Toward Explainable AI for Regression Models

Simon Letzgus[†1], Patrick Wagner[†2], Jonas Lederer[1], Wojciech Samek[*2,3], Klaus-Robert Müller[*1,3,4,5,6], and Grégoire Montavon[*1,3]

[1]Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany
[2]Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany
[3]BIFOLD – Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany
[4]Department of Artificial Intelligence, Korea University, Seoul 136-713, South Korea
[5]Max Planck Institute for Informatics, Stuhlsatzenhausweg 4, 66123 Saarbrücken, Germany
[6]Google Research, Brain Team, Berlin, Germany

***Abstract*—In addition to the impressive predictive power of machine learning (ML) models, more recently, explanation methods have emerged that enable an interpretation of complex non-linear learning models such as deep neural networks. Gaining a better understanding is especially important e.g. for safety-critical ML applications or medical diagnostics etc. While such Explainable AI (XAI) techniques have reached significant popularity for classifiers, so far little attention has been devoted to XAI for regression models (XAIR). In this review, we clarify the fundamental conceptual differences of XAI for regression and classification tasks, establish novel theoretical insights and analysis for XAIR, provide demonstrations of XAIR on genuine practical regression problems, and finally discuss the challenges remaining for the field.**

***Index Terms*—Explainable AI, Regression, Deep Neural Networks.**

I. Introduction

Machine learning, in particular deep learning, has supplied a vast number of scientific and industrial applications with powerful predictive models. As ML models are being increasingly considered for high-stakes autonomous decisions, there has been a growing need for gaining trust in the model without giving up predictive power. Explainable artificial intelligence (XAI) has developed as a response to the need of validating these highly powerful models [65], [9], [64]. Taking ML and XAI together, these technologies also offer a way of gaining new insights, e.g. nonlinear relations, into the complex data generating processes under study.

So far, the main body of work in the field of XAI has revolved around explaining decisions of classification models, often in the context of image recognition tasks [11], [90], [10], [92], [56]. Regression, a major workhorse in ML and signal processing, has essentially only received little attention. In practice to date, XAI approaches designed for classification are applied for regression problems. While such naive application to regression can occasionally still yield useful results, we will show in this paper that appropriate theoretically well-founded explanation models are necessary and overdue. For example, when explaining classification models, we can rely on the implicit knowledge associated with the class, and assume a decision boundary between the two or more classes. Additionally, the output itself can conveniently serve as a measure of model uncertainty or even evidence against the respective class can be analysed. In regression, on the other hand, we find none of these beneficial properties while we aim to explain a highly aggregated and application-specific model output that often corresponds to a physical entity with an attached unit.

In this paper, we will outline multiple challenges that emerge when explaining regression models, and we show how popular methods such as Layer-Wise Relevance Propagation (LRP) [10], Integrated Gradients [79], or the Shapley Value [71], [78], [48], can be applied or extended in a theoretically well-founded manner to properly address them. Our efforts are guided by how to formulate the question for which we would like an explanation in a way that addresses the user's interpretation needs. In particular, we aim for an explanation that inherits the unit of measurement of the prediction (e.g. physical or monetary unit). The explanation should also be sufficiently contextualized, not only by being specific to each data point, but also by localizing the explanation around a relevant range of predicted outputs. As an example, LRP and many other explanation methods assume (sometimes implicitly) a zero-reference value relative to which they explain or expand [64]. While in the classification setting one naturally explains relative to the decision boundary, i.e., $f(\boldsymbol{x}) = 0$, in regression settings we consider the reference value to be a crucial parameter to integrate the desired context into the explanation and avoid the latter to be dominated by (uninteresting) coarse effects. Therefore, we generally discourage practitioners to apply XAI methods developed for classification problems in an out-of-the-box manner to regression models.

We will provide several illustrative regression examples from various fields of signal processing, motivating the need for a distinct treatment of the regression and classification explanation problems respectively. In particular, we showcase our approach on a large CNN model for age prediction in face images, and we also demonstrate the capability of XAI for regression to deliver targeted scientific insights into the energy structure of molecules.

[†] S. Letzgus and P. Wagner contributed equally to this work.
[*] Corresponding authors: W. Samek, K.-R. Müller and G. Montavon.

II. A Brief Review of XAI

The field of Explainable AI is very broad, as it must consider at the same time various types of ML models and many interpretability requirements. The resulting multitude of methods can be divided along various conceptual lines. One can differentiate, for example, between the explanation of a model's overall (global) decision strategy [53], [44], [73] and uncovering a model's reasoning related to a specific sample (local) [10], [56], [11]. Methods can also be distinguished by how they present an explanation to the user: Some methods generate a maximally activating pattern [73], which can be interpreted to be prototypical for the decision strategy; some methods extract a contrastive example (criticism [39], counterfactuals [86]) which resembles the current data point, but without the features that cause a particular decision behavior; other methods identify a subset of features that are relevant to explain the decision outcome [17]; and finally, methods commonly referred to as attribution, assign to each input feature a score representing the contribution of that feature to the model output $f(\boldsymbol{x})$ [78], [10], [79]. A further distinction can be made between methods that explain based on individual input features [78], [11], [10], [79], combinations of input features (e.g. [16], [66], [26]), or higher-level concepts [40]. While explanation methods are often evaluated based on their technical merit (e.g. accuracy, runtime) [63], an increasingly relevant question is whether these explanations enable the human to truly understand the model at hand (also known as causability [35]) and whether these explanations can be turned into meaningful insights and decisions [25], [58], [12], [82].

This contribution will focus on single-instance (local), attribution-based explanations that assign a share of the model output $f(\boldsymbol{x})$ to the individual features of the respective input sample $\boldsymbol{x}$. While the focus on attribution may appear somewhat narrow, we note that attribution can serve as a building block for a broader class of explanations. For example, the SpRAy method [44] enables to turn a large collection of single-instance explanations into a single concise dataset-wide explanation. Also, attributions can be easily turned into a set of relevant features, e.g. by thresholding attribution scores. Lastly, because attributions characterize the decision function at a given data point [63], one can synthesize prototypical patterns or counterfactual examples from them as well, by removing the least or most relevant features. For the purpose of introducing the concept of attribution, it is useful to start with a simple setting such as a linear classifier. A linear (binary) classifier is typically implemented by a function

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i + b \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d)$ is the vector of input features, and where $w_1, \ldots, w_d, b$ are the parameters of the model. This function is typically followed by a sign function or some sigmoid function, where the output of such function either predicts the class directly, or assigns a probability of membership of the data point to the given class. The quantity $f(\boldsymbol{x})$ can be interpreted as the evidence for / against a particular class. In practice, it is convenient to focus on explaining the latter rather than the actual probability value or the classification result [52].

To obtain meaningful attributions, one also often considers the task of explanation relative to some neutral *reference* point or counterfactual, which could be for example the origin in input space, or a point similar to $\boldsymbol{x}$ but without the pattern that causes the particular classification outcome [10], [28], [86]. A reference point can be for example a root point $\widetilde{\boldsymbol{x}}$ of the prediction function f (i.e. $f(\widetilde{\boldsymbol{x}}) = 0$) with low distance $\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|$.

The concept of attribution and neutral point can be nicely illustrated for the simple linear example above. Assume we have computed some root point $\widetilde{\boldsymbol{x}}$ point of the function f. We can show that the linear model in Eq. (1) can be rewritten as:

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} w_i \cdot (x_i - \widetilde{x}_i) \tag{2}$$

where the root point now appears explicitly. With such reformulation, the bias term vanishes and the function is now structured as a simple sum over input features, which can also be interpreted as a first-order Taylor expansion at root point $\widetilde{\boldsymbol{x}}$ [10]. Equation (2) gives rise to a natural attribution scheme, which is to score input features according to the elements of the sum, i.e.

$$R_i = w_i \cdot (x_i - \widetilde{x}_i). \tag{3}$$

The score R_i can be interpreted as the contribution of input feature i to the function output. Each contribution is the product of the feature magnitude itself relative to the reference point $(x_i - \widetilde{x}_i)$ and the sensitivity of the model output to that feature (w_i). The vector $\boldsymbol{R} = (R_1, \ldots, R_d)$ forms the *explanation*. Note that this explanation method differs from a sensitivity analysis, where only the parameters w_i would be involved and where the same explanation would be consequently obtained for every input vector. Here instead, each input vector $\boldsymbol{x}$ produces an individual explanation.

In most practical applications, the ML models are nonlinear, which renders the simple method above inapplicable. While the question of explaining nonlinear models is still an active research topic, several approaches have been developed, with different assumptions on the model structure, the level of access we have to the structure of the model, model accuracy requirements, explanation accuracy requirements, and the amount of available compute power [78], [10], [56], [79], [14].

In this paper, we place our focus on 'post-hoc' explanations where we assume a given and already trained model (usually represented by some function f) and try to attribute the prediction for each data sample to its input features in a meaningful manner. In comparison to the 'ante-hoc' setting where we can set the structure of the model before training in a manner that makes it easy to extract an explanation, the post-hoc setting makes no such assumption. Instead, the post-hoc setting decouples the task of model building and the task of explanation, and it simply assumes that one has chosen the most suitable or best-performing model for the given task (e.g. the one delivering the highest accuracy, or incorporating the desired invariances). Within the scope of post-hoc attribution methods that we adopt here, three families of methods can be distinguished, (1) feature removal, (2) gradient-based, and (3) backward propagation. We present below these families of methods, highlighting some of their respective members, which we will show later on to be particularly suitable to adapt from classification to regression.

A. Removal-based explanations

Perhaps the most straightforward way of testing the contribution of a feature to the output of an ML model is to remove it and measure the difference at the output of the model. *Shapley values* [71], [78], [48] provides a theoretical framework for this type of explanation. The framework originated in game theory where a related problem, that of sharing the total gain between a set of cooperating players, was considered. It was shown that given a fairly limited set of axioms that an explanation should satisfy, known as efficiency ($\sum_i R_i = f(\boldsymbol{x})$, also known as conservation or completeness), symmetry, linearity, and null-player, the Shapley value [71] is one and the unique solution satisfying all axioms.

Let $\boldsymbol{x}$ be a data point composed of N features. Let $\sum_{\mathcal{S}|i\notin\mathcal{S}}$ be a sum over all subsets of features that do not contain feature i, and $\boldsymbol{x}_\mathcal{S}$ the data point $\boldsymbol{x}$ where only features in $\mathcal{S}$ have been retained (the other features have been set to zero or the value of some meaningful reference point $\widetilde{\boldsymbol{x}}$). The Shapley value is given by:

$$R_i = \sum_{\mathcal{S}|i\notin\mathcal{S}} \alpha_\mathcal{S} \cdot \big[f(\boldsymbol{x}_{\mathcal{S}\cup\{i\}}) - f(\boldsymbol{x}_\mathcal{S})\big]$$

where $\alpha_\mathcal{S} = |\mathcal{S}|!(N-|\mathcal{S}|-1)/N!$. In other words, the Shapley value tests the effect of adding the feature i assuming various subsets of features are present, and weighting the different subsets by the factor $\alpha_\mathcal{S}$.

