Neural QBAFs: Explaining Neural Networks Under LRP-Based Argumentation Frameworks

Purin Sukpanichnant(✉), Antonio Rago, Piyawat Lertvittayakumjorn, and Francesca Toni

Imperial College London, London, UK
{ps1620,a.rago,pl1515,ft}@imperial.ac.uk

Abstract. In recent years, there have been many attempts to combine XAI with the field of symbolic AI in order to generate explanations for neural networks that are more interpretable and better align with human reasoning, with one prominent candidate for this synergy being the subfield of computational argumentation. One method is to represent neural networks with quantitative bipolar argumentation frameworks (QBAFs) equipped with a particular semantics. The resulting QBAF can then be viewed as an explanation for the associated neural network. In this paper, we explore a novel LRP-based semantics under a new QBAF variant, namely *neural QBAFs* (nQBAFs). Since an nQBAF of a neural network is typically large, the nQBAF must be simplified before being used as an explanation. Our empirical evaluation indicates that the manner of this simplification is all important for the quality of the resulting explanation.

Keywords: Neural networks · Computational argumentation · Image classification

1 Introduction

Several attempts have been made to improve explainability of AI systems. One prominent research area of XAI is devoted to explaining black-box methods such as deep learning. A popular method from this area is *Layer-wise Relevance Propagation* (LRP) [11]. This method determines how relevant nodes in a neural network are towards the neural network output. However, LRP does not explicitly indicate the relationship between each node. To address this issue, we combine this method with computational argumentation. This is a field of study about how knowledge can be represented as relationships between arguments. Each complete set of relationship(s) is referred to as an Argumentation Framework (AF) [6]. There are several types of AF, depending on types of relationships. In this paper, we consider a type of AF known as Quantitative Bipolar Argumentation Frameworks (QBAFs) [2], which is a form of knowledge representation displaying relationships between arguments in forms of support and

S. Bandini et al. (Eds.): AIxIA 2021, LNAI 13196, pp. 429–444, 2022.
https://doi.org/10.1007/978-3-031-08421-8_30

attack. These attacks and supports lend themselves well to represent negative and positive influences from input features as obtained using LRP.

QBAFs are interpreted by *semantics* which, in a nutshell, determine the arguments' dialectical strengths, taking into account (the dialectical strength of) their attackers and supporters.

As QBAFs illustrate how arguments relate to one another, they can be applied to reflect the relationship between nodes of a neural network, which can be viewed as an explanation. However, to do this, one needs to match the neural network functioning and the QBAF semantics. In this paper, we focus on LRP as a semantics for suitable forms of the QBAFs that we introduce. QBAFs derived by an LRP-based semantics may be very large and too complicated for human cognition in the context of explanation. Hence a new variant of QBAF is needed. To address this issue, we introduce a new variant of QBAF, namely neural QBAFs (nQBAFs), under LRP-based semantics for generating argumentative explanations from neural networks and prove their dialectical properties. Finally, we conduct some preliminary experiments by applying our LRP-based semantics to the Deep Argumentative Explanation (DAX) method from [1] and the method from [13] in order to show practical issues with nQBAFs as explanations. This is work in progress, on exploring the use of LRP, in combination with other techniques, in visualisation for image classification: we leave a comparison with visualisations drawn from nQBAFs as future work.

2 Background

We start by defining relevant concepts for our setting. These amount to multi-layer perceptrons (MLPs), Layer-wise Relevance Propagation (LRP) and Quantitative Bipolar Argumentation Frameworks (QBAFs).

2.1 MLP Basics

A MLP is a form of feed-forward neural network where all neurons in one layer are connected to all neurons in the next layer. We follow [14] for background on MLPs, captured by Definitions 1 and 2 below.

Definition 1. *A Multi-layer Perceptron (MLP) is a tuple* $\langle V, E, B, \theta \rangle$ *where*

- $\langle V, E \rangle$ *is an acyclic directed graph.*
- $V = \uplus_0^{d+1} V_i$ *is the disjoint union of sets of nodes* V_i*;*
- *We call* V_0 *the input layer,* V_{d+1} *the output layer and* V_i *the i-th hidden layer for* $1 \leq i \leq d$*;*
- $E \subseteq \bigcup_{i=0}^{d} (V_i \times V_{i+1})$ *is a set of edges between subsequent layers;*
- $B : (V \setminus V_0) \to \mathbb{R}$ *assigns a bias to every non-input node;*
- $\theta : E \to \mathbb{R}$ *assigns a weight to every edge.*

Figure 1 (left) visualises a fragment of an MLP with at least two hidden layers. Note that any MLP referred to afterwards only has one output node. This may be obtained by extracting all nodes and the edges between these nodes from another MLP that have paths[1] to the chosen output node, including the output node itself.

MLPs typically result from training with sample data. Since this training is not a focus of this paper, we will simply assume that a trained MLP is available. For example, in Sect. 5, we will conduct experiments with a pre-trained MLP for image classification.

The next definition explains how we obtain an activation value for each node.

Definition 2. *For any $j \in V_0$, the* activation $x_j \in \mathbb{R}$ *of node j is an input value for j. For any k such that $1 \leq k \leq d+1$, the* activation *of node $i \in V_k$ is $x_i = act(B(i) + \Sigma_{n \in V_{k-1}} x_n \theta(n,i))$ where act: $\mathbb{R} \to \mathbb{R}$ is an activation function.*[2]

Activations are a fundamental component of a neural network. They are involved in the calculation process of a neural network from a given input towards the output layer. An activation of each node can also be used to explain what the neural network is emphasising, as we discuss in the next section.

2.2 LRP Basics

Layer-Wise Relevance Propagation (LRP) [11] is a method for obtaining explanations, for outputs of MLPs in particular. Intuitively, with LRP, each node of the MLP is given a relevance score, showing how this node contributes to the node of interest in the output layer. Starting from the output layer, the node we want to explain has its relevance score equal to its activation while other nodes of the output layer (if any) have zero relevance score. Then we can calculate the relevance score for each non-output node using Definition 3, adapted from the presentation of LRP in [9].

