering efficiently.

In our setting, $\mathcal{S}$ is the set V_l of neurons in layer $0 < l < d{+}1$ and the distance between neurons can be defined as the difference between their outputs for inputs in a given dataset Δ (e.g. the training dataset):

$$\delta(v_{l,i}, v_{l,j}) = \sqrt{\sum_{x \in \Delta} (\mathcal{O}_x^{\mathcal{M}}(v_{l,i}) - \mathcal{O}_x^{\mathcal{M}}(v_{l,j}))^2}. \quad (1)$$

After clustering, we have a partitioning $\mathcal{P} = \uplus_{l=1}^{d} \mathcal{P}_l$ of (the hidden layers of) our MLP $\mathcal{M}$, where $\mathcal{P}_l = \{C_1^l, \ldots, C_{K_l}^l\}$ is the clustering of the l-th layer, that is, $V_l = \uplus_{i=1}^{K_l} C_i^l$. We use the clustering to create a corresponding *clustered MLP* μ whose neurons correspond to clusters in the original MLP $\mathcal{M}$. We call these neurons *cluster-neurons* and denote them by v_C, where C is the associated cluster. Then:

Definition 3 (Graphical Structure of Clustered MLP)**.** Given MLP $\mathcal{M}$ and clustering $\mathcal{P} = \uplus_{l=1}^{d} \mathcal{P}_l$ of $\mathcal{M}$, the *graphical structure of the corresponding clustered MLP* μ is a directed graph (V^μ, E^μ) with

- $V^\mu = \uplus_{l=0}^{d+1} V_l^\mu$ consists of (ordered) layers of cluster-neurons such that:
  1. the input layer V_0^μ consists of a singleton cluster-neuron $v_{\{v_{0,i}\}}$ for every input neuron $v_{0,i} \in V_0$;
  2. the l-th hidden layer of μ (for $0 < l < d+1$) consists of one cluster-neuron v_C for every cluster $C \in \mathcal{P}_l$;
  3. the output layer V_{d+1}^μ consists of a singleton cluster-neuron $v_{\{v_{d+1,i}\}}$ for every output neuron $v_{d+1,i} \in V_{d+1}$;
- $E^\mu = \bigcup_{l=0}^{d} \left(V_l^\mu \times V_{l+1}^\mu\right)$.

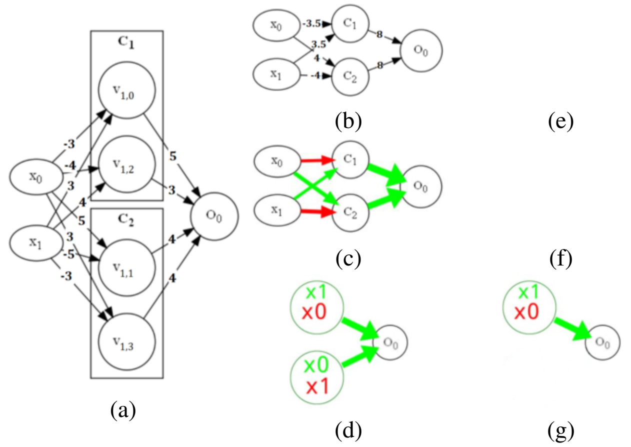


Figure 1: a) MLP for XOR, with cluster-neurons C_1, C_2. b) Clustered MLP for global explanation. c) Global explanation as a QAF. d) Word-cloud representation for the global explanation. e) Clustered MLP for the local explanation for $x_0{=}0, x_1{=}1$. f) Local explanation as a QAF. g) Word-cloud representation for the local explanation.

Example 1. Consider the MLP in Fig. 1.a, trained to approximate the XOR function from the dataset $\Delta = \{(0,0),(0,1),(1,0),(1,1)\}$ with target outputs $0,1,1,0$, respectively. The activation values of the hidden neurons for the four inputs are $(0,0,0,0)$, $(1.7,0,1.8,0)$, $(0,2.3,0,1.5)$, $(0,0,0,0)$, respectively. Applying K-means with δ as in Eq. 1 and $K = 2$ for the hidden layer results in clusters C_1, C_2 (indicated by rectangles in the figure). Fig. 1.b shows the graphical structure of the corresponding clustered MLP.