The Shapley value can be applied to any function $f(\boldsymbol{x})$, whether it is a neural network, a random forest, a kernel classifier, etc. Because there are exponentially many subsets that need to be evaluated, the Shapley value is however computationally infeasible for most problems with more than 15 or 20 input dimensions. Only for certain classes of models or allowing certain approximations, Shapley values can be computed in polynomial time [47], [4]. For general models, random approximations of Shapley values where elements of the sum are sampled according to $\alpha_\mathcal{S}$ can be used, which allows scaling the analysis to higher dimensions. For a recent review of feature removal-based methods, we refer to [20].

B. Gradient-Based explanations

Another set of methods bypasses the need to evaluate the function for multiple perturbations by relying instead on the gradient (e.g. [11]). The latter can be extracted quickly with a single forward/backward pass in the function's computational graph. A simple approach that generalizes Eq. (3) consists of replacing the weight w_i by the gradient evaluated locally and integrating it locally along some trajectory $\{\boldsymbol{x}(t)\colon 0 \le t \le 1\}$ connecting the root point and the data point i.e.

$$R_i = \int_0^1 \frac{\partial f}{\partial x_i(t)} \cdot \frac{\partial x_i(t)}{\partial t} \cdot dt$$

The method is known as *Integrated Gradients* (IG) [79]. Like Shapley values, the method satisfies the conservation property $\sum_i R_i = f(\boldsymbol{x})$. An advantage of the integrated gradients approach is that it does not require evaluating the function exponentially many times, and each gradient evaluation readily provides a d-dimensional explanatory feedback without having to test each input feature separately. The number of required function evaluations corresponds to the level of discretization of the integral. However, in comparison to the Shapley value approach, IG only explores a small region of the input space, typically the segment connecting the reference point to the data point. Hence, the explanation may be biased by this local scope and it may therefore fail to integrate important components of the decision process.

Approaches such as SmoothGrad [75] mitigate the problem by repeating the analysis multiple times with some randomness factor (in the case of IG, one can for example randomize the integration path [27], [64]). This comes however at an increased computational cost. For a review of gradient-based explanation techniques, we refer the reader to [3].

C. Propagation-based explanations

A last category of methods aims to leverage the neural network structure to produce explanations [90], [77], [10], [72]. The layer-wise relevance propagation (LRP) method [10], in particular, solves the explanation task by starting at the output of the network, and reverse propagating the prediction, layer by layer, until the input variables are reached. The propagation at each layer is implemented by a purposely designed propagation rule. Let j and k denote indices of neurons at two consecutive layers, and let a_j and a_k denote their respective activations, with w_{jk} the weight connecting these two activations. Denote by R_k the 'relevance' received by neuron k from the layers above, which can be interpreted by the contribution of neuron k in its corresponding layer to the output prediction $f(\boldsymbol{x})$. The propagation rules used by LRP are typically of the form:

$$R_j = \sum_k \frac{z_{jk}}{\sum_{0,j} z_{jk}} R_k$$

where z_{jk} models the contribution of neuron j to the activation of neuron k, where $\sum_k$ is a sum over neurons of the current layer, and where $\sum_{0,j}$ is a sum over neurons in the layer below plus the bias represented as an additional neuron with activation $a_0 = 1$. It is easy to show that when the neural network does not have biases (or if biases are not included in the propagation rule), we have conservation from layer to layer, i.e. we can build the chain of equalities

$$\textstyle\sum_i R_i = \cdots = \sum_j R_j = \sum_k R_k = \cdots = f(\boldsymbol{x})$$

which results in the same conservation property as for the Shapley value or integrated gradients.

In practice, various LRP rules can be used. For deep networks with ReLU activations, LRP-γ [50] was proposed, and sets $z_{jk} = a_j \cdot (w_{jk} + \gamma w_{jk}^+)$. The hyperparameter γ can be tuned to improve robustness of the explanation. Choosing $\gamma = 0$ yields typically noisy explanations that coincide with a simple 'Gradient × Input' explanation. Choosing γ larger, especially in lower layers, yields more robust explanations that become empirically closer to those of the Shapley value.—Here, the main advantage of LRP is that explanations can be produced in a single forward / backward pass, which makes the method applicable to highly complex models with hundreds or thousands of input features, such as convolutional neural networks (CNNs) used in vision.

The LRP method, although most studied on deep ReLU networks, can also be applied or extended to different types of models, e.g. long short-term memory networks (LSTMs) [8], graph neural networks (GNNs) [66], Bayesian Neural networks [15] or non-neural networks, e.g. for anomaly detection [37], [60] or clustering [36]. For a detailed overview of LRP, we refer the reader to [50], [64].

D. *Other methods*

For comprehensiveness, we also briefly mention some other approaches to explanation. Some explanations are defined as the result of an optimization problem (e.g. [28]). Certain explanation methods address the slightly different task of extracting a subset of relevant input features rather than scoring these features [17], [89]. Some explanations are obtained by training a local surrogate model which is easier to explain, e.g. a linear model or a decision tree (e.g. [56]). Some explanation methods assume a particular self-explainable structure (e.g. additivity or attention mechanisms) in the model [16], [92], [14], [59], [87]. Finally, some methods, commonly referred as higher-order methods aim to identify not individual features, but groups of features, that contribute only when occurring jointly [32], [81], [26], [66], [21], [47]. Moreover, methods have been proposed or adjusted to fit specific model architectures (e.g. LSTMs [8] or GNNs [88], [66]) as well as characteristics connected to different data types, such as time-series [19], [46], [57] or natural language [22].

In parallel to the broadening of explanation methods, there have been several works providing unifying views on the different explanation techniques [48], [3], [64], [20], drawing connections between gradient-based and perturbation-based explanation methods, as well as showing how propagation-based methods reduce to gradient-based methods for certain choices of parameters.

III. XAI for Regression (XAIR)

Regression is an important subfield of ML and signal processing, which focuses on predicting quantities that are real-valued instead of categorical. Such real-valued predictions are predominant in many practical application scenarios, often related to physics, engineering or economics. Explainable AI techniques have only recently started to be applied in such regression scenarios. For example, [80] have applied the Shapley value framework to shed light on a heat transfer phenomena modeled by a ML model. In the area of hydrology, [42] are making use of integrated gradients to identify what factors (precipitation, temperature, radiation, etc.) contribute to river discharge, as modeled by an LSTM neural network. In the area of energy engineering, [55] use Shapley values to attribute building energy consumption, as predicted by the XGBoost ML model, to input features such as unit density, number of floors, or built year. Finally, [66] contributes an extension of the LRP method to GNNs, and applies it in a learning scenario for quantum chemistry to explain molecular atomization energy in terms of individual atoms or groups of atoms.

So far, the most common way of applying XAI to regression has been to use solutions developed in the context of classification, and apply them to the regression case, either by using them in an out-of-the-box manner (e.g. [80], [55], [42], [66]) or by translating a regression into a multi-class classification problem (e.g. [43], [12]). While this direct approach is technically straightforward and has delivered useful practical insights, more precise and understandable explanations can be gained from carefully revisiting Explainable AI in the specific context of regression, and some initial steps have been taken in the particular area of counterfactual explanations [76], [33].

In this paper, we address the problem of XAI for regression in a broad manner and identify two important specificities of the regression problem that require an adaptation of XAI: The first specificity has to do with the nature of the prediction itself, where the prediction of a regression model is often attached with a particular *measurement unit* (e.g. physical or monetary). If using an appropriate explanation method, the unit of the prediction can be inherited by the explanation, which enables further interpretability of the explanation result for the user. The second specificity relates to the typically higher amount of information contained in a real-valued prediction compared to a classification (especially when the regression task has a high signal-to-noise ratio). For the explanation to faithfully address the user's need, further contextualization is required, for example, by specifying a particular *reference value*. These two aspects are illustrated in Fig. 1. We elaborate on them in more detail in the sections below.

A. *Explaining Quantities with Units (e.g. Physical / Monetary)*

The first aspect that distinguishes the task of explaining an ML regression model from that of explaining an ML classifier is the nature of the prediction itself, and what can we gain from attributing the prediction to the input features.

In the classification case, the output of the ML model can take various forms, for example, a decision outcome, a logit score, a class probability, a distance to the decision boundary, or else. For most of them, these forms are rather abstract and hard to interpret for the end-user. This is also the case for an attribution of these quantities to the input features, and consequently, these attributions are often used merely as a visualization (e.g. a histogram or a heatmap). Alternatively, simpler forms of explanation such as a list of the top-k most

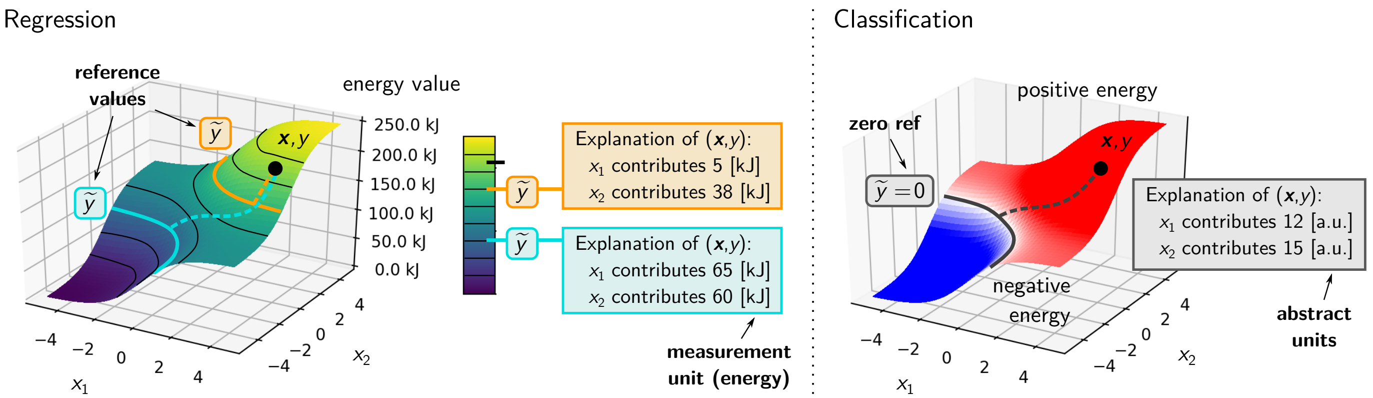

Fig. 1. Overview of Explainable AI for regression and for classification. The variable $\boldsymbol{x}$ denotes some point of interest for which we would like to explain the prediction $y = f(\boldsymbol{x})$. XAI for regression differs in several ways from the usual classification scenario: (1) the explanation scores can be interpreted as physical quantities of the same units as the prediction; (2) the explanation is produced relative to some user-defined reference value $\widetilde{y}$, and the latter can substantially affect the explanation.

relevant features or some visual mask in pixel space, have also shown to be efficient for explaining classification outcomes. This preference for simple explanations is also reflected in the design of XAI benchmarks, where only the set of most relevant features or the ordering of features from most to least relevant usually enters into consideration [63], [89], [88].