Definition 3. *Let $\langle V, E, B, \theta \rangle$ be an MLP, and $i \in V_k$, and $j \in V_{k+1}$ where $0 \leq k \leq d$, and the layer k has n nodes. Then the* relevance score *the node i receives from the node j is $R_{i \leftarrow j}$ such that $R_{i \leftarrow j} = \frac{z_{ij}}{\Sigma_{l=1}^{n} z_{lj}} R_j$ where z_{ij} is the contribution from i to j during the forward pass, i.e., $z_{ij} = x_i \theta(i,j) + \frac{B(j)}{n} + \frac{\epsilon}{n}$ where $\epsilon \in \mathbb{R}$ is a small positive stabiliser.*

Note that this definition assumes that ϵ is distributed equally to the n nodes: we adopt this assumption from [9]. To calculate the relevance score node i has towards the output node of interest, i.e. R_i, we simply sum all the relevance scores it receives from all the nodes of the layer $k+1$. In other words, $R_i = \Sigma_j R_{i \leftarrow j}$.

From Definition 3, we obtain also that LRP has *conservative properties* (for $i \in V_k$, and $j \in V_{k+1}$), i.e., $R_j = \Sigma_i R_{i \leftarrow j}$ and $\Sigma_i R_i = \Sigma_j R_j$.

[1] The definition of *path* is adopted from [1], where there exists a *path* via E (set of edges) from n_a to n_b (from a node to another) iff $\exists n_1, ..., n_t$ with $n_1 = n_a$ and $n_t = n_b$ such that $(n_1, n_2), ..., (n_{t-1}, n_t) \in E$.

[2] Note that, with an abuse of notation, $\theta(n,i)$ stands for $\theta((n,i))$, for simplicity. Unless explicitly stated, this notation is used throughout the rest of the paper.

2.3 QBAF Basics

QBAFs [2] are abstractions of debates between arguments, where arguments may attack or support one another and are equipped with a base score, which reflects the arguments' intrinsic, initial dialectical strength. We adopt the formal definition of QBAFs from [2].

Definition 4. *A* QBAF *is a tuple* $\langle A, Att, Supp, \gamma \rangle$ *where*

- A *is a set (whose elements are referred to as* arguments*);*
- $Att \subseteq A \times A$ *is the* attack *relation;*
- $Supp \subseteq A \times A$ *is the* support *relation;*
- $\gamma : A \to D$ *is a function that maps every argument to its* base score *(from some set* D *of a given set of values).*[3]

A QBAF may be equipped with a notion of dialectical strength, given by a *strength function* $\sigma : A \to D$, indicating a dialectical strength value (again from D) for each argument, taking into account the strength of the attacking and supporting arguments within the debate represented by the QBAF, as well as the argument's intrinsic strength given by γ. Several notions of σ (called *semantics* in the literature on computational argumentation) have been given in the literature (e.g. see [3]) but their formal definitions are outside the scope of this paper. Various *dialectical properties* for semantics σ have been studied in the literature (e.g. see [3]) as a way to validate their use in concrete settings and to compare across different semantics. We will follow this approach in this paper.

Variants of QBAFs can be extracted from neural networks, e.g. as in [1,14]. An example of the structure underpinning these QBAFs is given in Fig. 1 (centre, for the MLP on the left): here, the nodes represent the arguments and the edges represent the union of the attack and support relations. In these works, the extracted QBAF can be seen as indicating how some nodes in the neural network relate to others, and hence can be viewed as an explanation of that neural network. We follow this approach in this paper, but using a variant of QBAFs, defined next.

3 nQBAFS and LRP-Based Argumentation Semantics

We study LRP as a semantics σ for novel forms of QBAFs extracted from MLPs. We aim to prove that this LRP-based semantics satisfies multiple dialectical properties, which we believe are intuitive when QBAFs are used as the basis for explanations of MLPs.

The novel QBAFs take into account the structure of MLPs. As of Definition 3, a non-output node in an MLP may contribute to several nodes of the next layer, as in Fig. 1 (left). For any non-output node i, if we consider each edge from i to a node of the next layer and represent the node i with a unique argument for every

[3] In this paper, we will choose $D = \mathbb{R}$.

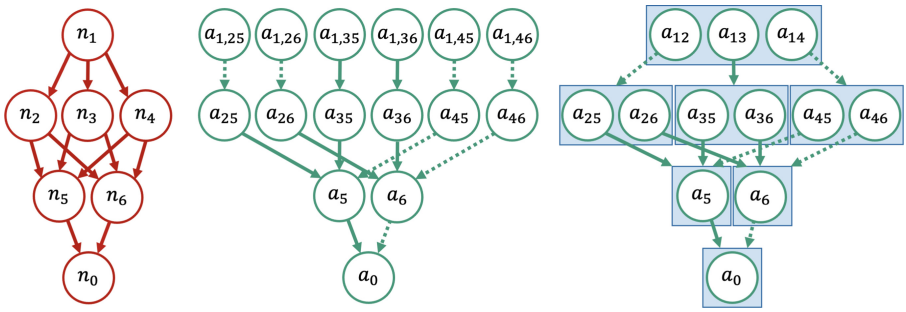


Fig. 1. Example of an MLP (left), a standard QBAF (centre) and the associated nQBAF (right). Each box refers to a group of arguments. In the QBAF and the nQBAF, dashed lines represent attacks and solid lines represent supports.

edge (as in [4,14]), there would be several arguments representing that node i. This method would also be non-scalable since the relation between arguments in the resulting QBAF would become too complex to analyse as more layers are considered. To avoid this, we define a new, leaner form of QBAFs, where arguments referring to the same node are grouped together.