We define *global explanations* (for all inputs) and *local explanations* (for specific inputs) for MLPs by translating their corresponding clustered MLPs into QAFs, leveraging on the formal correspondence in Prop. 1. By doing so, we see QAFs themselves, equipped with the 'final strength' semantics from Section 4, as explanations. Using the terminology of [21], thus, our approach is in the spirit of *post-hoc approximate* approaches for argumentative explainable AI. Below and in Section 8 we will discuss how QAFs can be tailored to human users to support varied explanatory experiences.

The clustered MLPs for global and local explanations share the same graphical structure but differ in the parameters of the cluster-neurons, that is, (i) the biases of cluster-neurons and (ii) the weights for edges between cluster-neurons. We define these parameters in terms of *aggregation functions*, specifically a *bias aggregation function* $\text{Agg}^b : \mathcal{P} \rightarrow \mathbb{R}$, mapping clusters to biases, and an *edge aggregation function* $\text{Agg}^e : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R} \cup \{\perp\}$, mapping pairs of clusters to weights if the pairs correspond to edges in μ, or $\perp$ otherwise. Given any concrete such aggregation functions (as defined later), the parameters of μ can be defined as follows.

Definition 4 (Parameters of Clustered MLP)**.** Given an MLP $\mathcal{M}$, let (V^μ, E^μ) be the graphical structure of the corresponding clustered MLP μ. Then, for bias and edge aggregation functions Agg^b and Agg^e, respectively, μ is $(V^\mu, E^\mu, \mathcal{B}^\mu, \mathcal{W}^\mu, \varphi)$ with *parameters* $\mathcal{B}^\mu, \mathcal{W}^\mu$ as follows:

- for every cluster-neuron $v_C \in V^\mu$, the bias (in $\mathcal{B}^\mu$) of v_C is $\text{Agg}^b(C)$;
- for every edge $(v_{C_1}, v_{C_2}) \in E^\mu$, the weight (in $\mathcal{W}^\mu$) of the edge is $\text{Agg}^e((C_1, C_2))$.

5.1 Sparse Argumentative Global Explanations

We use the following aggregation functions, which minimize the deviation (with respect to the least-squares error) of bias and weights of cluster-neurons and the neurons they contain (as we explain in the SM in [13]).

Definition 5 (Global Aggregation Functions)**.** The *average bias and edge aggregation functions* are, respectively:

$$\mathrm{Agg}^b(C) = \frac{1}{|C|} \sum_{v_{l,i} \in C} b_i^l;$$

$$\mathrm{Agg}^e((C_1, C_2)) = \sum_{v_{l,i} \in C_1} \frac{1}{|C_2|} \sum_{v_{l+1,j} \in C_2} W_{j,i}^l.$$

The former simply averages the biases of neurons in the cluster. For the latter, intuitively, the weight of an edge between cluster-neurons v_{C_1} and v_{C_2} has to capture the effects of all neurons summarized in C_1 on neurons summarized in C_2. Every neuron in $C1$ is connected to all neurons in $C2$, thus the aggregated weight between them encapsulates all the weights between $C1$ and $C2$. As v_{C_2} acts as a replacement of all neurons in C_2, it has to aggregate their activation. We achieve this aggregation by averaging again.

The following example illustrates global explanations drawn from clustered MLPs via their understanding as QAFs.