In the regression case, however, where we typically predict real-valued quantities such as price, energy, etc., one has the opportunity to retain the measurement unit in the explanation. In a real-world system, information such as the value (e.g. in some monetary unit) attributed to a particular subentity, or the amount of energy attributed to a particular interaction or a subsystem, have a more direct use. They may provide a mechanism for pricing, or for predicting what amount of energy might be transferred from one subsystem to another, e.g. a chemical reaction. To ensure this physical interpretation of attributions, the property of conservation (i.e. $\sum_i R_i = f(\boldsymbol{x})$) becomes crucial.[1] Decomposition methods such as the Shapley value, integrated gradients or LRP, which we have introduced in Section II, are then strongly preferred over methods such as sensitivity analysis or feature selection, which per se do not fulfill this property.

B. Explaining with a User-Provided Reference Value

The second aspect that separates classification and regression from the perspective of explanation, has to do with the type of output domain, in particular, whether it is categorical or real-valued.

In classification, we often like to formulate the problem of explanation in terms of a baseline scenario, e.g. "what makes a deep neural network predict there is a cat in an image vs. not a cat". Such questions can be addressed in technical terms by considering the logit (or probability) score $f(\boldsymbol{x})$ at the output of the network, and analyze it relative to some *fixed* value, the decision threshold, representing where the decision "cat" transitions to "not a cat". This is roughly the regime at which techniques such as the Shapley value, integrated gradients, and LRP usually operate.

In regression, however, outputs are not categorical and the question arises naturally with what exactly we should contrast the prediction of interest. In this paper, we argue that a *reference value* should be specified as part of the question for which the user seeks an explanation. For example, the user may ask "why a physical system is predicted to have an energy of $2500\,\mathrm{kJ/mol}$ vs. $1000\,\mathrm{kJ/mol}$", or "why an item is currently valued at 1200 dollars compared to its usual 1000 dollars price". XAIR methods thereby enable the user to get a contextualized explanation with respect to a specific reference value. This delivers more targeted explanations than obtained by existing XAI methods with their often implicit and fixed reference values. We will further motivate and illustrate the benefits of extending XAI frameworks in this manner in the following section.

C. Motivating Example: Auction Scenario

To illustrate the specificity of explanation in the regression context, we consider a toy example consisting of a first-price auction scenario, where a seller would like to sell some item at the highest possible price to the auction participants (see also Fig. 2). Let $P_1, \ldots, P_d$ describe the prices (expressed in some given monetary unit) at which the d participants place their bids. We assume a typical regression problem where these prices are predictable using some function $P_i = \mathcal{P}(\boldsymbol{x}_i)$ where $\boldsymbol{x}_i$ is a representation of the ith participant, e.g. based on demographic features or bidding history.

The outcome of the first-price auction is given by the maximum value between the d bidders, i.e.

$$y = \max(P_1, \ldots, P_d).$$

For simplicity we assume two bidders ready to purchase at price $P_1 = 1100$ and $P_2 = 900$ monetary units, respectively. Note that if we would produce an explanation with a reference price of $\widetilde{y} = 0$, both bidders would be attributed roughly half of the money as they share responsibility in setting the price higher than zero.

[1] Note, that with a simple scaling of attributions any explanation method can formally fulfill the conservation criterion, yet, it is unlikely that the rescaled scores truly reflect the contribution of each variable to the predicted score.

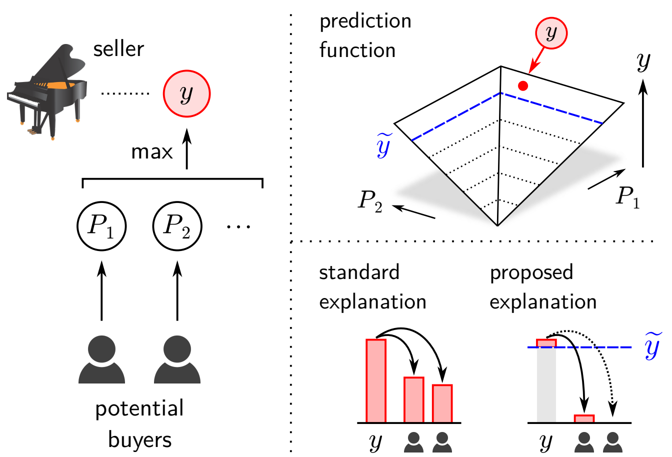

Fig. 2. Auction scenario. The ML function predicts the outcome (selling price) of the auction. The prediction must be explained in a meaningful manner in terms of the individual buyers and their features.

However, if we assume that the seller sets a base price of 1000 monetary units (e.g. corresponding to the value of the item according to the seller), and that the seller does not sell the item if no one bids above that base price, the question to answer becomes how to explain the benefit of the seller, i.e. "why the selling price is y and not 1000". In other words, we want to know what is the explanation of the model output when considering a reference value $\widetilde{y} = 1000$. Analyzing the real-valued function P at $\widetilde{y} = 1000$ intuitively reveals that only the first bidder contributes to the benefit of the seller as the second bidder is unwilling to buy the item even at the base price. (We will present methods in Section IV to produce these explanations systematically.)

An alternative approach would have been to explain the classification decision $P > 1000$ based on standard XAI techniques. However this approach is limited by the fact that multiple functions (e.g. P, $\tanh(P-1000)+1000$, $P^2/1000$, etc.) support the same decision boundary. Yet most of them do not correctly reflect the price away from the decision boundary. Using them would lead to distorted explanations, that would not reliably identify buyers' contributions to the selling price. This problem is fully avoided by explaining the regression function directly.

Overall, with the help of a regression-based explanation, the seller acquires a precise knowledge of the amount each participant of the auction has contributed to the selling price, specifically, to the difference between the selling price and the base price (or the seller's personal benefit). These scores can then be further attributed to e.g. the participants' demographic features, so that one can identify in which demographic group future auctions should be advertised for maximum gain.

IV. Bringing XAI Methods to Regression

The motivating example above has highlighted how important the choice of reference value $\widetilde{y}$ and the conservation of units are in the regression case. They are an integral part of formulating the question the user seeks an explanation for. State-of-the-art explanation techniques, however, still lack the possibility to incorporate user-provided reference values $\widetilde{y}$ and therefore the flexibility to meaningfully incorporate this added information.

Our starting point is to state the function we wish to explain. We call such function $g(\boldsymbol{x})$, and define it as:

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) - \widetilde{y} \qquad (4)$$

i.e. the original function $f(\boldsymbol{x})$ *relative* to the user-specified reference value $\widetilde{y}$. We now consider whether existing XAI techniques, in particular, techniques presented in Section II, readily apply to this newly defined function, or whether modifications of these techniques are needed.

Removal-based and *gradient-based* explanation methods (presented in Sections II-A and II-B) are designed in a way that they can apply to *any* function. These methods typically only require a root point $\widetilde{\boldsymbol{x}}$ at which the function has value zero. We note, however, that if $\widetilde{\boldsymbol{x}}$ is a root point of the function $f(\boldsymbol{x})$, it is typically *not* a root point of the function $g(\boldsymbol{x})$, and the function $g(\boldsymbol{x})$ has potentially multiple new root points $\widetilde{\boldsymbol{x}}'$ to choose from. Such dilemma is highlighted in Fig. 3.

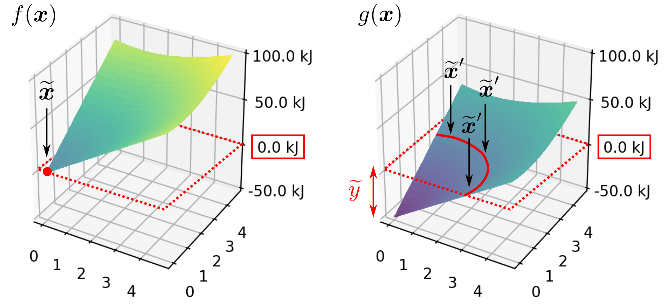

Fig. 3. Effect of shifting the function on the function's root points. After shifting, the original root point $\widetilde{\boldsymbol{x}}$ is no longer a root point, and there are potentially many new root points to choose from.

Choosing one particular root point $\widetilde{\boldsymbol{x}}'$ might introduce some spurious modeling bias into the explanation. Alternately, one can consider a set of root points (e.g. weighted according to some probability distribution) and compute the expectation over the resulting explanations. This reduces (although not fully eliminates) the risk of modeling bias but results in a significant increase of computational cost. Lastly, one can tamper with the function $g(\boldsymbol{x})$. For example, when $0 \leq \widetilde{y} \leq f(\boldsymbol{x})$, one can apply the clipping nonlinearity $g^+(\boldsymbol{x}) = \max(0, g(\boldsymbol{x}))$ to ensure that the original root point remains a root point, i.e. $g^+(\widetilde{\boldsymbol{x}}) = 0$. Consequently, one can use $\widetilde{\boldsymbol{x}}$ for the analysis.[2] While this avoids introducing explanation bias through the choice of a new reference point, the rectification function itself may introduce bias, although less significant. Furthermore, this tampering approach is only applicable when the value $\widetilde{y}$ is located between 0 and $f(\boldsymbol{x})$.

In contrast to *removal-based* or *gradient-based* explanation methods, *propagation-based* methods such as LRP (cf. Section II-C) do not require a reference point $\widetilde{\boldsymbol{x}}$ to produce an explanation. Propagation-based methods rely instead on the

[2] A similar construction can be obtained for the case $f(\boldsymbol{x}) \leq \widetilde{y} \leq 0$, where we define instead the function $g^-(\boldsymbol{x}) = \min(0, g(\boldsymbol{x}))$.

neural network's internal representation of the function to progressively redistribute the prediction from layer to layer until the input features are reached. These methods assume, however, that the neural network supporting the explanation is reasonably disentangled. While the newly defined function $g(\boldsymbol{x})$ can be seen as a neural network (by simply adjusting the top-layer bias), the neural network representation may not be sufficiently disentangled to explain subtle variations of the function $f(\boldsymbol{x})$ around the provided reference value $\widetilde{y}$. Furthermore, in Explainable AI, biases are typically having the role of 'unexplained' factors [3], and thus, the explanation of such bias-adjusted model would essentially remain the same except for an unexplained term.

We propose two strategies to enable propagation-based methods to explain $g(\boldsymbol{x})$: First, *retraining*, where a surrogate neural network is trained to replicate $g(\boldsymbol{x})$ on some relevant band of prediction values. Second, *restructuring*, where the last few layers are rewritten (without retraining) in a way that the network outputs $g(\boldsymbol{x})$, and the representation has been sufficiently altered to enable the desired fine-grained explanation. These two approaches are presented in detail below.