Definition 5. *A* neural quantitative bipolar argumentation framework *(nQBAF) is a tuple* $\langle A, Att, Supp, \gamma \rangle$ *where*

- A *is a set (of* arguments*);*
- $Att \subseteq A \times \mathcal{P}(A)$[4] *is the* attack *relation;*
- $Supp \subseteq A \times \mathcal{P}(A)$ *is the* support *relation;*
- $\gamma : A \cup \mathcal{P}(A) \rightarrow \{0\}$ *is a function that maps every argument and set of arguments to a fixed base score of zero.*

Thus, attack and support relations may exist not just between arguments, as in standard QBAFs, but also between arguments and sets thereof. Given that we choose $D = \mathbb{R}$ as the set of values that could be used as base score and strength of arguments, the choice of γ indicates that each argument and set of arguments starts with a "neutral" base score of zero.

We first need to relate arguments of an nQBAF and nodes of a given MLP $\langle V, E, B, \theta \rangle$. Each argument represents only one node but a node can be represented by several arguments. Accordingly, we assume a function $\rho : A \cup \mathcal{P}(A) \rightarrow V \cup \{\bot\}$ mapping each argument/set of arguments to a node of the MLP, if one exists (or mapping to $\bot$ otherwise). We omit the formal definition of ρ for lack of space. As an illustration, for the MLP in Fig. 1 (left), in the derived nQBAF (right), $n_1 = \rho(\alpha_{12}) = \rho(\alpha_{13}) = \rho(\alpha_{14}) = \rho(\{\alpha_{12}, \alpha_{13}, \alpha_{14}\})$, $n_2 = \rho(\alpha_{25}) = \rho(\alpha_{26}) = \rho(\{\alpha_{25}, \alpha_{26}\})$, $n_3 = \rho(\alpha_{35}) = \rho(\alpha_{36}) = \rho(\{\alpha_{35}, \alpha_{36}\})$, $n_4 = \rho(\alpha_{45}) = \rho(\alpha_{46}) = \rho(\{\alpha_{45}, \alpha_{46}\})$, $n_5 = \rho(\alpha_5) = \rho(\{\alpha_5\})$, $n_6 = \rho(\alpha_6) = \rho(\{\alpha_6\})$, $n_0 = \rho(\alpha_0) = \rho(\{\alpha_0\})$ and, for any other set S of arguments, $\rho(S) = \bot$.

[4] Note that $\mathcal{P}(A)$ is the power set of a set A.

Algorithm 1: Extracting A from an MLP with the output node α_0

```
A ← {α_0};
currentLayer ← d;
while currentLayer >= 0 do
    for n_i in V_currentLayer do
        for n_j in V_currentLayer+1 do
            if (n_i, n_j) in E then
                A ← A ∪ {α_ij}
    currentLayer ← currentLayer − 1
for α_mn in A do
    if ρ(α_mn) in V_0 then
        A ← A ∪ {α_(mn)'mn}
```

We then have to determine which pairs (i.e. edges as shown in Fig. 1 (right)) belong to the attack or support relations. This is done using two *relation characterisations*, inspired by those in [1]: $c_+, c_- : A \times \mathcal{P}(A) \rightarrow \{true, false\}$ where, for any argument i and group of arguments j such that $\rho(i) \neq \bot$ and $\rho(j) \neq \bot$ are in adjacent layers (i.e. $(\rho(i), \rho(j)) \in E$):

- $c_+(i, j)$ is true iff $R_{\rho(i) \leftarrow \rho(j)} > 0$, and
- $c_-(i, j)$ is true iff $R_{\rho(i) \leftarrow \rho(j)} < 0$.

With c_+ and c_-, we can formally define our *Att* and *Supp* relations and the nQBAF derived from an MLP, as follows:

Definition 6. *The* nQBAF derived from $\langle V, E, B, \theta \rangle$ *is* $\langle A, Att, Supp, \gamma \rangle$ *where*

- A *is defined according to Algorithm 1;*
- $Att = \{(i, j) \in A \times \mathcal{P}(A) \mid c_-(i, j)$ *is true*$\}$*;*
- $Supp = \{(i, j) \in A \times \mathcal{P}(A) \mid c_+(i, j)$ *is true*$\}$*;*
- $\gamma : A \cup \mathcal{P}(A) \rightarrow \{0\}$.

Algorithm 1 extracts the set of arguments by iterating backwards from the last hidden layer to the input layer. It also add imaginary arguments to the set of arguments for input nodes, for the reason discussed in the next section.

Before we define our strength function , let us introduce some notation:

- $Att(x) = \{a \in A \mid (a, x) \in Att\ \}$ for all $x \in \mathcal{P}(A)$;
- $Supp(x) = \{s \in A \mid (s, x) \in Supp\ \}$ for all $x \in \mathcal{P}(A)$;
- $\mathcal{G} = \{g \in \mathcal{P}(A) \mid \exists a \in A[(a, g) \in Att \vee (a, g) \in Supp]\}$.

Now we define the LRP-based semantics for our nQBAF as follows:

Definition 7. *The* LRP-based semantics *of the nQBAF derived from an MLP* $\langle V, E, B, \theta \rangle$ *is* $\sigma : A \cup \mathcal{G} \rightarrow \mathbb{R}$ *such that*

$$\sigma(x) = \begin{cases} x_i & \text{if } \rho(x) \in V_{d+1} \text{ with final activation } x_i \\ R_{m \leftarrow \rho(y)} & \text{if } x = \alpha_{(mn)'mn}, z = \alpha_{mn} \text{ and } \exists!(z,y) \in Att \cup Supp \\ R_{\rho(x) \leftarrow \rho(y)} & \text{if } \exists!(x,y) \in Att \cup Supp \\ \Sigma_{a \in x} \sigma(a) & \text{if } x \in \mathcal{G} \\ 0 & \text{otherwise} \end{cases}$$

Now we are able to conceive the relations between arguments, and to what amount each argument supports or attacks a group of arguments, but how natural is it? Does it follow the way humans naturally debate? To answer these questions, we have to consider whether our nQBAFs satisfy dialectical properties.

4 Properties for nQBAFS Under LRP Semantics

We now consider dialectical properties that determine how natural the argumentation is for any argumentation framework, i.e. how similar it is to human reasoning and debate. Our dialectical properties, as shown in Table 1, are based on those in [1] and [2] but are adapted specifically for nQBAFs. In the table, we associate these properties with names, mostly borrowing from the literature, where, however, they have been used for other types of argumentation frameworks.