Example 2. The QAF corresponding to the clustered MLP from Fig. 1.b can be visualised as in Fig. 1.c, where attacks are in red, supports in green, and the thickness of the edges reflects their weight. The same QAF can be visualized in different ways, e.g. to emphasize the role of each cluster-neuron. For instance, we can use a word-cloud representation as in Fig. 1.d (showing, e.g., that x_0 and x_1 play a negative and positive role, respectively, towards C_1, which supports the output). This word-cloud representation gives full insights into the reasoning of the MLP (with the learned rule $(\overline{x_0} \wedge x_1) \vee (x_0 \wedge \overline{x_1})$).

In general, word-cloud representations of cluster-neurons can be systematically generated by associating each cluster with a set of "most relevant" features, as follows. Every input neuron is associated with the corresponding feature; cluster-neurons in the first hidden layer are associated with the set of "most relevant" features from the input layer; cluster-neurons in the second layer are associated with the "most relevant" sets of sets of features that correspond to the most "relevant features" from the previous layer; and so on. We can measure "relevance" by the magnitude of its influence, which is the absolute value of edge weights for global explanations. For each word-cloud, the k most relevant features can be selected. As in Fig. 1.d, these features can be shown in green (red) if their influence is positive (negative, respectively). Also, the font size of features in word-clouds can be proportional to the magnitude of their influence.

5.2 Sparse Argumentative Local Explanations

While global explanations attempt to faithfully explain the behaviour of the MLP on all inputs, our local explanations focus on the behaviour in the neighborhood of the input x from the dataset Δ, similarly to LIME [43]. To do so, we generate random neighbors of x to obtain a *sample dataset* Δ', and weigh them with an exponential kernel from LIME [43], assigning lower weight to a sample $x' \in \Delta'$ that is further away from the target x:

$$\pi_{x',x} = exp(-D(x',x)^2/\sigma^2)$$

with D the Euclidean distance, σ the width of the exponential kernel.

We aggregate biases as before but replace the edge aggregation function with the following.

Definition 6 (Local Edge Aggregation Function)**.** The *local edge aggregation function* with respect to the *input* x is

$$\mathrm{Agg}_x^e(C_1, C_2) = \sum_{x' \in \Delta'} \pi_{x',x} \sum_{v_{l,i} \in C_1} \frac{1}{|C_2|.\mathcal{O}_{x'}^{\mu}(v_{C_1})} \sum_{v_{l+1,j} \in C_2} W_{i,j}^l \mathcal{O}_{x'}^{\mathcal{M}}(v_{l,i})$$

where $\mathcal{O}_{x'}^{\mathcal{M}}(v_{l,i})$ is the activation value of the neuron $v_{l,i}$ in the original MLP and $\mathcal{O}_{x'}^{\mu}(v_{C_1})$ is the activation value of the cluster-neuron C_1 in the clustered MLP.

Note that, by this definition, the edge weights are computed layer by layer from input to output.

Example 3. Fig. 1.e shows the clustered MLP for the local explanation of the XOR example (Fig. 1.a) where $x_0 = 0$ and $x_1 = 1$. Fig. 1.f shows the local explanation as a QAF. The word-cloud representation is also shown in Fig. 1.g. In this example, and in general for word-clouds for local explanations, we can measure "relevance" by the absolute value of edge weight times activation.

6 Desirable Properties of Explanations

To evaluate SpArX, we propose three measures for assessing faithfulness and comprehensibility of explanations. In this section, we assume as given an MLP $\mathcal{M}$ of depth d and a corresponding clustered MLP μ.

To begin with, we consider a *faithfulness* measure inspired by the notion of fidelity considered for LIME [43], based on measuring the *input-output* difference between the original model (in our case, $\mathcal{M}$) and the substitute model (in our case, the clustered MLP/QAF).