A. Retraining

The first approach consists of retraining the network, so that it accurately predicts the new function $g(\boldsymbol{x})$ on some relevant band of function values $\tau^- \leq y \leq \tau^+$. In particular, we can define some surrogate neural network model $\hat{g}(\boldsymbol{x}, \theta)$, where θ represents the set of parameters of the network, and optimize the objective

$$\mathcal{E}(\theta) = \sum_{n=1}^{N} \big(\hat{g}(\boldsymbol{x}_n; \theta) - g(\boldsymbol{x}_n)\big)^2 \cdot 1_{\{\tau^- \leq g(\boldsymbol{x}_n) \leq \tau^+\}}$$

where $1_{\{\cdot\}}$ is an indicator function. To prevent the learning algorithm to only recalibrate top-layer biases, we propose to freeze these biases to their original value, or to incorporate additional penalties to the objective function.

Note that unlike a classification-based retraining approach, where the model would be instructed to classify each example into bins representing different ranges of regression values, the approach described here preserves the unit of measurement (e.g. price or energy) at the output of the original model. This also holds for explanations of such regression models, where attribution on the input features produces scores expressible in the same unit.

For the retraining approach to deliver good results, one needs to pay particular attention to the data used for retraining. Choosing a dataset that is too limited may result in a model that predicts in the same way but is prone to the apparition of Clever Hans effects [44], [6] which may unfaithfully explain the original prediction function. Conversely, the new model may also inhibit preexisting Clever Hans effects from the original model concealing its weaknesses and preventing a meaningful model validation. Lastly, for complex models that require large training sets, retraining results in a high computational cost, especially if a large number of reference values are of interest.

B. Restructuring

To avoid the retraining step, an alternative approach consists of manually restructuring the network in a way that the network output becomes exactly $g(\boldsymbol{x})$. Restructuring should be profound enough for the neural network internal representation to be meaningfully affected and capable of influencing the explanation process in the desired manner. The restructuring approach is remotely related to 'neuralization' [37], [36], [60], which converts non-neural network models (e.g. kernel machines) into functionally equivalent neural network models, without retraining.

Our restructuring approach assumes that the last two layers of the neural network are ReLU and linear respectively. (This includes the special case where the linear layer is an average pooling layer or a combination of a linear and an average-pooling layer.) Restructuring is implemented as a backward pass, where the reference value adjustment of the function $g(\boldsymbol{x})$ is propagated first in the linear layer, and then in the ReLU layer. These two steps of propagation, which we describe below, are profound enough to meaningfully change the representation and the produced explanation.

Step 1: Propagation in the linear layer: Denote by $y = \sum_j a_j w_j + b$ the equation for the original top-level linear layer. The shifted top-layer can be rewritten as the same linear layer with the reference value $\widetilde{y}$ transformed into a reference activation vector $\widetilde{\boldsymbol{a}}$:

$$\begin{aligned} y - \widetilde{y} &= \textstyle\sum_j a_j w_j + b - \widetilde{y} \\ &= \textstyle\sum_j (a_j - \widetilde{a}_j) \cdot w_j + b \end{aligned} \tag{5}$$

For this equation to hold, we need to choose $\widetilde{\boldsymbol{a}}$ such that $\sum_j \widetilde{a}_j w_j = \widetilde{y}$. In practice, there are many possible choices for $\widetilde{\boldsymbol{a}}$. We propose a 'flooding' strategy where the reference point $\widetilde{\boldsymbol{a}}$ is chosen in a way that only large neuron activations (corresponding to global effects) remain in $\widetilde{\boldsymbol{a}}$. Specifically, we search for a reference point $\widetilde{\boldsymbol{a}}^{(\text{flood})}$ at the intersection of Eq. (5) and the parameterized line:

$$\{(\boldsymbol{a} - t \cdot \mathbf{1})^+, t \in \mathbb{R}\}$$

where $(\cdot)^+$ applies element-wise. This reference point absorbs all large activations (coarse-grained effects), so that the explanation is contextualized and can focus on small activations representing fine local variations.

Example. *To illustrate the effect of the restructuring approach on the internal representation and subsequently on the explanation, we consider the auction example of Section III-C. In the case where there are two buyers, the max function between two positive values can be written as a neural network using the composition of ReLU neurons:*

$$f(\boldsymbol{x}) = \frac{1}{2}\underbrace{(x_1 + x_2)^+}_{a_3} + \frac{1}{2}\underbrace{(x_1 - x_2)^+}_{a_4} + \frac{1}{2}\underbrace{(x_2 - x_1)^+}_{a_5}$$

In the case where the two actors bid a similar price, e.g. 1100 *and* 900 *monetary units respectively, the neuron activation* a_3 *modeling the coarse effect (i.e. the price average) has a much higher value than* a_4 *and* a_5 *that model the fine effect* *(i.e. the price difference). This implies that the application of our restructuring approach to this example will significantly reduce the influence on a_3 on the explanation but preserve the influence of a_4 and a_5. This modified representation helps in turn to deliver the desired explanation where the difference between the two bidders, modeled by the neuron a_4, is now more strongly expressed.*

Note that the proposed restructuring approach is only directly applicable to a network with a single output because the reference point $\widetilde{\boldsymbol{a}}$ depends on the weights of the output neuron. In presence of a multi-output neural network, different restructurings need to be applied for each output neuron.

Step 2: Propagation in the ReLU layer: The restructuring step we have performed above implies that each ReLU neuron in the layer below is now attached with an offset $\theta = -\widetilde{a}_j$. Such a shifted ReLU neuron is not directly explainable within common propagation-based explanation frameworks such as LRP. We observe however that we can rewrite the shifted ReLU activation as a linear combination of three standard ReLU activations:

$$\rho(z_j) - \widetilde{a}_j = \rho(z_j - \widetilde{a}_j) + \rho(-z_j) - \rho(-z_j + \widetilde{a}_j)$$

where ρ denotes the ReLU nonlinearity. The ReLU neuron and its restructuring are depicted in Fig. 4. In practice, each neuron in the given layer is first triplicated; then all outgoing weights of that layer are multiplied by 1, 1, and -1 for each neuron copy respectively; incoming weights are multiplied by 1, -1 and 1; and finally, the offset $\widetilde{a}_j$ is multiplied by 1, 0, and -1 before being included in the corresponding neuron biases.

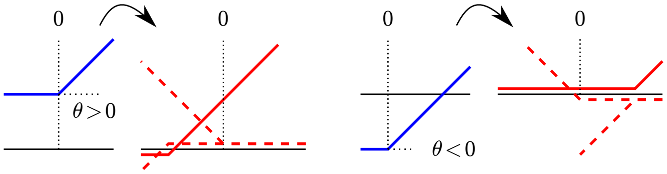

Fig. 4. Transformation of shifted ReLUs neuron (blue) into a sum of multiple non-shifted ReLUs (red). The latter is functionally equivalent but more amenable to the LRP propagation technique.

With these two steps of restructuring, we arrive at a functionally equivalent architecture composed exactly of the same layer types, and where the number of neurons in the last hidden layer has increased by a factor of three. The main advantage of the resulting neural network is that it supports a better redistribution of the output score on these layers (e.g., using LRP). Furthermore, the restructuring approach guarantees that the function to explain remains the same.

Compared to the retraining approach, the restructuring approach avoids an inadvertent modification of the model's decision strategy through a suboptimal retraining algorithm or an improper choice of data distribution for retraining. As a downside, the restructuring approach requires a particular top-layer structure, which makes the approach less model-independent. In practice, however, many state-of-the-art models, for example, neural networks for computer vision have this top-layer structure.

V. Validation Experiments

In this section, we will evaluate the two proposed XAIR approaches, retraining and restructuring, in combination with the LRP explanation technique. Produced explanations will be evaluated on a collection of low-dimensional datasets where the application of the Shapley value remains computationally feasible and can (for these scenarios) serve as a reference explanation method for comparison. Our two proposed approaches will be compared against two baselines which simply consist of shifting or scaling the original explanation. We will demonstrate that our two approaches perform substantially better than the baselines. In particular, our restructuring approach does not incur any retraining cost and thus runs as quickly as the original LRP procedure.

A. Datasets

For our validation experiments, we make use of 5 different low-dimensional datasets (between 4 and 13 dimensions), referred to as *max*, *linear*, *friedman*, *diabetes*, and *boston*. The *max* dataset corresponds to our motivational example of Section III-C where we compute $y = \max(x_1, \ldots, x_d)$ from $d = 8$ uniformly distirbuted features. The *linear*[3] dataset is a simple linear regression problem where 4 out of 8 input dimensions have predictive power. The *friedman*[4] regression problem is another synthetic dataset where a non-linear function is predicted from independent uniformly distributed features [29]. The *diabetes*[5] dataset is a well-known regression example composed of 442 instances, where one has to predict disease progression from 10 patient features. Finally, the *boston housing*[6] dataset consists of 506 instances of housing values in Boston, which are to be predicted from 13 geographical features. To facilitate optimization (and without any loss of generality), for all experiments, we standardize all features independently and rescale targets such that they are between zero and one (the learned model can be rescaled back to the original units after training for the purpose of explanation).

B. Experimental setup

We train on each dataset a two-layer neural network, consisting of an intermediate layer of 256 ReLU neurons and one linear neuron as output. We optimize weights to minimize the mean square error between prediction and target, using stochastic gradient descent (varying number of epochs and learning rates for different datasets). Overall, our trained models are able to solve the tasks reasonably well with coefficients of determination (R^2) higher than 0.9.

Given the simple low-dimensional datasets used in our experiment, we can generate reference explanations by applying the Shapley value [71], [78], which is expensive to

[3] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman2.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html

[6] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html

compute but theoretically well founded. Because the Shapley value is a removal-based explanation technique, we adopt the clipping strategy $g^{+}(\boldsymbol{x})$, one of the options proposed in Section IV), which introduces the minimum amount of bias into the explanation. Consequently, we only retain in our evaluation instances where the clipping approach is applicable, i.e. satisfying $0 \leq \widetilde{y} \leq f(\boldsymbol{x})$. We then generate attributions for the respective reference values using our two proposed methods (retraining and restructuring). For the retraining approach, we set the parameter controlling the band of function values to $\tau^{-} = -0.3$ and $\tau^{+} = \infty$, and we initialize the layers with the weights from the original model. In both cases we use the LRP-γ rule with $\gamma = 2.5$ and $\gamma = 0$ for the first and second layers. We compare the proposed methods against two simple baselines that represent a simple post-processing of a standard LRP explanation (without training or restructuring), in particular, our 'shift' baseline computes the new relevance scores $R'_i = R_i - \widetilde{y}/d$ and our 'scaling' baseline computes $R'_i = R_i \cdot (y - \widetilde{y})/y$. We evaluate our method for different reference values $\widetilde{y}_q = qf(\boldsymbol{x})_{\max} + (1-q)f(\boldsymbol{0})$.

