Building and Interpreting Deep Similarity Models

Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, Grégoire Montavon

Abstract—Many learning algorithms such as kernel machines, nearest neighbors, clustering, or anomaly detection, are based on distances or similarities. Before similarities are used for training an actual machine learning model, we would like to verify that they are bound to meaningful patterns in the data. In this paper, we propose to make similarities interpretable by augmenting them with an *explanation*. We develop BiLRP, a scalable and theoretically founded method to systematically decompose the output of an already trained deep similarity model on pairs of input features. Our method can be expressed as a composition of LRP explanations, which were shown in previous works to scale to highly nonlinear models. Through an extensive set of experiments, we demonstrate that BiLRP robustly explains complex similarity models, e.g. built on VGG-16 deep neural network features. Additionally, we apply our method to an open problem in digital humanities: detailed assessment of similarity between historical documents such as astronomical tables. Here again, BiLRP provides insight and brings verifiability into a highly engineered and problem-specific similarity model.

Index Terms—Similarity, layer-wise relevance propagation, deep neural networks, explainable machine learning, digital humanities.

1 Introduction

Building meaningful similarity models that incorporate prior knowledge about the data and the task is an important area of machine learning and information retrieval [1], [2]. Good similarity models are needed to find relevant items in databases [3], [4], [5]. Similarities (or kernels) are also the starting point of a large number of machine learning models including discriminative learning [6], [7], unsupervised learning [8], [9], [10], [11], and data embedding/visualization [12], [13], [14].

An important practical question is how to select the similarity model appropriately. Assembling a labeled dataset of similarities for validation can be difficult: The labeler would need to inspect meticulously multiple pairs of data points and come up with exact real-valued similarity scores. As an alternative, selecting a similarity model based on performance on some proxy task can be convenient (e.g. [15], [16], [17], [18]). In both cases, however, the selection procedure is exposed to a potential lack of representativity of the training data (cf. the 'Clever Hans' effect [19]).—In this paper, we aim for a more direct way to assess similarity models, and make use of explainable ML for that purpose.

Explainable ML [20], [21], [22] is a subfield of machine learning that focuses on making predictions interpretable for the human. Numerous methods have been proposed in the context of ML classifiers [23], [24], [25], [26]. For example, layer-wise relevance propagation (LRP) [24] explains the prediction of a neural network classifier by performing a backward pass in the network, which results in an attribution of the prediction to the different input features.

In this paper, we bring explainable ML to similarity. We consider similarity models of the type:

$$y(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x}) \, , \, \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x}') \rangle,$$

e.g. dot products built on some hidden layer of a deep neural network. We assume the similarity model to be already trained. Explanation techniques developed in the context of classifiers (e.g. [24], [25]) cannot be directly applied, because they often assume some form of local *linearity* whereas dot products have *bilinearity*. Hence, we propose a method for explanation that adapts to this new setting.

Our method which we call 'BiLRP' is illustrated in Fig. 1. BiLRP explanations can be produced in three steps:

- *Step 1:* Feed a pair of inputs to the neural network to compute the feature representations.
- *Step 2:* Compute an LRP explanation for each dimension of the two feature representations.
- *Step 3:* Apply an outer product between the two collections of LRP explanations.

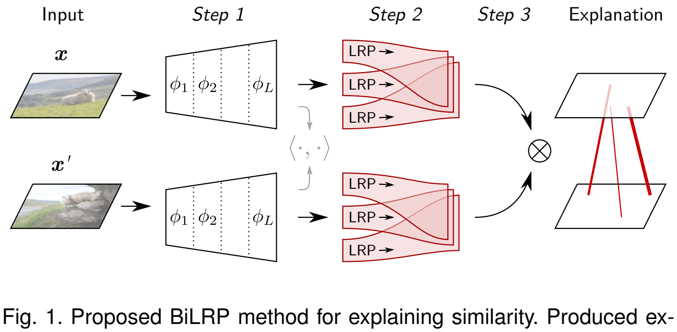


Fig. 1. Proposed BiLRP method for explaining similarity. Produced explanations are in terms of *pairs* of input features.