Definition 7 (Input-Output Unfaithfulness)**.** The *local input-output unfaithfulness* of μ to $\mathcal{M}$ with respect to *input* x and *dataset* Δ is

$$\mathcal{L}^{\mathcal{M}}(\mu) = \sum_{x' \in \Delta} \pi_{x',x} \sum_{v \in V_{d+1}} (\mathcal{O}_{x'}^{\mathcal{M}}(v) - \mathcal{O}_{x'}^{\mu}(v))^2.$$

The *global input-output unfaithfulness* of μ to $\mathcal{M}$ with respect to dataset Δ is

$$\mathcal{G}^{\mathcal{M}}(\mu) = \sum_{x' \in \Delta} \sum_{v \in V_{d+1}} (\mathcal{O}_{x'}^{\mathcal{M}}(v) - \mathcal{O}_{x'}^{\mu}(v))^2.$$

The lower the input-output unfaithfulness of the clustered MLP μ, the more faithful μ is to the original MLP.

The input-output unfaithfulness measures deviations in the input-output behaviour of the substitute model, but, since clustered MLPs maintain much of the structure of the original MLPs, we can define a more fine-grained notion of *structured faithfulness* by comparing the outputs of the individual neurons in the MLP with the outputs of the cluster-neurons summarizing them in the clustered MLP.

Definition 8 (Structural Unfaithfulness)**.** Let K_l be the number of clusters at hidden layer l in μ ($0<l\leq d$) and K_{d+1} be the number of output neurons. Let $K_{l,j}$ be the number of neurons in the jth cluster-neuron $C_{l,j}$ ($0 < l \leq d+1$, with $K_{d+1,j} = 1$). The *local structural unfaithfulness* of μ to $\mathcal{M}$ with respect to *input* x and *dataset* Δ is:

$$\mathcal{L}_s^{\mathcal{M}}(\mu) = \sum_{x' \in \Delta} \pi_{x',x} \sum_{l=1}^{d+1} \sum_{j=1}^{K_l} \sum_{v_{l,i} \in C_{l,j}} (\mathcal{O}_{x'}^{\mathcal{M}}(v_{l,i}) - \mathcal{O}_{x'}^{\mu}(C_{l,j}))^2.$$

The *global structural unfaithfulness* $\mathcal{G}_s^{\mathcal{M}}(\mu)$ is defined analogously by removing the similarity terms $\pi_{x',x}$.[2]

The lower the structured unfaithfulness of the clustered MLP μ, the more structurally faithful μ is to the original MLP. Note that our notion of structural faithfulness is different from the notions of structural descriptive accuracy by [2]: they characterise bespoke explanations, defined therein, of probabilistic classifiers equipped with graphical structures and cannot be used in place of our notion, tailored to local and global explanations with SpArX.

Finally, we consider the *cognitive complexity* of explanations based on their size, inspired by the cognitive tractability notion in [20]. We use the number of cluster-neurons/arguments as a measure.

Definition 9 (Cognitive Complexity)**.** Let K_l be the number of clusters at hidden layer l in μ ($0 < l \leq d$). Then, the *cognitive complexity* of μ is defined as

$$\Omega(\mu) = \prod_{0<l<d+1} K_l.$$

Note that there is a tradeoff between faithfulness and cognitive complexity. By reducing the number of cluster-neurons, we reduce cognitive complexity. However, this also results in higher variance in the neurons summarized in the clusters, so the faithfulness of the explanation may suffer. We will explore this trade-off in Section 8.

Finally, note that other properties of explanations by symbolic approaches, notably by [4], are unsuitable for our mechanistic explanations as QAFs. Indeed, these existing properties focus on the input-output behaviour of classifiers, rather than their mechanics.

7 Experiments

We conducted four sets of experiments to evaluate SpArX with respect to (i) the trade-off between its sparsification and its ability to maintain faithfulness (Section 7.1 for global and Section 7.2 for local explanations), and (ii) SpArX's scalability (Section 7.3). We used four datasets for classification: `iris`[3] with 150 instances, 4 continuous features and 3 classes; `breast cancer`[4] with 569 instances, 30 continuous features and 2 classes; `COMPAS` [29] with 11,000 instances and 52 categorical features, to classify *two_year_recid*; and `forest covertype`[5] [17] with 581,012 instances, 54 features (10 continuous, 44 binary), and 7 classes.