Before defining the properties, we first make an addition regarding the input layer. Every dialectical property which follows considers the strength of a group of arguments based on its attackers and supporters. As of now, there are no attackers or supporters for groups of arguments representing nodes of the input layer, so it is likely most properties will not be satisfied here. To resolve this issue, we add imaginary arguments to target the input nodes. These added arguments are not considered as part of the set of all groups of arguments $\mathcal{G}$. Formally, for any $g \in \mathcal{G}$ such that $\rho(g) \in V_0$, $Att(g) = \{x \in A \mid \rho(x) = \bot \wedge \exists a \in g[\sigma(x) = \sigma(a) \wedge \sigma(x) < 0]\}$ and $Supp(g) = \{x \in A \mid \rho(x) = \bot \wedge \exists a \in g[\sigma(x) = \sigma(a) \wedge \sigma(x) > 0]\}$ and $|Att(g) \cup Supp(g)| = |g|$. For example, a given input node may be represented by a group of arguments $\{\alpha_i, \ldots, \alpha_n\}$ and has a set of supporting/attacking arguments $\{\alpha_{ci}, \ldots, \alpha_{cn}\}$ corresponding to each argument of the group.

According to Table 1, to explain, Additive Monotonicity requires that the strength of a group of arguments is the sum of that of its supporters and attackers. Balance requires that the strength of a group of arguments differs from the sum of base scores of that group only if such a group is a target of other arguments. Weakening requires that when there are no supporters but at least one attacker, the strength of a group of arguments is lower than the total sum of base scores of that group. Conversely, Strengthening considers the situation when there are no attackers but at least one supporter instead. Weakening Soundness is loosely the opposite direction of Weakening, requiring that if the strength of a group of arguments is lower than the sum of base scores of that group, then the group must have at least one attacker. Similarly, Strengthening Soundness is loosely the opposite direction of Strengthening. Equivalence states that groups of arguments with equal conditions in terms of attackers, supporters and the sum of base scores within a group have the same strength. Attack Counting

Table 1. Dialectical properties for nQBAFs adapted from [1] and [2] where $\mathcal{G}$ represents the set of all groups of arguments in the argumentation framework.

#	Property	Name
1	$\forall g \in \mathcal{G}, \sigma(g) = \Sigma_{x \in Att(g)} \rho(x) + \Sigma_{x \in Supp(g)} \rho(x)$	Additive Monotonicity
2	$\forall g \in \mathcal{G}, Att(g) = \emptyset \wedge Supp(g) = \emptyset \rightarrow \sigma(g) = \Sigma_{x \in g} \gamma(x)$	Balance
3	$\forall g \in \mathcal{G}, Att(g) \neq \emptyset \wedge Supp(g) = \emptyset \rightarrow \sigma(g) < \Sigma_{x \in g} \gamma(x)$	Weakening
4	$\forall g \in \mathcal{G}, Att(g) = \emptyset \wedge Supp(g) \neq \emptyset \rightarrow \sigma(g) > \Sigma_{x \in g} \gamma(x)$	Strengthening
5	$\forall g \in \mathcal{G}, \sigma(g) < \Sigma_{x \in g} \gamma(x) \rightarrow Att(g) \neq \emptyset$	Weakening Soundness
6	$\forall g \in \mathcal{G}, \sigma(g) > \Sigma_{x \in g} \gamma(x) \rightarrow Supp(g) \neq \emptyset$	Strengthening Soundness
7	$\forall g_1, g_2 \in \mathcal{G}, Att(g_1) = Att(g_2) \wedge Supp(g_1) = Supp(g_2) \wedge \Sigma_{x \in g_1} \gamma(x) = \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_1) = \sigma(g_2)$	Equivalence
8	$\forall g_1, g_2 \in \mathcal{G}, Att(g_1) \subset Att(g_2) \wedge Supp(g_1) = Supp(g_2) \wedge \Sigma_{x \in g_1} \gamma(x) = \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_2) < \sigma(g_1)$	Attack Counting
9	$\forall g_1, g_2 \in \mathcal{G}, Supp(g_1) \subset Supp(g_2) \wedge Att(g_1) = Att(g_2) \wedge \Sigma_{x \in g_1} \gamma(x) = \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_1) < \sigma(g_2)$	Support Counting
10	$\forall g_1, g_2 \in \mathcal{G}, Att(g_1) = Att(g_2) \wedge Supp(g_1) = Supp(g_2) \wedge \Sigma_{x \in g_1} \gamma(x) > \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_1) > \sigma(g_2)$	Base Score Reinforcement
11	$\forall g_1, g_2 \in \mathcal{G}, g_1 <_a g_2 \wedge Supp(g_1) = Supp(g_2) \wedge \Sigma_{x \in g_1} \gamma(x) = \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_1) > \sigma(g_2)$	Attack Reinforcement
12	$\forall g_1, g_2 \in \mathcal{G}, Att(g_1) = Att(g_2) \wedge g_1 >_s g_2 \wedge \Sigma_{x \in g_1} \gamma(x) = \Sigma_{x \in g_2} \gamma(x) \rightarrow \sigma(g_1) > \sigma(g_2)$	Support Reinforcement

(Support Counting) requires that a strictly larger set of attackers (supporters, respectively) determines a lower (higher, respectively) strength. Base Score Reinforcement requires that a higher sum of base scores gives a higher strength. For the last two properties, we have to define the notion of weaker and stronger attack/support relations between sets.

Definition 8. *For any set* $A, B \in \mathcal{G}$*:*
$A <_a B$ *iff* $\Sigma_{x \in Att(A)} \sigma(x) > \Sigma_{x \in Att(B)} \sigma(x)$*;*
$A <_s B$ *iff* $\Sigma_{x \in Supp(A)} \sigma(x) < \Sigma_{x \in Supp(B)} \sigma(x)$*;*
$A >_a B$ *iff* $B <_a A$*;* $A >_s B$ *iff* $B <_s A$*.*

Then, Attack Reinforcement states that a weaker set of attackers determines a higher strength whereas Support Reinforcement states that a stronger set of supporters determines a higher strength.