Explanation performance is measured as the mean square error (MSE) between the produced explanation and the Shapley-based reference explanation. Unlike many explanation evaluation metrics (e.g. based on IOU scores [91] or pixel-flipping curves [63]), the MSE explicitly accounts for the magnitude of the feature attributions and not only their ordering. When reporting the results, we also divide the MSE score by the average MSE of random attributions scaled to the difference between y_i and $\widetilde{y}$. A value of 1, therefore, represents the error equivalent to a random assignment of attributions that satisfy the completeness property. Error bars are calculated by repeating every experiment 10 times.

C. Results

Table I evaluates the proposed methods and baselines for the different datasets over a selection of reference values. It can be observed that the proposed restructuring and retraining approaches outperform the baselines for all configurations. Figure 5 (left) shows the average normalized MSE per $\widetilde{y}_q$, excluding the shift-baseline due to its poor performance. Figure 5 (right) shows the average normalized MSE per dataset, again without the shift-baseline. Average performance is qualitatively consistent across all datasets and reference values with model restructuring in the lead. In absolute terms, all methods show a higher MSE on the boston dataset compared to the others.

When comparing 'model restructuring' to 'explanation scaling' (the closest competitor among baselines) we observe an overall decrease in MSE of more than 22%. The advantage ranges from around 15% for the max dataset up to 34% for the friedman dataset. In terms of reference values, the MSE of the restructuring method is only around 17% lower for $\widetilde{y}_{0.25}$. The difference increases with larger reference values up to more than 25% for $\widetilde{y}_{0.5}$.

Retraining has an overall 17% lower MSE compared to explanation scaling. Over the different datasets, the relative performance advantage of retraining compared to the scaling baseline ranges from 3% for the boston dataset up to more than 27% for the friedman dataset.

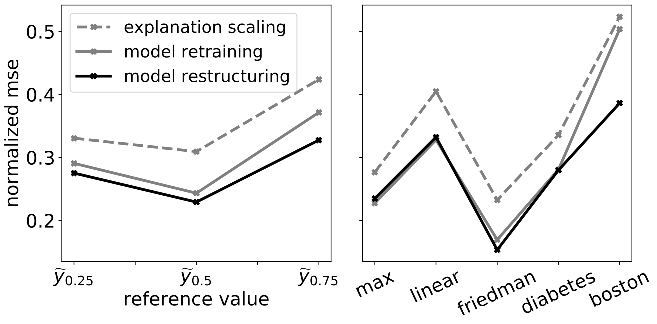

Fig. 5. Mean normalized explanation error (in MSE) of methods by choice of reference value $\widetilde{y}_q$ (left) and dataset (right).

To gain further insights on the performance improvement of restructuring and retraining vs. explanation scaling we show in Figure 6 an exemplary explanation from the friedman dataset. Firstly, note the different magnitudes on the y-axis induced by the different reference values which all methods are able to account for. However, the restructuring and retraining approach are able to better approximate the structural changes in the attribution with increasing reference values, especially for features of high importance. That illustrates the major benefit of restructuring and retraining over a simple scaling of the original LRP attributions.

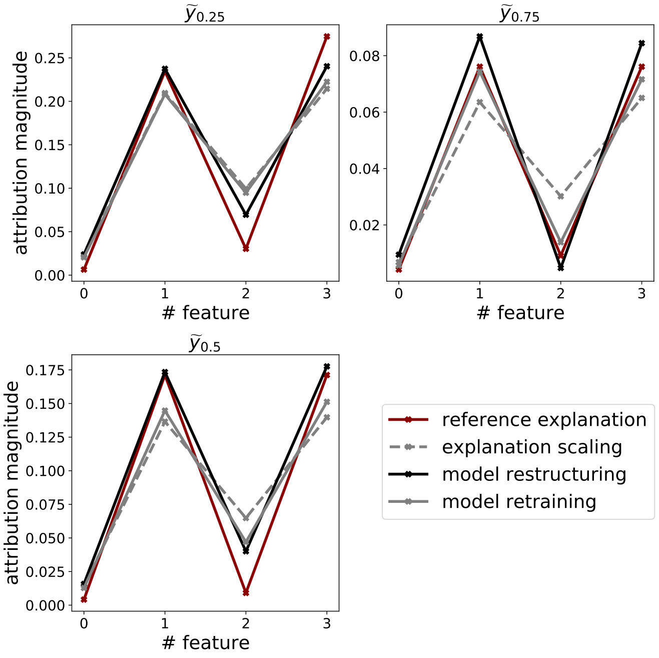

Fig. 6. Explanations for a selected example from the friedman dataset, where we compared different explanation methods (explanation scaling, model restructuring and model retraining) to the Shapley-based reference explanation, for the respective reference values $\widetilde{y}_q$.

Overall, we find the restructuring best suited to explain models with respect to alternative reference values. Furthermore, we see some more options connected to the network structure

TABLE I

RESULTS FOR VALIDATION EXPERIMENTS. PERFORMANCE MEASURED IN NORMALIZED MSE. INPUTS NORMALIZED, TARGETS SCALED BETWEEN 0/1, AND MODEL WITH 256 NEURONS IN HIDDEN LAYER WITH RELU-ACTIVATIONS.

		Baselines		Our methods	
dataset		explanation shift	explanation scaling	model retraining	model restructuring
max	$\widetilde{y}_{0.25}$	1.0343 ± 0.171	0.3316 ± 0.015	**0.2763 ± 0.014**	0.2968 ± 0.016
	$\widetilde{y}_{0.5}$	4.9424 ± 0.703	0.2603 ± 0.017	**0.2042 ± 0.015**	0.2102 ± 0.017
	$\widetilde{y}_{0.75}$	22.9945 ± 3.097	0.2380 ± 0.022	0.2022 ± 0.016	**0.1979 ± 0.020**
linear	$\widetilde{y}_{0.25}$	2.3851 ± 0.411	0.4008 ± 0.030	**0.3179 ± 0.026**	0.3549 ± 0.040
	$\widetilde{y}_{0.5}$	8.5959 ± 1.967	0.3407 ± 0.049	**0.2450 ± 0.037**	0.2674 ± 0.054
	$\widetilde{y}_{0.75}$	57.8571 ± 47.084	0.4731 ± 0.078	0.4208 ± 0.139	**0.3747 ± 0.042**
friedman	$\widetilde{y}_{0.25}$	0.8433 ± 0.109	0.2067 ± 0.037	0.1835 ± 0.038	**0.1345 ± 0.047**
	$\widetilde{y}_{0.5}$	3.5789 ± 0.788	0.2194 ± 0.044	0.1525 ± 0.042	**0.1241 ± 0.040**
	$\widetilde{y}_{0.75}$	47.0557 ± 16.629	0.2714 ± 0.121	**0.1723 ± 0.105**	0.2017 ± 0.052
diabetes	$\widetilde{y}_{0.25}$	0.6823 ± 0.080	0.3547 ± 0.037	0.3205 ± 0.040	**0.3122 ± 0.024**
	$\widetilde{y}_{0.5}$	2.8446 ± 0.524	0.3443 ± 0.082	0.2835 ± 0.075	**0.2784 ± 0.048**
	$\widetilde{y}_{0.75}$	7.0100 ± 2.107	0.3075 ± 0.126	**0.2383 ± 0.102**	0.2489 ± 0.068
boston	$\widetilde{y}_{0.25}$	0.7655 ± 0.111	0.3593 ± 0.029	0.3553 ± 0.043	**0.2779 ± 0.032**
	$\widetilde{y}_{0.5}$	2.9533 ± 0.384	0.3819 ± 0.041	0.3315 ± 0.044	**0.2662 ± 0.048**
	$\widetilde{y}_{0.75}$	43.8898 ± 15.818	0.8289 ± 0.098	0.8238 ± 0.123	**0.6149 ± 0.133**

that could further increase the performance of restructuring. For smaller networks with only 20 neurons, for instance, we have observed the overall MSE difference between restructuring and explanation scaling to increase. This improvement in performance can be explained by the fact that restricted model capacity prevents large components to be scattered onto many neurons and consequently perceived as many small components and flooded. We speculate that approaches that add neurons incrementally during training may also help to further dissociate coarse from fine effects in the architecture and consequently lead to a further increase of explanation accuracy.

Finally, we recall that in some cases, restructuring may not be possible, for example, because the model's top layer structure does not allow for the proposed adjustments, or may simply not be desirable, because it requires to tamper with the network internals outside the common training and explanation interfaces. In such cases, the more flexible retraining approach (which comes as a close second in our benchmark), can be used as an alternative. Our experiments in the next section will demonstrate the capabilities of the restructuring and retraining approaches on large real-world models.

VI. APPLICATION TO REAL-WORLD REGRESSION PROBLEMS

In this section, we will demonstrate on two real-world application examples, one from the computer vision domain and one from the field of quantum chemistry, the benefits of our XAIR approach. Our approach, which preserves the unit of the prediction in the explanation, and lets the user specify a meaningful reference value, will be shown to enable valuable insights into regression models that cannot be obtained using standard XAI methods out of the box.

A. Facial Image Age Prediction

As a first illustrative application example, we analyze a popular regression task from computer vision: age prediction from facial images [34], [7], [1]. This problem represents a typical regression task but is often approached by classifying images into different age bins and therefore seems particularly suited to highlight the difference between XAIR and standard XAI. The aim is to highlight the facial features associated with a certain age prediction. This has been done before for the classification approach [43], [1] but, to the best of our knowledge, never for regression models.

For this analysis, we make use of a dataset containing $\sim$ 20k facial images associated with biological age[7] (biased toward younger ages). Each image is pre-processed so that all of them have the same size (200x200) and the faces are aligned and centered. We used a VGG-16 [74] model pre-trained on ImageNet [23], [62] as a feature extractor followed by one ReLU layer with 256 neurons, a dropout-layer, and a final linear layer mapping the 256 neurons to a real-valued age prediction. We selected three representative reference values across the whole target domain, namely 10, 40, and 80 years, and restructured the network's final layers as described in section IV-B. To contrast these explanations with those delivered by a standard classification XAI pipeline, we also trained a binary classification model by appending a sigmoid activation layer that classifies whether images are labeled with age below or above 50 years. During training, we minimized the MSE or the binary cross-entropy for the regression and classification cases respectively, both using Adam. In all cases we used LRP-$\alpha_1\beta_0$ rule [10], [52] in the convolutional layers and LRP-ϵ rule [10] (where biases are ignored) for the fully connected layers.