- O. Eberle is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany.
- J. Büttner is with the Max Planck Institute for the History of Science, 14159 Berlin, Germany.
- F. Kräutli is with the Max Planck Institute for the History of Science, 14159 Berlin, Germany.
- K.-R. Müller is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany; the Department of Brain and Cognitive Engineering, Korea University, Seoul 136-713, Korea; and the Max Planck Institut für Informatik, 66123 Saarbrücken, Germany. E-mail: klaus-robert.mueller@tu-berlin.de.
- M. Valleriani is with the Max Planck Institute for the History of Science, 14159 Berlin, Germany.
- G. Montavon is with the Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany. E-mail: gregoire.montavon@tu-berlin.de.

(Corresponding Authors: Grégoire Montavon, Klaus-Robert Müller)

The output of BiLRP is an attribution of the predicted similarity score to the *pairs* of input features (e.g. pixels) of the two inputs.

BiLRP can be embedded in the theoretical framework of deep Taylor decomposition [27]. Specifically, the procedure can be expressed as a collection of second-order Taylor expansions performed in each layer. Elements of these expansions identify the exact layer-wise redistribution strategy. BiLRP can also be interpreted as building layer after a layer a robustified Hessian of the similarity model, that lets us extract meaningful explanations, even when the similarity is built on complex deep neural networks.

We apply BiLRP on similarity models built at various layers of the well-established VGG-16 image classification network [28]. Our explanation method brings useful insights into the strengths and limitations of each similarity model. We also illustrate how the insights brought by BiLRP can be actioned to produce an improved similarity model. We then move to an open problem in the digital humanities, where similarity between scanned astronomical tables needs to be assessed [29]. We build a highly engineered similarity model that is specialized for this task. Again BiLRP proves useful by being able to inspect the similarity model and validate it from limited data.

Altogether, the method we propose brings transparency into a key ingredient of machine learning: similarity. Our contribution paves the way for the systematic design and validation of similarity-based ML models in an efficient, fully informed, and human-interpretable manner.

1.1 Related Work

Methods such as LLE [30], diffusion maps [31], or t-SNE [14] give insight into the similarity structure of large datasets by embedding data points in a low-dimensional subspace where relevant similarities are preserved. While these methods provide useful visualization, their purpose is more to find *global* coordinates to comprehend a whole dataset, than to explain why two *individual* data points are predicted to be similar.

The question of explaining individual predictions has been extensively studied in the context of ML classifiers. Methods based on occlusions [32], [33], surrogate functions [25], [34], gradients [23], [35], [36], [37], or reverse propagation [24], [32], have been proposed, and are capable of highlighting the most relevant features. Some approaches have been extended to unsupervised models, e.g. anomaly detection [38], [39] and clustering [40], and attention models have also been developed to explain tasks different from classification such as image captioning [41] or similarity [42]. Our work goes further along this direction and explains similarity built on general neural network models, and by identifying relevant *pairs* of input features.

Several methods for joint features explanations have been proposed. Some of them extract feature interactions globally [43], [44]. Other methods produce individual explanations for simple pairwise matching models applied on the input features [45], or some activations maps of a convolution network [46]. Another method incorporates explicit multivariate structures into the model to identify joint contributions [47]. Another method extracts joint feature explanations in nonlinear models by estimating the integral of the Hessian [48]. In comparison, our BiLRP method leverages the deep layered structure of the model to robustly explain predicted similarity in terms of input features.

A number of works improve similarity models by leveraging prior knowledge or ground truth labels. Proposed approaches include structured kernels [49], [1], [50], [51], or siamese/triplet networks [52], [53], [54], [55], [56]. Beyond similarity, applications such as collaborative filtering [57], transformation modeling [58], and information retrieval [59], also rely on building high-quality matching models between pairs of data.—Our work has an orthogonal objective: It assumes an already trained well-performing similarity model, and makes it explainable to enhance its verifiability and to extract novel insights from it.

2 Towards Explaining Similarity

In this section, we present basic approaches to explain the predictions of a similarity model in terms of input features. The similarity model is considered to be already trained. We first discuss the case of a simple linear model, and then extend the concept to more general nonlinear cases.