Any nQBAF satisfies all given properties. This indicates that our LRP-based nQBAFs may align with human reasoning.

Proposition 1. *nQBAFs under LRP-based semantics satisfy Properties 1–12.*

Proof. We will make use of Lemmas 1-3 in the Appendix.

Property 1. Any group of arguments in $\mathcal{G}$ represents a single node. From Definition 7, the strength of this group is the sum of that of its members. We can view members as contributions this node receives from all nodes in the next layer. So overall, the total sum is the relevance score of this node. This also holds for the output node of interest by Definition 7 and our choice of LRP. By conservative properties of LRP, this sum is equal to the sum of contributions this node

gives to all the nodes in the previous layer. Since Algorithm 1 constructs a set of arguments A by considering all the pairs of nodes in adjacent layers, each contribution must be the strength of a unique argument in the previous layer. From Lemma 3, any non-attacking and non-supporting argument that does not represent an output node has zero strength, so the sum can be calculated from adding strengths of attacking and supporting arguments from the previous layer altogether. Hence this property is satisfied. □

Property 2. For an arbitrary group $g \in \mathcal{G}$, if $Att(g) = Supp(g) = \emptyset$ then by Property 1 we have $\sigma(g) = 0 + 0 = 0$. As γ gives all arguments a base score of zero, then $\Sigma_{x \in g}\gamma(x) = 0$. So $\sigma(g) = \Sigma_{x \in g}\gamma(x)$. As g is arbitrary, this is true for all $g \in \mathcal{G}$. □

Property 3. From Lemmas 1 and 2, any attacker has a negative strength while any supporter has a positive strength. For any group $g \in \mathcal{G}$, if $Att(g) \neq \emptyset$ and $Supp(g) = \emptyset$ then by Property 1 the strength $\sigma(g)$ must be negative. As γ gives all arguments a base score of zero, then $\Sigma_{x \in g}\gamma(x) = 0$. Hence $\sigma(g) < \Sigma_{x \in g}\gamma(x)$. □

Property 4. Similar to the proof of Property 3 above, any group $g \in \mathcal{G}$ that only has supporters has a positive strength by Property 1, which is more than $\Sigma_{x \in g}\gamma(x) = 0$. □

Property 5. Take arbitrary $g \in \mathcal{G}$ and assume $\sigma(g) < \Sigma_{x \in g}\gamma(x)$. We have to show that $Att(g) \neq \emptyset$. Assume $Att(g) = \emptyset$. There are two cases: $Supp(g) = \emptyset$ or $Supp(g) \neq \emptyset$. The first case leads to $\sigma(g) = \Sigma_{x \in g}\gamma(x)$ by Property 2, and the second case leads to $\sigma(g) > \Sigma_{x \in g}\gamma(x)$ by Property 4, both of which are contradictions. Hence $Att(g) \neq \emptyset$. □

Property 6. Take arbitrary $g \in \mathcal{G}$ and assume $\sigma(g) > \Sigma_{x \in g}\gamma(x)$. We have to show that $Supp(g) \neq \emptyset$. Assume $Supp(g) = \emptyset$. There are two cases: $Att(g) = \emptyset$ or $Att(g) \neq \emptyset$. The first case leads to $\sigma(g) = \Sigma_{x \in g}\gamma(x)$ by Property 2, and the second case leads to $\sigma(g) < \Sigma_{x \in g}\gamma(x)$ by Property 3, both of which are contradictions. Hence $Supp(g) \neq \emptyset$. □

Property 7. By Property 1, any group with similar attackers and supporters must have the same strength so this property is satisfied. □

Property 8. Assume we have two groups with similar sets of attackers and supporters. By Property 7, both groups have the same strength. Since any attacker has a negative strength (by Lemma 1), adding it to any group reduces the group strength by Property 1. Hence the property follows. □

Property 9. Assume we have two groups with similar sets of attackers and supporters. By Property 7, both groups have the same strength. Since any supporter has a positive strength (by Lemma 2), adding it to any group increases the group strength by Property 1. Hence the property follows. □

Property 10. Since every argument has a base score of zero (by our choice of γ), every group's sum of base scores is zero so the antecedent is always false. This property is therefore satisfied. □

Property 11. Take two arbitrary groups $g_1, g_2 \in \mathcal{G}$. Assume $g_1 <_a g_2$ and $Supp(g_1) = Supp(g_2)$. We have to show that $\sigma(g_1) > \sigma(g_2)$. Since g_1 and g_2 have the same supporters, $\Sigma_{x \in Supp(g_1)} \sigma(x) = \Sigma_{x \in Supp(g_2)} \sigma(x)$. As $g_1 <_a g_2$, then $\Sigma_{x \in Att(g_1)} \sigma(x) > \Sigma_{x \in Att(g_2)} \sigma(x)$. By Property 1, $\sigma(g_1) > \sigma(g_2)$ and this property is satisfied. □

Property 12. Take two arbitrary groups $g_1, g_2 \in \mathcal{G}$. Assume $Att(g_1) = Att(g_2)$ and $g_1 >_s g_2$. We have to show that $\sigma(g_1) > \sigma(g_2)$. Since g_1 and g_2 have the same attackers, $\Sigma_{x \in Att(g_1)} \sigma(x) = \Sigma_{x \in Att(g_2)} \sigma(x)$. As $g_1 >_s g_2$, then $\Sigma_{x \in Supp(g_1)} \sigma(x) > \Sigma_{x \in Supp(g_2)} \sigma(x)$. By Property 1, $\sigma(g_1) > \sigma(g_2)$ and this property is satisfied. □

5 Empirical Study

We apply our nQBAF variant as an underpinning argumentation framework for explaining a neural network-based image classifier. However, the network consists of several layers of multiple nodes, so the resulting argumentation framework will be too large to comprehend. To resolve this issue, we simplify the nQBAF variant further by grouping groups of arguments together. As each group of arguments has its well-defined strength, it can be treated as another type of argument that can be grouped together in a manner similar to its underlying arguments. Accordingly, all the dialectical properties are still satisfied by this additional layer of grouping. This double-layer grouping idea is, in essence, equivalent to grouping nodes of a neural network together. This idea is also exhibited in two approaches, namely *deep argumentative explanation* (DAX) [1] and the approach in [13] by Google. In this paper, we apply the LRP-based semantics on both approaches, each of which generates a separate set of explanations. We then analyse the obtained explanations qualitatively.