The first three datasets are standard benchmarks in the literature, from three different domains (biology/medicine/law). We used these datasets to evaluate the (input-output and structural) unfaithfulness of the global and local explanations generated by SpArX. These datasets however only require small MLPs (see the SM in [13]). We then used the last dataset, which is another standard benchmark but requires deeper MLPs with more hidden neurons (see the SM in [13]), to evaluate the scalability of SpArX.

For the experiments with the first three datasets, we used MLPs with 2 hidden layers and 50 hidden neurons each, whereas for the experiments with the last dataset we used 1-5 hidden layers with 100, 200 or 500 neurons. For all experiments, we used the RELU activation function for the hidden neurons and softmax for the output neurons. We give classification performances for all MLPs and average run-times for generating local explanations in the SM in [13].

When experimenting with SpArX, one needs to choose the number of clusters/cluster-neurons at each hidden layer: we do so by specifying a *compression ratio* (for example, a compression ratio of 0.5 amounts to obtaining half cluster-neurons than the original neurons).

7.1 Global Faithfulness (Comparison to HAP)

Since SpArX essentially compresses an MLP to construct a clustered MLP/QAF, one may ask how it compares to existing compression approaches.[6] To assess the faithfulness of our global explanations, we compared SpArX's clustering approach to the state-of-the-art compression method Hessian Aware Pruning (HAP) [49], which uses relative Hessian traces to prune insensitive parameters in NNs. We measured both input-output and structural unfaithfulness of SpArX and HAP to the original MLP, using the result of HAP compression in place of μ when applying Definitions 7 and 8 for comparison.

Input-Output Faithfulness. Table 1 shows the input-output unfaithfulness of global explanations ($\mathcal{G}^{\mathcal{M}}(\mu)$ in Definition 7) obtained from SpArX and HAP using the three chosen datasets and different compression ratios. The unfaithfulness of global explanations in SpArX is lower than HAP, especially for high compression ratios. Note that this does not mean that SpArX is a better compression method, but that the compression method in SpArX is better for our purposes (i.e., compressing the MLP while keeping its mechanics).

Structural Faithfulness. Table 2 gives the structural global unfaithfulness ($\mathcal{G}_s^{\mathcal{M}}(\mu)$ in Definition 8) for SpArX and HAP, on the three chosen datasets, using different compression ratios. Our method has a much lower structural unfaithfulness than HAP by preserving activation values close to the original model.

Method	Compression	Datasets		
	Ratio	Iris	Cancer	COMPAS
HAP SpArX	0.2	0.05 **0.00**	0.48 **0.02**	0.02 **0.00**
HAP SpArX	0.4	0.23 **0.00**	0.53 **0.05**	0.11 **0.00**
HAP SpArX	0.6	0.23 **0.00**	0.58 **0.10**	0.20 **0.00**
HAP SpArX	0.8	0.28 **0.00**	1.00 **0.21**	0.26 **0.00**

Table 1: Global input-output unfaithfulness of sparse MLPs generated by HAP vs our SpArX method. (Best results in **bold**)

7.2 Local Faithfulness (Comparison to LIME)

In order to evaluate the local input-output unfaithfulness of SpArX ($\mathcal{L}^{\mathcal{M}}(\mu)$ in Definition 7), we compared SpArX and LIME [43][7], which approximates a target point locally by an interpretable substitute model.[8] Table 3 shows the input-output unfaithfulness of the local explanations for LIME and SpArX. We used the same sampling approach as LIME [43]. We averaged the unfaithfulness measure for

[2] See the SM in [13] for the formal definition.

[3] https://archive.ics.uci.edu/ml/datasets/iris

[4] https://archive.ics.uci.edu/ml/datasets/cancer

[5] https://archive.ics.uci.edu/ml/datasets/covertype

[6] Whereas existing NN compression methods typically retrain after compression, we do not, as we want to explain the original NN.