Because retraining is very time consuming in case of deep and complex models, we opt for the restructuring approach. For restructuring the model with respect to the desired reference value we used the flooding approach described in Section IV-B. For the cases where $f(\boldsymbol{x}) < \widetilde{y}$, the flooding equation was slightly modified to $(\boldsymbol{a} - (t \cdot \mathbf{1}_{w \leq 0} - t/4 \cdot \mathbf{1}_{w>0}))^+, t \in \mathbb{R}$,

[7] https://www.kaggle.com/frabbisw/facial-age

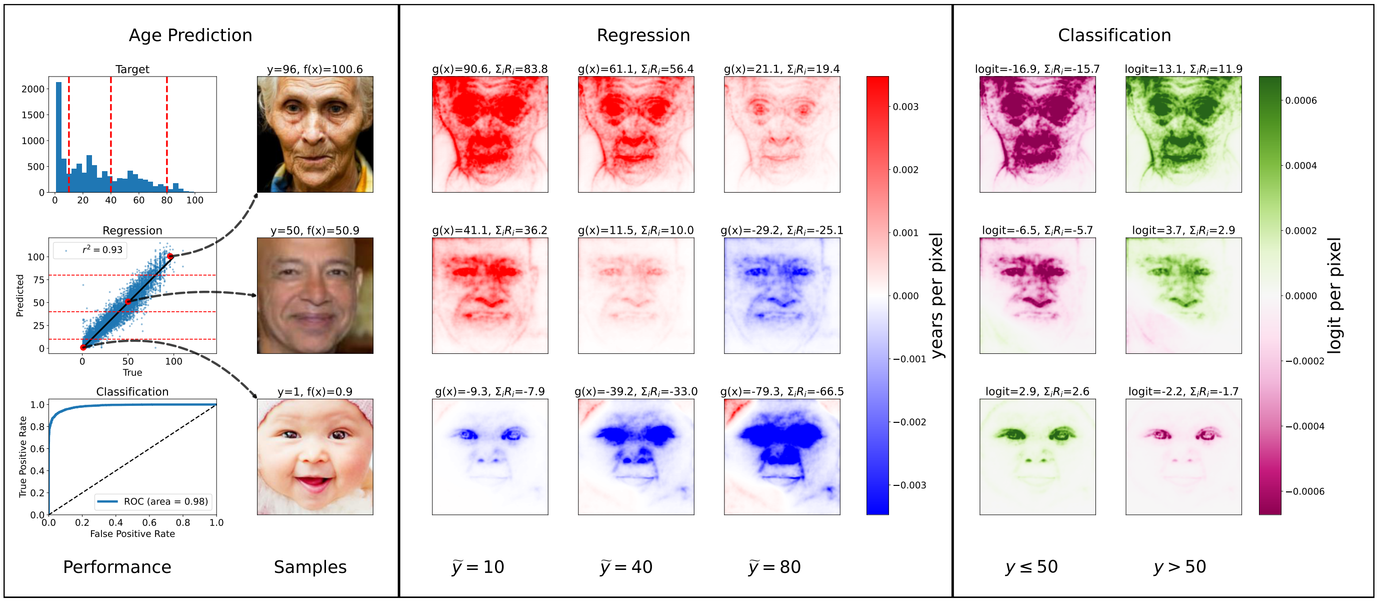

Fig. 7. Explanations for age prediction based on facial images. The *Age Prediction* section shows the distribution of targets (top - blue histogram) including the three reference values (10, 40, 80 - vertical red lines), performance plots for both, the regression and the classification model, and the three representative sample images (with true ages 1, 50, 96). The explanations of the regression models for three different reference values are aligned as a 3x3 grid in the center of the figure. Red/blue pixel color indicates positive/negative evidence relative to a given reference point. On the very right vertical section, the classifier decisions are visualized with green/pink pixels indicating evidence for/against the respective class (> 50, ≤ 50).

where the asymmetry between positive and negative weights allows to explore activation patterns associated to higher ages.

The results of this experiment are presented in Fig. 7, where we picked three representative examples from the test set such that the actual age is in relative proximity to a selected reference value. Looking at the regression explanations (center block of the figure) we observe a gradient from strongly red heatmaps (composed of positive scores) in the top-left corner, to strongly blue heatmaps (composed of negative scores) in the bottom-right corner. This reflects the conservative nature of the explanation technique where the sum explanation scores correspond approximately to the signed difference between the predicted age $f(\boldsymbol{x})$ and the chosen reference value $\widetilde{y}$.

When looking at the explanations for the 96 year old person (top row) with respect to reference value of age 10 (left) we observe that all facial features significantly contribute to the high age model output. The closer the reference point moves to the true age, the more differentiated the heatmap becomes. It becomes apparent that it is mainly the area around the person's eyes that, according to the model, makes the person look older than, for example, 80 years (right). This example once more emphasizes one of the essential motivations for this paper: In regression, explanations relative to faraway reference values might be useful in certain cases, but often it is more interesting to explain relative to a close-by reference value.

Further insights about the model can be gained from the 50-year-old person's explanations. Looking at the explanation relative to the nearest age reference $\widetilde{y} = 40$ (in the very center of the figure), we observe an absence of strongly positive or negative scores over the person's facial features. This indicates that, according to the model, the person does not have any facial features that would be particularly atypical for his age group. In comparison, classifier-based explanations could not convey such fine-grained insights.

Finally, the baby picture explanations (bottom row) allow for similar conclusions as for the old person's example (but with a flipped sign). The explanation relative to reference value age 10 (left) shows that the model mainly used the region around the eyes to decide that the person was younger than 10 years and explanations for reference points further away from the true age do not allow for such a detailed interpretation. Moreover, the example enables another interesting observation: with rising reference values the model allocates more importance to the babies hairline/hat-area. A possible interpretation is that in the context of young age headware is irrelevant as the model mainly uses the young facial features for its judgement. In reference to older ages, however, the model seems to have associated the hat with an age increase, which could be due to the hat resembling white hair. This illustrates once more how the choice of different reference values allows for a contextualization of the explanations that reaches beyond the abilities of any classification model (compare classification heatmaps on the right side).

In summary, the experiment shows that our XAIR approach allows for richer explanations and therefore more insights into what features make a face appear to belong to a certain age prediction. Moreover, the preservation of units allows for an attribution of a number of years to specific facial features, a quality that might be even more beneficial in other domains, such as quantum chemistry (cf. Section VI-B).

B. *Explanations in Quantum Chemistry*

Recently, ML methods have contributed broadly to atomistic simulations e.g. for ultra-fast approximate solutions of the Schroedinger Equation (factor 10^7 faster [61], [85]) and fast and accurate force fields for molecular dynamics (e.g. [18], [54], [83]). Especially, the development of ML architectures for predicting molecular properties has been advancing rapidly with ever more complex models achieving gradual improvements in performance (e.g. [61], [69], [30], [70], [84], [41]). However, a good model should exhibit qualities beyond high accuracy, namely, it should be able to capture certain principles from physics [69], [85]. This allows for increased confidence in the model results through sanity-checks by an expert as well as new insights into physical phenomena previously not understood (e.g. [69], [38], [82]). XAI methods, especially XAIR methods, can be a key for both which we will demonstrate in the following sections by applying our proposed retraining approach in the quantum chemistry domain.

In particular, we have a closer look at the effect of different reference values when explaining predicted molecular quantum-chemical properties. These properties are real-valued and expressed in various physical units (e.g. kcal/mol, etc.). We chose to predict the atomization energy, which is the energy of a molecule relative to a total separation into individual atoms. The atomization energy is a negative quantity, since forming a molecule is energetically preferable compared with a separation into individual atoms. Another way of thinking about the atomization energy is that it is the energy needed for breaking all bonds of the compound and separating its atoms infinitely far apart, and which is effectively equivalent to the 'negative' atomization energy.

For the prediction of the negative atomization energy, which we will simply call "energy" from now on, we utilize the SchNet [68], [70], [67], a GNN architecture specifically designed for predicting molecular properties. For the explanations, we employ a higher-order extension of LRP designed for explaining GNNs (GNN-LRP [66]). The output prediction is attributed to so-called walks that comprise several nodes and edges of the graph. Since the predicted energy corresponds to interactions between atoms, we expect most relevance to be attributed to the edges and in particular to the bonds. Furthermore, it is common sense that bonds of higher order are more stable and thus more energy is needed to break them and it is important that SchNet models also capture this property. We will show that using a reasonable reference value facilitates the extraction of insights from the model, by precisely answering the following questions:

1) Are bonds accountable for the major energy contributions (as chemists would expect)?
2) Are the energy contributions of bonds increasing with increasing bond order (as chemists would again expect)?

In our experiments, we first train the SchNet model with reference value zero (achieving MAE of $0.017\,\mathrm{eV}$) and compute the respective explanations. To get more contextualized explanations, we consider a reference value corresponding to the mean energy per atom, averaged over all molecules consisting of single bonds only, and apply our retraining approach. For the retrained model (MAE of $0.015\,\mathrm{eV}$), the average relevance attribution of single atoms should be driven close to zero, and we should therefore be able to better distinguish the different bond orders in the explanation. In Fig. 8, we compare the explanation of the two SchNet models mentioned above. The explanation (relevance heatmap) corresponding to the SchNet trained with reference value zero is depicted in Fig. 8 (left). The strength of different bonds can hardly be distinguished, and a large part of the relevance is attributed to the atoms and not to the bonds. In contrast, for the model trained with non-zero reference value shown in Fig. 8 (right), the bond structure becomes more prominent, and we can also distinguish between bonds of different bond strengths. The double bond and the aromatic ring exhibit large relevance scores, while little relevance is attributed to single bonds.

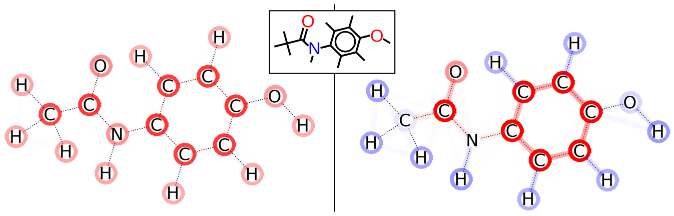

Fig. 8. Comparison between (left) SchNet trained with zero reference value, and (right) SchNet retrained with non-zero reference value for paracetamol (acetaminophen). The heatmaps show the aggregated relevance scores for atoms and bonds, respectively. The structure formula shows the bond orders present in the molecule.

To substantiate the observations above, we evaluate the relevance attributions of the two models for a set of 100 molecules randomly drawn from the QM9 dataset [61]. To this end, we calculate the mean relevance attributions corresponding to atoms and different bond types for the two SchNet models, respectively. Relying on the relevance conservation, we can associate the relevance aggregated on particular bonds and atoms as their respective energy contributions, as done in [66]. The evaluation is shown in Fig. 9. Both models capture the physics w.r.t. bond orders fairly well, which is an increasing energy contribution with increasing bond order. For the SchNet model trained with reference value at zero, the atoms contribute the most energy to the total prediction, while the atoms of the model trained with non-zero reference contribute less. This explains why even though both models are designed according to chemical intuition, the predictions of the zero reference model are hardly interpretable. It is because the large relevance attribution to atoms dominates the relevance attribution of interest on the bonds.