5.1 DAX Basics

DAX [1] is a general methodology for building local explanations (i.e. input-based explanations) for a neural network outputs. Unlike other explanation methods which are only based on inputs (and thus can be deemed to be flat), DAX takes account of the hidden layers too. DAX is based on extracting an argumentation framework from a neural network; explanations are then drawn from the framework, represented in a comprehensible format to humans. The extraction of the argumentation framework requires the choice of a semantics (for determining the strength of arguments) directly matching the behaviour of the neural network.

Here we apply DAX using our LRP semantics at its core. Also, we choose nQBAFs as the argumentation framework underpinning DAXs. We may theoretically achieve a full (local) explanation by viewing the entire nQBAF extracted from a neural network. However, the explanation would be too large for complex networks, therefore too complicated for humans to comprehend. To make things human-scale, we only consider a fragment of the nQBAF, in the spirit of [1], as well as grouping groups of arguments representing a single node (i.e. grouping nodes) together, and visualise the grouping as an explanation.

5.2 The Basics of Google's Method

Google's method [13] combines *feature visualisation* (i.e. what is a neuron looking for?, see [12]) with *attribution* (i.e. how does a specific node contributes to the output?) to generate a local explanation for a neural network output. We use the implementation of this method available at [7], changing the attribution method from a linear correlation to LRP. We leverage on the existing implementation's choices for visualisation.

5.3 Settings

For both methods, we aim to explain a Keras VGG16 model [16] (with linear activation function for the output layer) pretrained on the ImageNet dataset [5]. Since the whole model is too large, we only consider the last convolutional layer, explaining what the layer prioritises in a given image. We test our method in combination with DAX, comparing it to Google's method, on three images: a police van from [19], a barbell from [18], and a diaper from [20]. In all cases, we use the output node with maximum activation as the output class, with such an activation referred to as the output prediction.

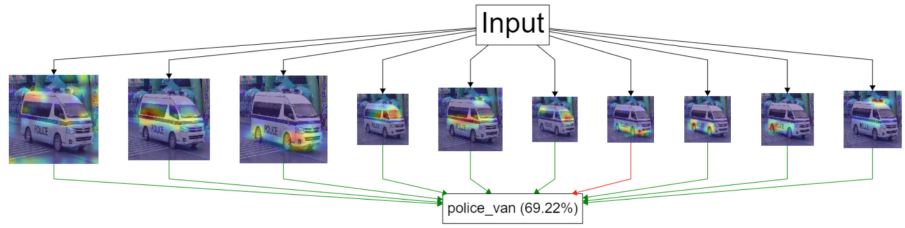
To generate explanations using DAX, we modify the code from the ArgFlow library [4] and apply to each of the three images. For each explanation, the size of each image illustrates the attribution thereof towards the output class, with red and green arrows depicting attacking and supporting the output class prediction respectively.

For Google's approach, we modify the code from [7] which is one of the Colaboratory notebooks in [13]. We then apply the code to the three images, each results in the set of images indicating parts of the original image. Each number below each factor refers to how much attribution each component has towards the output prediction. The arrow sizes also reflect these attributions.

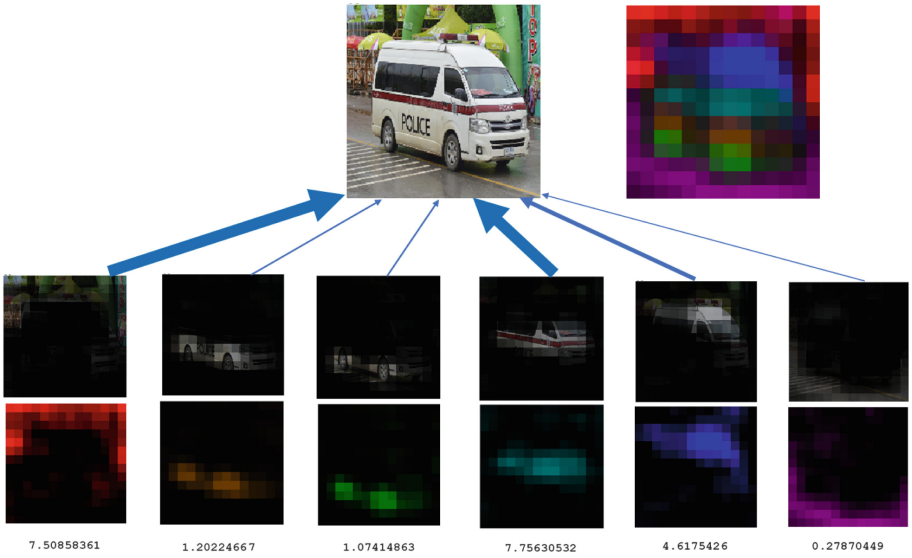
5.4 DAX Vs Google Comparisons

Image 1: Police Van. Explanations from both methods (as shown in Fig. 2) indicate that the model focuses mostly on the background and the red stripe of the van. There are some subtle differences between them mainly with the strength for each factor, but their factors are quite similar. However, an interesting point is that DAX considers the siren light of the van as one of the top six factors contributing to the output class prediction (according to the rightmost image of Fig. 2a) while Google's approach does not present this (arguably important) factor.