[7] https://github.com/marcotcr/lime

[8] We used ridge regression, suitable with tabular data in LIME. We used the substitute model as μ when applying Definition 7 to LIME.

Method	Compression Ratio	Datasets		
		Iris	Cancer	COMPAS
HAP SpArX	0.2	0.23 **0.00**	9.54 **0.83**	0.10 **0.02**
HAP SpArX	0.4	0.89 **0.00**	61.57 **1.04**	0.24 **0.03**
HAP SpArX	0.6	1.37 **0.00**	61.57 **1.40**	0.46 **0.04**
HAP SpArX	0.8	3.00 **0.02**	116.20 **2.34**	1.20 **0.05**

Table 2: Global structural unfaithfulness of sparse MLPs generated by HAP vs our SpArX method.

all test examples. The results show that the local explanations produced by our approach are more input-output faithful to the original model. Thus, basing local explanations on keeping the MLP mechanics helps also with their input-output faithfulness.

Method	Datasets		
	Iris	Cancer	COMPAS
LIME	0.3212	0.1623	0.0224
SpArX (0.6)	**0.0257**	**0.0055**	**0.0071**
SpArX (0.8)	**0.0707**	**0.0156**	**0.0083**

Table 3: Local input-output unfaithfulness of LIME vs our SpArX method (with different compression ratios, in brackets).

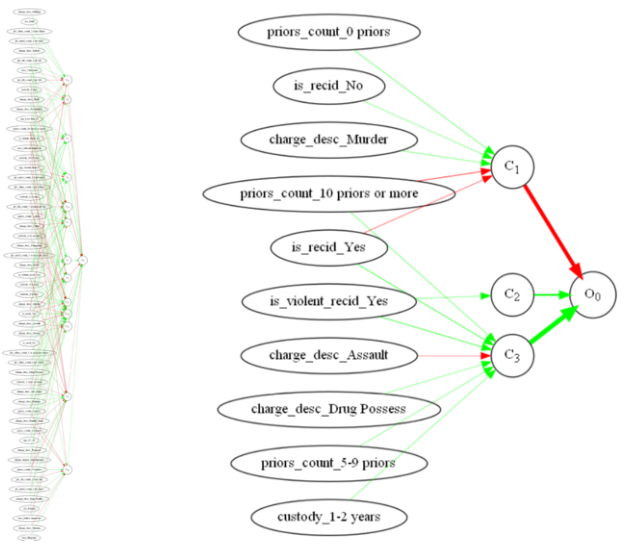
#Layers	Method	#Neurons		
		100	200	500
1	LIME	0.2375	0.2919	0.3123
1	SpArX	**0.0000**	**0.0018**	**0.0000**
2	LIME	0.2509	0.2961	0.3638
2	SpArX	**0.0019**	**0.0015**	**0.0034**
3	LIME	0.3130	0.3285	0.3127
3	SpArX	**0.0028**	**0.0026**	**0.0000**
4	LIME	0.3395	0.3459	0.3243
4	SpArX	**0.0001**	**0.0049**	**0.0000**
5	LIME	0.3665	0.3178	0.3288
5	SpArX	**0.0030**	**0.0064**	**0.0000**

Table 4: Evaluating scalability of SpArX (`forest covertype` dataset): local input-output unfaithfulness of SpArX (with 80% compression ratio) and LIME using various MLPs with different numbers of hidden layers (#Layers) and neurons (#Neurons).