To further emphasize the importance of choosing a well-suited reference value for explaining models from quantum chemistry, we propose the following experiment: The QM9 dataset consists of relaxed molecules in their ground state. Distorting the structure of these molecules would hence yield energetically less favourable states. It is essential that the ML model captures this feature by predicting an energy change, in our case to lower energies. To this end, we consider a $H_7C_5N_3O$ molecule from the QM9 dataset. We predict the energy for the molecule in its relaxed state as well as for a distorted structure, where we stretched the bond between the Carbon and the Oxygen atom by $0.4\,\mathrm{\AA}$. The zero reference model and the non-zero reference model predict an energy change of $1.45\,\mathrm{eV}$ and $2.46\,\mathrm{eV}$, respectively. Even though the model predictions for the distorted molecule differ from each

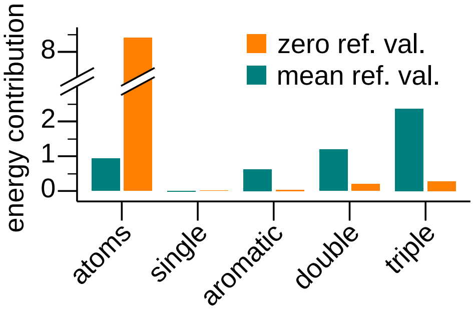

Fig. 9. Average energy contribution per bond and atom for a SchNet model trained with zero-reference value (zero ref. val.) and another SchNet model trained with non-zero-reference value (mean ref. val.). The bonds are distinguished by bond order. The depicted energy contribution has arbitrary units.

other, both models predict a decrease in energy, which is what we expect from a well-trained model. For both models, we compute the relevance scores for the ground state structure and for the distorted structure, respectively. A reasonable relevance attribution is expected to indicate the region on the molecule that is responsible for the energy change.

The results are depicted in Fig. 10. For the non-zero reference model, we can clearly see a local change in the relevance heatmap when stretching the C-O bond. The edge and atom relevance in the distorted region strongly decreases and even switches to a negative value. Comparing left and right heatmap shows that the structure distortion results in a less favourable molecule conformation. Furthermore, the heatmap clearly indicates which substructure of the molecule is responsible for the energy difference with respect to the relaxed structure. For the zero reference model, in contrast, we can hardly spot any difference between the heatmaps of relaxed and distorted structures.

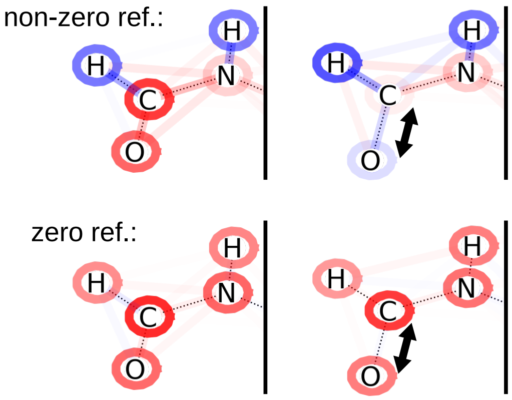

Fig. 10. Relevance scores obtained by GNN-LRP for the (left) relaxed and (right) distorted geometry of $H_7C_5N_3O$. The distorted molecule is obtained by stretching the C-O bond as indicated by the arrow. The first row shows the relevance attribution for the non-zero reference model. The second row shows the relevance heat-map for the zero reference model. The depicted relevance scores are aggregated for each bond and atom, respectively. To increase visibility of the edge relevance attribution, we show the scaled relevance scores $R' = R^{0.7}$.

In conclusion, both SchNet models solve the regression task with comparable accuracy. The statistical evaluation shows that both models capture the physics behind the problem. However, choosing a well-suited reference value clearly facilitates explaining the distinction between physical concepts known to an expert, such as bond orders. Moreover, it enables the identification of energetically (un)favorable substructures. It thereby offers a better validation of the model to a chemist user, in line with chemical intuition.

VII. Best practice guidelines

This section provides a set of good practices in order to successfully produce XAIR explanations. The discussion focuses on the two main aspects treated in this paper, which are (1) the conservation of measurement units from the modeled system up until the production of the explanation, and (2) the incorporation of the reference value specified by the user into the explanation process. Our recommendations are meant to apply regardless of the concrete XAIR application, or the exact choice of neural network model. For more general recommendations on XAI we refer to [64].

A. *Conservation of measurement units*

Many ML tasks have the objective to predict a quantity that inherently is expressed with some measurement unit, whether it is the energy of a physical system, the value of an economic good, or else. Often, for the purpose of facilitating training, the targets are transformed in some way, such as converting the real-valued prediction problem into a simpler problem (e.g. classification or ranking), or applying some nonlinear transformation. When the measurement units are of interest, we discourage such intermediate step as it loses the latter interpretation.

Predict using the original measurement unit.

To ensure that the lack of target normalization does not make training more difficult, we can still perform the training in transformed space (e.g. standardized targets) but we need to make sure the model is rescaled back to the original units after training.

We now would like to translate the unit-preserving predictions into unit-preserving explanations. Explanation techniques such as the Shapley value, integrated gradients, or LRP are all based on a conservation principle, which when applied to a real-valued prediction, enables an understanding of the produced explanation in terms of the same measurement unit.

Use a conservative explanation technique.

By applying this recommendation, we ensure that such a precise meaning of the explanation is available, and it can be used to make quantified inferences, which are more useful to the user than a simple visualization. Examples include estimating the stability of some subgroup of atoms in a molecule, or estimating the value of individual components of some more complex economic good.

B. Choosing a reference value

The choice of reference value has a profound impact on the quality of the explanations as shown multiple times throughout the paper. This being said, it is impossible to recommend a general reference value due to the multitude of potential questions the practitioner might want to ask using the XAI method. Therefore the appropriate choice relies solely on the respective applicant's question and domain knowledge.

Ask the user for an appropriate reference value $\widetilde{y}$.

Once a reference value has been specified by the user, one needs to ask how to integrate this information into a meaningful explanation. Various ways of producing such an explanation have been described in Section IV, in particular, we have proposed *retraining* and *restructuring* as two practical solutions that enable scaling to large models. In general, if restructuring is possible, i.e. the top-level structure of the neural network consists of a ReLU layer followed by some linear layer (this includes simple average pooling), we recommend using this approach as it shows the highest performance in our benchmark while requiring least intervention.

If possible, incorporate $\widetilde{y}$ using restructuring.

If instead the structure of the network is different, or even if the structure is appropriate but we do not want to tamper with the neural network internals and use exclusively the usual training and explanation interfaces, the retraining approach becomes the favorable alternative, provided that we have access to the training data and some compute resources. Retraining is however a more complex procedure that requires further heuristics. We address this topic in the section below.

C. Retraining guidelines

Apart from being able to manage potentially high associated computational cost, the most important practical requirement for retraining is to have access to a suitable training dataset. In practice this might be prevented for legal reasons such as privacy or close source as well as prohibitively large dataset size. For the data to be suitable it is crucial to have the same (or similar) distribution as the original data to arrive at a model which implements $g(x)$ as precisely as possible.

Ensure access to appropriate training data.

Assuming we have the necessary data, and once the corresponding targets are adjusted to the desired reference value, the model implementing $g(x)$ can be retrained, if possible, using the original training configuration. For coherence, we recommend training with the adjusted target at least until the training error reaches a comparable level as in the original model training, rather than training for a fixed number of epochs.

Retrain until the original accuracy is reached.

Because the retraining might substantially perturb the original learned solution, it might be preferable to freeze certain parts of the original model such as the low-level layers extracting generic features, and retraining only the top layers. Alternatively, in order to not tamper with the existing features, all weights of the model can be frozen, and only the biases attached to features at each layer are adjusted. Importantly we also need to avoid the scenario where the top-layer bias would simply shift to the new reference value thereby preventing any meaningful change in the explanation. If this happens, we recommend freezing the top-layer biases to their original value and only learning biases in the layers below.

Constrain the parameters to avoid trivial solutions.

In any case, we recommend to closely monitor model retraining to ensure that the imposed constraints allow for a satisfactory solution of the adjusted regression problem. A further degree of freedom is the selection of the training data. If interested in a particular reference value, we recommend excluding training data points with target values too far from the selected reference value $\widetilde{y}$, by choosing appropriate band parameters τ^+ and τ^-. This last recommendation however holds only to the extent that generalization performance of the model is not penalized by such exclusion. Lastly, model training and retraining is a complex process with many degrees of freedom and tools for analyzing success. Here, general guidance on training neural networks or other nonlinear models remains applicable (e.g. [31], [49], [13], [51], [45]).

D. Implementation and sanity checks

In practice, because the methods we propose for explaining regression models do not require a change of the explanation technique but only a modification of the function or of the architecture implementing such function, existing XAI frameworks remain applicable. Software framework such as Captum[8], iNNvestigate [2][9], DeepExplain[10] and Zennit [5][11] implement a number of popular explanation methods. Some of them, such as iNNvestigate, come with an implementation of LRP.

Whether the practitioners use a third party implementation or their own implementation, there are a number of sanity checks (or unit tests) that can be applied to verify that the procedure satisfies some basic necessary conditions. The most important one is *conservation*, in particular, on needs to verify that $\sum_i R_i = f(\boldsymbol{x}) - \widetilde{y}$. If the network has biases and one can consequently not expect the explanation to be fully conservative, the sanity check can still be applied on a version of the network where the biases are turned to zero. Other sanity checks are specific to the method. For example, one can verify that LRP explanations reduce to special (and easy to implement) cases, for particular choices of LRP hyperparameters. For example, we can verify that an

[8] https://captum.ai/

[9] https://github.com/albermax/innvestigate

[10] https://github.com/marcoancona/DeepExplain

[11] https://github.com/chr5tphr/zennit

implementation consisting of LRP-ϵ and LRP-γ rules reduces to Gradient × Input if setting the hyperparameters ϵ and γ to zero.

VIII. Conclusions & Outlook

Explainable AI has demonstrated great potential in making the decision of black-box models transparent to the user. Recently, methods have been proposed to scale explanations to highly complex neural network classifiers composed of millions of neurons. ML is, however, a vast field, with many formulations of the learning problem that address the numerous application scenarios encountered in practice, including unsupervised learning, supervised learning, reinforcement learning, and within supervised learning, subcategories such as classification or regression or ranking.

In this paper, we found that explanation methods that are based on a conservation principle (e.g. Shapley values, integrated gradients, or LRP) are particularly favorable in the regression scenario as they allow for a decomposition of the predicted quantity on the input features that preserves an interpretation in the same measurement units as the prediction tasks (e.g. units of energy or monetary units). However, we have also demonstrated that Explainable AI cannot simply be transferred between different types of ML problems without adaptation. In particular, an out-of-the-box deployment of explanation techniques for classification omits the fact that a real-valued prediction must often be explained with respect to a particular reference value for the explanation to be meaningful.