Image 2: Barbell. According to Fig. 3, both methods explain that the model focuses on the plates and the background. However, DAX considers the plates to contribute to the prediction more than the background, while it is the opposite for the Google's explanation. Somewhat counter-intuitively though, DAX considers the plates to both attack (the fourth image from the right of Fig. 3a) and support (the rightmost image from the right of Fig. 3a) the class prediction, even



(a) The DAX approach

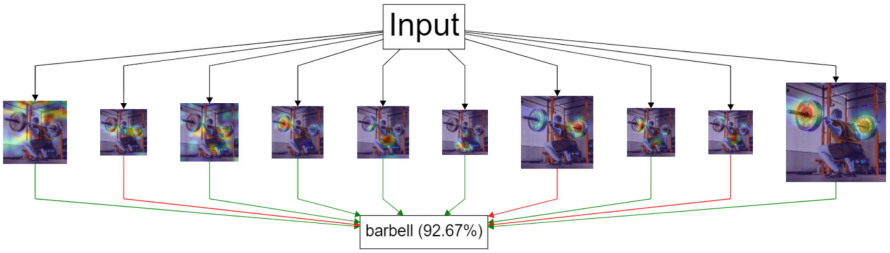


(b) Google's approach

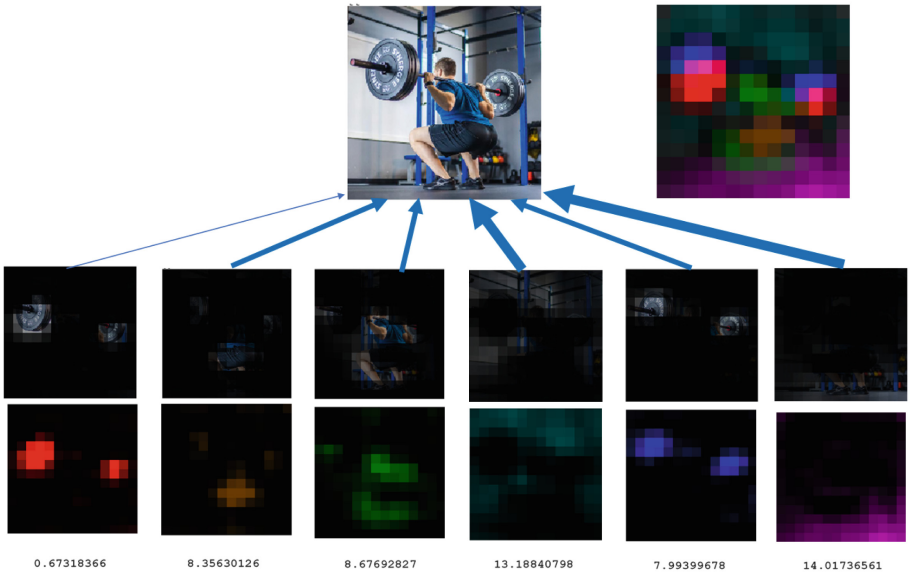
Fig. 2. Explanations given using (a) the DAX approach (with attacks in red and supports in green, either indicated in the filters or as arrows, and the size of arguments for the filters indicating their dialectical strength, see [1] for details) and (b) Google's approach for the police van image with the predicted class *police_van* (with arrows indicating support, and the size of arrows representing the LRP values). The police van image source is (https://bit.ly/3Fi1oqx). (Color figure online)

though the attacking argument (the fourth image from the right) is much less strong. If the DAX is faithful to the model, then this incongruence may result from an incongruence in the model.

Image 3: Diaper. From Fig. 4, both methods indicate that the model focuses on other things instead of the diaper. The DAX in Fig. 4a shows that the model focuses on the baby instead of the diaper. It even indicates that the diaper attacks the prediction of the class itself. In contrast, Google's explanation (Fig. 4b) indicates that the model focuses on the background and the diaper, giving the baby lower attributions.



(a) The DAX approach

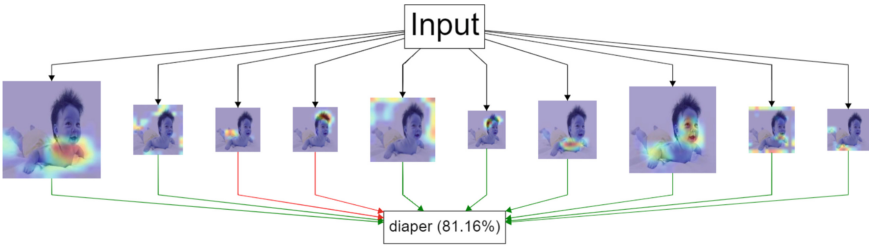


(b) Google's approach

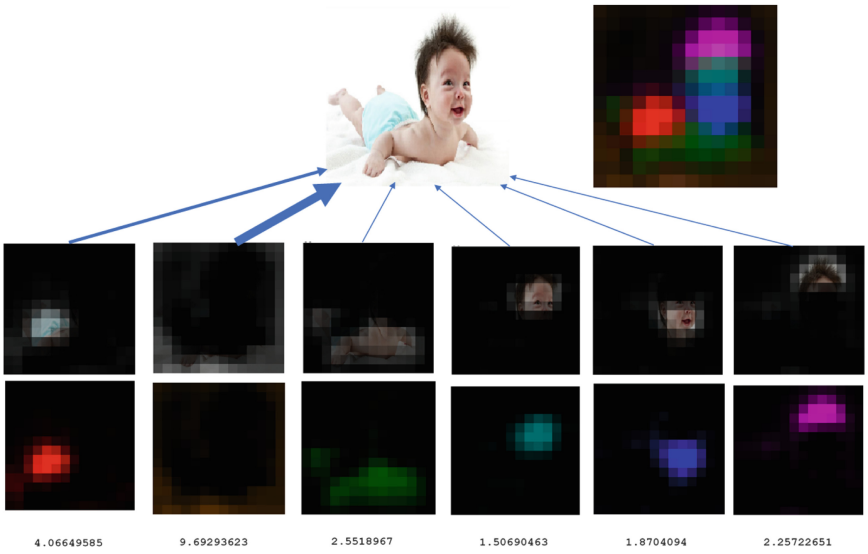
Fig. 3. Explanations given using (a) the DAX approach and (b) Google's approach for the barbell image with the predicted class *barbell*. The barbell image source is (https://amzn.to/3Db2xOQ). (Color figure online)

5.5 Discussion

The comparisons above clearly indicate that even with similar semantics (LRP), for the same model, explanations vary depending on how the grouping (of argument groups) is done. Google's approach seems to take account of the fact that concepts are usually recognised around particular positions of an image, whereas DAX only focuses on the concepts. DAX seems to unearth conflicts, with the same feature both attacking and supporting a prediction. Overall, more experimentation is needed to understand which explanation method is more "faithful" to the underlying model.