Let us begin with a simple scenario where $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and the similarity score is given by some dot product $y(\boldsymbol{x}, \boldsymbol{x}') = \langle W\boldsymbol{x}, W\boldsymbol{x}' \rangle$, with W a projection matrix of size $h \times d$. The similarity score is bilinear with $(\boldsymbol{x}, \boldsymbol{x}')$. This score can be naturally attributed to pairs of input features (i, i') by rewriting it as the sum:

$$y(\boldsymbol{x}, \boldsymbol{x}') = \textstyle\sum_{ii'} \langle W_{:,i}, W_{:,i'} \rangle \cdot x_i x'_{i'}$$

and identifying the elements of the sum as the respective contributions. Clearly, input features interact to produce a high/low similarity score.

In practice, more accurate models of similarity can be obtained by relaxing the linearity constraint. Consider some similarity model $y(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ built on some abstract feature map $\phi \colon \mathbb{R}^d \to \mathbb{R}^h$ which we assume to be differentiable. A simple and general way of attributing the similarity score to the input features is to compute a Taylor expansion [24] at some reference point $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')$:

$$\begin{aligned} y(\boldsymbol{x}, \boldsymbol{x}') = y(&\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}') \\ &+ \textstyle\sum_i \, [\nabla y(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')]_i \, (x_i - \widetilde{x}_i) \\ &\quad + \textstyle\sum_{i'} \, [\nabla y(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')]_{i'} \, (x'_{i'} - \widetilde{x}'_{i'}) \\ &\qquad + \textstyle\sum_{ii'} \, [\nabla^2 y(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')]_{ii'} \, (x_i - \widetilde{x}_i) \, (x'_{i'} - \widetilde{x}'_{i'}) \\ &\qquad + \dots \end{aligned}$$

Here, ∇^2 denotes the Hessian. The explanation is then obtained by identifying the multiple terms of the expansion. Like for the linear case, some of these terms can be attributed to pairs of features (i, i').

For special choices of functions, namely when ϕ is a piecewise linear positive homogeneous function, we find that choosing the reference point $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}') = \delta \cdot (\boldsymbol{x}, \boldsymbol{x}')$ with δ almost zero leads to a simplified 'Hessian × Product' formulation:

$$y(\boldsymbol{x}, \boldsymbol{x}') = \textstyle\sum_{ii'} \, [\nabla^2 y(\boldsymbol{x}, \boldsymbol{x}')]_{ii'} \, x_i \, x'_{i'} \qquad (1)$$

where second-order contributions can be easily computed. This simple method we contribute will serve as a baseline in the experiments.

A limitation of methods relying on the model derivatives is that these derivatives can be noisy, especially when the function to analyze is a deep neural network. Derivative noise has been observed e.g. in [60], [22].

3 Explaining Similarity with BiLRP

In the following, we introduce our new BiLRP method for explaining similarities. It is based on merging the following two ideas:

1) Second-order Taylor expansions for producing explanations in terms of pairs of input features, as described in Section 2,
2) The layer-wise relevance propagation (LRP) [24] technique that robustly explains complex deep neural network predictions.

BiLRP assumes as a starting point that the similarity score is structured as a dot product over features of a neural network:

$$y(\boldsymbol{x},\boldsymbol{x}') = \big\langle \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x})\,,\, \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x}')\big\rangle.$$

The functions $\phi_1,\dots,\phi_L$ are the different layers of the network and can either be linear/ReLU layers, or more general positively homogeneous functions. (The same network can also be written as a single network $y(\boldsymbol{x},\boldsymbol{x}') = \psi_L \circ \cdots \circ \psi_1(\boldsymbol{x},\boldsymbol{x}')$ where ψ subsumes the two branches of the computation.) Then, inspired by LRP, the BiLRP method applies a purposely designed message passing procedure from the top layer where the similarity score is produced to the input layer where the explanation is formed. However, unlike standard LRP, BiLRP sends messages between *pairs* of neurons that jointly contribute to the similarity score.

The presentation of BiLRP is structured as follows: Section 3.1 explains how the messages to propagate are obtained from second-order Taylor expansions. Section 3.2 discusses theoretical properties of BiLRP and how the method can be interpreted as building a robustified Hessian of the similarity model. Finally, Section 3.3 shows how BiLRP can be computed in a way that makes use of LRP as an inner computation, thereby considerably easing implementation.