We have proposed a collection of techniques for incorporating such reference values, that addresses the various constraints of the explanation method or the model used for prediction. In particular, we have contributed a lightweight restructuring approach, that enables an accurate incorporation of the reference value into the explanation without having to retrain the model nor having to evaluate the function multiple times. This restructuring approach is conceptually related to the ‘neuralization’ approach [36], [37] which was proposed to extend XAI methods from supervised to unsupervised learning, but this time it was used for extending from classification to regression.

After verifying the performance of our approaches on a set of benchmark experiments, we have demonstrated concrete practical use cases on image data as well as on molecular data from atomistic simulations, where chemically plausible explanation of molecular energy could be produced.

As our paper contributes a first step toward extending XAI to regression in a systematic and theoretically founded manner, further work is needed to address other specificities of the regression task over simple classification, such as model uncertainty which is typically decoupled in the regression case from the prediction itself. Ways to combine the prediction and the uncertainty of the prediction into a single explanation remains an open question.

Furthermore, while we have demonstrated empirically a close correspondence between our explanations and a putative reference explanation based on the Shapley value, thereby demonstrating the faithfulness of our method, it will be necessary to extend the evaluation to account for the overall benefit to the user. This could take the form of user experiments, which typically take into consideration whether an attribution on input features is helpful to the user, or whether more structured explanations would be desirable. Such explanations could for example be based on extracted interpretable latent variables, which would allow to not only answer *what* input features are important, but also *why* they are important for a given prediction.

Finally, to further assess the overall practical benefits of using explanations, it will be important to consider end-to-end scenarios [25], where one can precisely measure the utility gain of incorporating explanations into a given learning system, over not incorporating them, in particular, how much a user can objectively gain from using an explanation expressed in the same measurement units as the prediction, and from the additional level of granularity offered by choosing a particular reference value. Also related to the question of evaluation is the level of degradation that can be expected if allowing for adversarial entities into the prediction or explanation process [24], and what countermeasure can be provided to such scenario. Lastly, manageability and energy efficiency aspects such as the cost of implementing, maintaining, and running XAI approaches in an environment of increasingly complex ML models, should also be taken into account in the overall assessment.

Acknowledgements

This work was partly funded by the German Ministry for Education and Research (under refs 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18056A, 01IS18025A and 01IS18037A), the German Research Foundation (DFG) as Math+: Berlin Mathematics Research Center (EXC 2046/1, project-ID: 390685689), the Investitionsbank Berlin under contract No. 10174498 (Pro FIT programme), and the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 965221. Furthermore KRM was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea Government (No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University). Correspondence to WS, KRM and GM.

References

[1] A. Abdolrashidi, M. Minaei, E. Azimi, and S. Minaee. Age and gender prediction from face images using attentional convolutional network. *arXiv preprint arXiv:2010.03791*, 2020.

[2] M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. T. Schütt, G. Montavon, W. Samek, K.-R. Müller, S. Dähne, and P. Kindermans. innvestigate neural networks! *J. Mach. Learn. Res.*, 20:93:1–93:8, 2019.

[3] M. Ancona, E. Ceolini, C. Öztireli, and M. H. Gross. Gradient-based attribution methods. In *Explainable AI*, volume 11700 of *Lecture Notes in Computer Science*, pages 169–191. Springer, 2019.

[4] M. Ancona, C. Öztireli, and M. H. Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 272–281. PMLR, 2019.

[5] C. J. Anders, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin. Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *arXiv preprint arXiv:2106.13200*, 2021.

[6] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022.

[7] R. Angulu, J. R. Tapamo, and A. O. Adewumi. Age estimation via face images: a survey. *EURASIP Journal on Image and Video Processing*, 2018(1):1–35, 2018.

[8] L. Arras, J. A. Arjona-Medina, M. Widrich, G. Montavon, M. Gillhofer, K.-R. Müller, S. Hochreiter, and W. Samek. Explaining and interpreting lstms. In *Explainable AI*, volume 11700 of *Lecture Notes in Computer Science*, pages 211–238. Springer, 2019.

[9] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020.

[10] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015.

[11] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[12] A. Binder, M. Bockmayr, M. Hägele, S. Wienert, D. Heim, K. Hellweg, M. Ishii, A. Stenzinger, A. Hocke, C. Denkert, et al. Morphological and molecular breast cancer profiling through explainable machine learning. *Nature Machine Intelligence*, 3:355–366, 2021.

[13] M. L. Braun, J. M. Buhmann, and K.-R. Müller. On relevant dimensions in kernel feature spaces. *The Journal of Machine Learning Research*, 9:1875–1908, 2008.

[14] W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*. OpenReview.net, 2019.

[15] K. Bykov, M. M.-C. Höhne, K.-R. Müller, S. Nakajima, and M. Kloft. How much can I trust you?–quantifying uncertainties in explaining neural networks. *arXiv preprint arXiv:2006.09000*, 2020.

[16] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. *Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission*, page 1721–1730. 2015.

[17] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 882–891. PMLR, 2018.

[18] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

[19] S. Cho, W. Chang, G. Lee, and J. Choi. Interpreting internal activation patterns in deep temporal neural networks by finding prototypes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 158–166, New York, NY, USA, 2021. Association for Computing Machinery.

[20] I. Covert, S. Lundberg, and S.-I. Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

[21] T. Cui, P. Marttinen, and S. Kaski. Recovering pairwise interactions using neural networks. *arXiv preprint arXiv:1901.08361*, 2019.

[22] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. A survey of the state of explainable AI for natural language processing. In *AACL/IJCNLP*, pages 447–459. Association for Computational Linguistics, 2020.

[23] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009.

[24] A. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, pages 13567–13578, 2019.

[25] F. Doshi-Velez and B. Kim. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017.

[26] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani, and G. Montavon. Building and interpreting deep similarity models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 10.1109/TPAMI.2020.3020738, 2020.

[27] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7):620–631, May 2021.

[28] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, pages 3449–3457. IEEE Computer Society, 2017.

[29] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1 – 67, 1991.

[30] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[31] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[32] M. Grabisch and M. Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999.

[33] S. S. Hada and M. Á. Carreira-Perpinán. Exploring counterfactual explanations for classification and regression trees. In *XKDD (ECML PKDD International Workshop)*, 2021.

[34] H. Han, C. Otto, and A. K. Jain. Age estimation from face images: Human vs. machine performance. In *2013 international conference on biometrics (ICB)*, pages 1–8. IEEE, 2013.

[35] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer. Towards multi-modal causability with graph neural networks enabling information fusion for explainable ai. *Information Fusion*, 71:28–37, 2021.

[36] J. R. Kauffmann, M. Esders, G. Montavon, W. Samek, and K.-R. Müller. From clustering to cluster explanations via neural networks. *arXiv preprint arXiv:1906.07633*, 2019.

[37] J. R. Kauffmann, K.-R. Müller, and G. Montavon. Towards explaining anomalies: A deep taylor decomposition of one-class models. *Pattern Recognit.*, 101:107198, 2020.

[38] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chemical Reviews*, 121(16):9816—-9872, 2021.

[39] B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[40] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[41] J. Klicpera, J. Groß, and S. Günnemann. Directional message passing for molecular graphs. 2020.

[42] F. Kratzert, M. Herrnegger, D. Klotz, S. Hochreiter, and G. Klambauer. Neuralhydrology - interpreting lstms in hydrology. In *Explainable AI*, volume 11700 of *Lecture Notes in Computer Science*, pages 347–362. Springer, 2019.

[43] S. Lapuschkin, A. Binder, K.-R. Müller, and W. Samek. Understanding and comparing deep neural networks for age and gender classification. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1629–1638, 2017.

[44] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10:1096, 2019.

[45] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[46] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 2021.

[47] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[48] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *NIPS*, pages 4765–4774, 2017.

[49] G. Montavon. Introduction to neural networks. In *Machine Learning Meets Quantum Physics*, pages 37–62. Springer International Publishing, 2020.

[50] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. Layer-wise relevance propagation: An overview. In *Explainable AI*, volume 11700 of *Lecture Notes in Computer Science*, pages 193–209. Springer, 2019.

[51] G. Montavon, M. L. Braun, and K.-R. Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(9):2563–2581, 2011.

[52] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[53] A. M. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NIPS*, pages 3387–3395, 2016.

[54] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi. Machine learning for molecular simulation. *Annu. Rev. Phys. Chem.*, 71:361–390, 2020.

[55] S. Papadopoulos and C. E. Kontokosta. Grading buildings on energy performance using city benchmarking data. *Applied Energy*, 233:244–253, 2019.

[56] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144. ACM, 2016.

[57] T. Rojat, R. Puget, D. Filliat, J. D. Ser, R. Gelin, and N. D. Rodríguez. Explainable artificial intelligence (XAI) on timeseries data: A survey. *CoRR*, abs/2104.00950, 2021.

[58] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[59] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019.

[60] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

[61] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, 2012.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[63] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017.

[64] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[65] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, editors. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*. Springer, 2019.

[66] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page doi: 10.1109/TPAMI.2021.3115452, 2021.

[67] K. Schütt, P. Kessel, M. Gastegger, K. Nicoli, A. Tkatchenko, and K.-R. Müller. Schnetpack: A deep learning toolbox for atomistic systems. *Journal of Chemical Theory and Computation*, 15(1):448–455, 2018.

[68] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[69] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8:13890, 2017.

[70] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[71] L. S. Shapley. *A Value for n-Person Games*, pages 307–318. Princeton University Press, 1953.

[72] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR, 2017.

[73] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[74] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[75] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[76] T. Spooner, D. Dervovic, J. Long, J. Shepard, J. Chen, and D. Magazzeni. Counterfactual explanations for arbitrary regression models, 2021.

[77] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (Workshop)*, 2015.

[78] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, 2010.

[79] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.

[80] H. R. Tamaddon-Jahromi, N. K. Chakshu, I. Sazonov, L. M. Evans, H. Thomas, and P. Nithiarasu. Data-driven inverse modelling through neural network (deep learning) and computational heat transfer. *Computer Methods in Applied Mechanics and Engineering*, 369:113217, 2020.

[81] M. Tsang, D. Cheng, and Y. Liu. Detecting statistical interactions from neural network weights. In *6th International Conference on Learning Representations*, 2018.

[82] O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller. Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature Communications*, (12):7273, 2021.

[83] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller. Machine learning force fields. *Chemical Reviews*, 121(16):10142—10186, 2021.

[84] O. T. Unke and M. Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of Chemical Theory and Computation*, 15(6):3678–3693, 2019. PMID: 31042390.

[85] O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.*, 4:347–358, 2020.

[86] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: automated decisions and the gdpr. *Harvard Journal of Law and Technology*, 31(2):841–887, 2018.

[87] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[88] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, pages 9240–9251, 2019.

[89] J. Yoon, J. Jordon, and M. van der Schaar. INVASE: instance-wise variable selection using neural networks. In *ICLR (Poster)*. OpenReview.net, 2019.

[90] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV (1)*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014.

[91] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.*, 126(10):1084–1102, 2018.

[92] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929. IEEE Computer Society, 2016.