7.3 Scalability

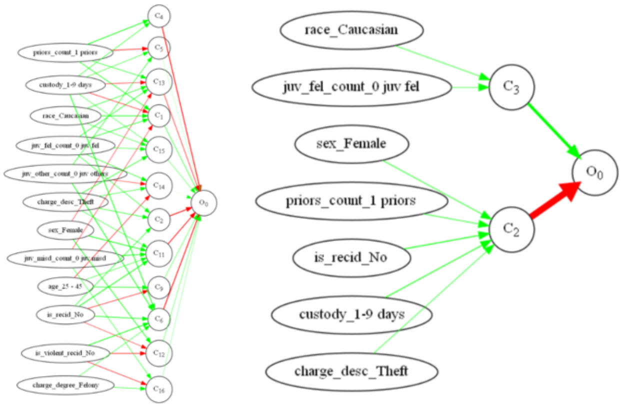
To evaluate the scalability of SpArX, we measured its input-output faithfulness on MLPs of increasing complexity, in comparison with LIME [43], using `forest covertype` as a sufficiently large dataset to be tested with various MLP architectures of different sizes. We have trained 15 MLPs with varying numbers of hidden layers (#Layers) and different numbers of hidden neurons (#Neurons) at each hidden layer (see details in the SM in [13]).

Table 4 compares the input-output unfaithfulness of the local explanations by SpArX using 80% compression ratio[9] with LIME, all averaged over the test set. The results confirm that SpArX explanations are scalable to different MLP architectures of different sizes.



(a) 20% compression ratio (b) 85% compression ratio

Figure 2: Global explanations by SpArX of an MLP with 20% and 85% compression ratios for `COMPAS`. Here O_0 concerns recommitting a crime after two years. (Sub-figure (a) is given to emphasize poor interpretability due to size, so readability is not a concern).



(a) 20% compression ratio (b) 85% compression ratio

Figure 3: Local explanations by SpArX of an MLP with 20% and 85% compression ratios for `COMPAS`. Here $O_0 = 0$ suggests that the individual will not be recommitting a crime after two years. (Again, the readability of sub-figure (a) is not a concern).

8 Towards Tailoring SpArX to Users

The explanations obtained with SpArX are in the form of QAFs drawn from the sparsified MLPs, in the spirit of much work in argumentative explainable AI [21]. In this section, we explore how they might be tailored to users. We first consider the property of cognitive complexity for SpArX (see Definition 9) and the tradeoff between faithfulness and cognitive complexity (Section 8.1) and then illustrate how local and global explanations can be the starting point to obtain more natural explanations (Section 8.2). Throughout this section, we focus on examples only, all drawn in the context of an MLP trained on the `COMPAS` dataset with one hidden layer and 20 neu-

[9] For experiments with lower compression ratios, see the SM in [13].

rons in the hidden layer (see details in the SM in [13]). We chose COMPAS because it is a very popular dataset in the literature, and it is the largest, amongst those we consider, for binary classification.

8.1 Cognitive Complexity

The cognitive complexity of SpArX depends on the number of clusters per layer. Fewer clusters lead to a more interpretable explanation at the cost of achieving lower (structural) faithfulness.

Fig. 2a shows a *global explanation* for the given MLP for COMPAS, with 20% compression ratio and pruning edges with low weights.[10] The classification results of the clustered MLP underpinning this explanation is 98.32%, the same as the original MLP. This explanation is clearly hard to interpret by a user. Fig. 2b shows the global explanation with 85% compression rate and, again, pruning the edges with low weights. This global explanation is more comprehensible (and the classification results are 94.60%, the same as the original model).

Using the same MLP for the COMPAS dataset, *local explanations* of an input example, with 20% and 85% compression ratios, are shown in Fig. 3 (both clustered MLPs compute the same output $O_0 = 0$, indicating that the input individual, with the features as given in the figure, is predicted to not re-offend within two years). Fig. 3b is more interpretable than Fig. 3a, but the two clustered MLPs make the same prediction for the given input, faithfully to the MLP.