(a) The DAX approach



(b) Google's approach

Fig. 4. Explanations given using (a) the DAX approach and (b) Google's approach for the baby image with the predicted class *diaper*. The diaper image source is (https://bit.ly/3D8FZya). (Color figure online)

6 Conclusions

We presented a variant of Quantitative Bipolar Argumentation Frameworks (QBAFs) called neural QBAFs (nQBAFs) and considered how the LRP-based semantics satisfies the modified dialectical properties for nQBAFs. We also conducted preliminary experiments explaining an image classifier, by applying the LRP-based semantics to two approaches: Deep Argumentative Explanation (DAX) and Google's approach, and comparing both sets of explanations. DAX groups argument groups (i.e. nodes) in the same filter together, while Google's approach groups them by means of matrix factorisation optimising for activations. The comparison shows that how argument groups (each representing a node) are grouped can affect the resulting explanations. As future work, we plan

to conduct experiments with using nQBAFs for visualisation for text classification, in comparison with DAX and Google's approaches with LRP as well as other methods, such as smoothgrad [17], deeplift [10], gradcam [15] and TCAV [8]. Finally, it would be interesting to conduct experiments to assess demands on the cognitive load for end-users using different (instantiations of) visualisations.

Acknowledgements. The first author was funded in part by Imperial College London under UROP (Undergraduate Research Opportunities Programme). The last author was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101020934). Finally, Rago and Toni were partially funded 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 listed.

Appendix: Lemmas for Dialectical Properties Proofs

Lemma 1. *Any attacking argument has a negative strength.*
$\forall a \in A[\exists x \in \mathcal{P}(A)[a \in Att(x)] \rightarrow \sigma(a) < 0]$

Proof. Take arbitrary $a \in A$. Assume there exists some $x \in \mathcal{P}(A)$ such that $a \in Att(x)$. Since $a \in Att(x)$, $(a, x) \in Att$ so $c_{-}(a, x)$ is true, meaning $R_{\rho(a) \leftarrow \rho(x)} < 0$. As $\sigma(a) = R_{\rho(a) \leftarrow \rho(x)}$ by Definition 7, then $\sigma(a) < 0$. □

Lemma 2. *Any supporting argument has a positive strength.*
$\forall a \in A[\exists x \in \mathcal{P}(A)[a \in Supp(x)] \rightarrow \sigma(a) > 0]$

Proof. Take arbitrary $a \in A$. Assume there exists some $x \in \mathcal{P}(A)$ such that $a \in Supp(x)$. Since $a \in Supp(x)$, $(a, x) \in Supp$ so $c_{+}(a, x)$ is true, meaning $R_{\rho(a) \leftarrow \rho(x)} > 0$. As $\sigma(a) = R_{\rho(a) \leftarrow \rho(x)}$ by Definition 7, then $\sigma(a) > 0$. □

Lemma 3. *Any argument that neither supports nor attacks any group and does not represent an output node has zero strength.*
$\forall a \in A[\forall x \in \mathcal{P}(A)[(a, x) \notin Supp \wedge (a, x) \notin Att] \wedge \rho(a) \notin V_{d+1} \rightarrow \sigma(a) = 0]$

Proof. This proposition follows immediately from Definition 7. □

References

1. Albini, E., Lertvittayakumjorn, P., Rago, A., Toni, F.: Deep argumentative explanations (2021). https://arxiv.org/abs/2012.05766
2. Baroni, P., Rago, A., Toni, F.: How many properties do we need for gradual argumentation?. In: AAAI (2018)
3. Baroni, P., Rago, A., Toni, F.: From fine-grained properties to broad principles for gradual argumentation: a principled spectrum. Int. J. Approx. Reason. **105**, 252–286 (2019). https://doi.org/10.1016/j.ijar.2018.11.019

4. Dejl, A., et al.: Argflow: a toolkit for deep argumentative explanations for neural networks. In: International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1761–1763 (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–357 (1995). https://doi.org/10.1016/0004-3702(94)00041-X
7. Google, L.: Neuron groups - building blocks of interpretability (2018). https://bit.ly/3a483Xc
8. Kim, B., et al.: Interpretability beyond feature attribution: auantitative testing with concept activation vectors (TCAV). In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, 10–15 July 2018, vol. 80, pp. 2668–2677. PMLR (2018). https://proceedings.mlr.press/v80/kim18d.html
9. Lertvittayakumjorn, P., Specia, L., Toni, F.: FIND: human-in-the-loop debugging deep text classifiers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 332–348. Association for Computational Linguistics, November 2020. https://doi.org/10.18653/v1/2020.emnlp-main.24
10. Li, J., Zhang, C., Zhou, J.T., Fu, H., Xia, S., Hu, Q.: Deep-lift: deep label-specific feature learning for image annotation. IEEE Trans. Cybern. 1–10 (2021). https://doi.org/10.1109/TCYB.2021.3049630
11. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.-R.: Layer-wise relevance propagation: an overview. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700, pp. 193–209. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6_10
12. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**(11), e7 (2017). https://doi.org/10.23915/distill.00007
13. Olah, C., et al.: The building blocks of interpretability. Distill **3**(03), e10 (2018). https://doi.org/10.23915/distill.00010
14. Potyka, N.: Interpreting neural networks as quantitative argumentation frameworks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 7, pp. 6463–6470, May 2021. https://ojs.aaai.org/index.php/AAAI/article/view/16801
15. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
17. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise (2017)
18. Synergee Fitness Worldwide, I.: (2019). https://amzn.to/3Db2xOQ
19. Wataree: Police van Thailand (2019). https://bit.ly/3Fi1oqx
20. websubstance: Baby tummy time (nd). https://bit.ly/3D8FZya