8.2 From SpArX to Natural Explanations

Global and local explanations obtained by SpArX can be presented so that humans can progressively inquire about the reasoning of the underlying MLP, e.g. by instantiating templates as in [19, 20]. For illustration, consider the global explanation in Fig. 2b again. Unlike shallow input-output explanations, we can see the role of each hidden cluster-neuron in the proposed method. There are two sets of hidden cluster-neurons, namely an attacker (C_1) and two supporters (C_2 and C_3). They could be shown incrementally, following prompts from a human user, to explain the output. Specifically, C_1 attacks the output, indicating that the input individual will not recommit a crime in a two-year period. Three features are supporting C_1 and two features are attacking it, and they could also be shown to a human user on demand. The attacking features also support C_3. This means that they strengthen the support by C_3 and weaken the attack by C_1. Therefore, *priors_count_10 priors or more* and *is_recid_Yes* both strongly affect the output. Indeed, looking at the COMPAS dataset, more than 99% of individuals that have these two features recommitted the crime in a two years period. C_2 and C_3 are supporting the output. $C2$ is only supported by the *is_violent_recid_Yes* feature. This suggests that if individuals have a violent recidivism they are likely to recommit a crime after two years. In the COMPAS dataset, this conjecture is 100% valid. These kinds of interpretations go beyond the shallow input-output explanations offered e.g. by LIME, but can be automatically drawn from the QAFs generated by SpArX. $C3$ is supported by several features and is attacked by one feature. Looking at this argumentative global explanation one can understand the effect of each feature as well as of each hidden neuron on the output.

Similar considerations can be made for local explanations, e.g. the explanation in Fig. 3b. The class label for this input example is 0, which means that the individual has not recommitted the crime in a two-year period. The local explanation shows this fact by emphasising C_2 as a stronger attacker than the supporter C_3. C_2 says that since the sex of the individual is female, she has only 1 prior, she has no recidivism, she has custody time of 1 to 9 days and the crime was theft, she will not recommit the crime in a two years period.

For these readings of explanations to be natural, human-interpretable presentations of the cluster-neurons are needed. Specifically, we could use the word-cloud presentation from Section 5 (see Examples 2 and 3). For illustration, Fig. 4 shows the local explanation in Fig. 3b with word-clouds for presenting hidden cluster-neurons and output neurons.[11] Other ways to present cluster-neurons may be useful: we leave the exploration of alternatives as well as human studies to assess their amenability to humans to future work.

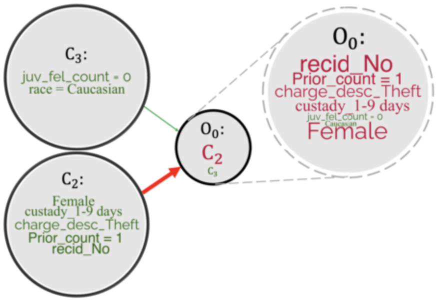


Figure 4: Word-cloud presentation of the cluster-neurons and output neuron for the local explanation in Fig. 3b. To enhance comprehensibility, we can display the word-cloud of the output node with respect to the input features instead of the clusters (see dashed lines).

9 Conclusion

We introduced SpArX, a novel method for generating argumentative explanations for MLPs. In contrast to shallow input-output explainers like LIME, SpArX maintains structural similarity to the original MLP in order to give faithful explanations, while allowing tailoring them to comprehensibility for users. Our experimental results show that the explanations by SpArX are more faithful to the original model than LIME. We have also compared SpArX with a state-of-the-art NN compression technique called HAP, showing that SpArX is more faithful to the original model. Further, our explanations are more *structurally* faithful to the original model by providing deeper insights into the mechanics thereof, and can be tailored to users for cognitive tractability and to obtain natural explanations.

Future research includes extending SpArX to other types of NNs, e.g. CNNs, as well as furthering it to cluster neurons across hidden layers. It would also be interesting to explore whether SpArX could be extended to exploit formal relationships between NNs and other symbolic approaches, e.g. in [24]. Further, it would be interesting to explore formalizations such as in [30] for characterizing uncertainty as captured by SpArX.

Acknowledgments

This research was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101020934, ADIX) and by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme. Any views or opinions expressed herein are solely those of the authors.

[10] Note that pruning is only done here for visualization.

[11] See the SM in [13] for the word-cloud variant of the global explanation in Fig. 2b.