3.1 Extracting BiLRP Propagation Rules

To build meaningful propagation rules, we make use of the 'deep Taylor decomposition' (DTD) [27] framework. DTD consists of applying Taylor expansions at each layer to identify the way the prediction must be redistributed to the layer below.

Assume we have already run a few steps of propagation starting from the output until some intermediate layer of the network. At this stage, we have an attribution of the similarity score on pairs of neurons at this layer. Let $R_{kk'}$ be a 'relevance score' that measures the share of similarity that has been attributed to the pair of neurons (k,k') at this layer.

In the DTD framework, this quantity is first expressed as a function of the vector of activations $\boldsymbol{a}$ in the layer below.

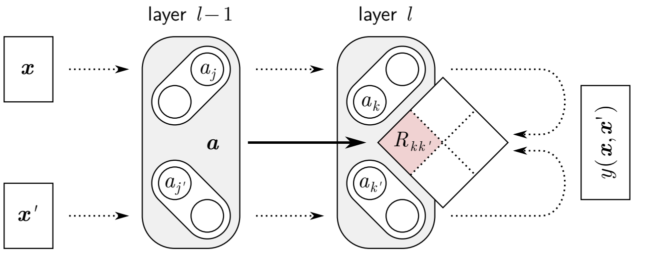


Fig. 2. Diagram of the map used by DTD to derive BiLRP propagation rules. The map connects activations at some layer to relevance in the layer above.

The relation between these two quantities is depicted in Fig. 2. Then, DTD seeks to perform a Taylor expansion of the function $R_{kk'}(\boldsymbol{a})$ at some reference point $\widetilde{\boldsymbol{a}}$:

$$\begin{aligned} R_{kk'}(\boldsymbol{a}) = R_{kk'}(\widetilde{\boldsymbol{a}}) & \\ &+ \textstyle\sum_j [\nabla R_{kk'}(\widetilde{\boldsymbol{a}})]_j \cdot (a_j - \widetilde{a}_j) \\ &\quad + \textstyle\sum_{j'} [\nabla R_{kk'}(\widetilde{\boldsymbol{a}})]_{j'} \cdot (a_{j'} - \widetilde{a}_{j'}) \\ &\qquad + \textstyle\sum_{jj'} [\nabla^2 R_{kk'}(\widetilde{\boldsymbol{a}})]_{jj'} \cdot (a_j - \widetilde{a}_j)\,(a_{j'} - \widetilde{a}_{j'}) \\ &\qquad\quad + \dots \end{aligned}$$

so that messages $R_{jj' \leftarrow kk'}$ can be identified. In practice, the function $R_{kk'}(\boldsymbol{a})$ is difficult to analyze, because it subsumes a potentially large number of forward and backward computations. Therefore, DTD introduces the concept of a 'relevance model' $\widehat{R}_{kk'}(\boldsymbol{a})$ which locally approximates the true function $R_{kk'}(\boldsymbol{a})$, but only depends on the neighboring parameters and activations [27]. For linear/ReLU layers [61], we define the relevance model:

$$\widehat{R}_{kk'}(\boldsymbol{a}) = \underbrace{\big(\textstyle\sum_j a_j w_{jk}\big)^+}_{a_k} \underbrace{\big(\textstyle\sum_{j'} a_{j'} w_{j'k'}\big)^+}_{a_{k'}} c_{kk'}$$

with $c_{kk'}$ a constant set in a way that $\widehat{R}_{kk'}(\boldsymbol{a}) = R_{kk'}$. (This relevance model is justified later in Proposition 3.) We now have an easily analyzable model, more specifically, a model that is bilinear on the joint activated domain and zero elsewhere. We search for a root point $\widetilde{\boldsymbol{a}}$ at the intersection between the two ReLU hinges and the plane $\{\widetilde{\boldsymbol{a}}(t,t') \mid t,t' \in \mathbb{R}\}$ where:

$$\begin{aligned} [\,\widetilde{\boldsymbol{a}}(t,t')\,]_j &= a_j - t a_j \cdot (1 + \gamma \cdot 1_{w_{jk}>0}), \\ [\,\widetilde{\boldsymbol{a}}(t,t')\,]_{j'} &= a_{j'} - t' a_{j'} \cdot (1 + \gamma \cdot 1_{w_{j'k'}>0}) \end{aligned}$$

with $\gamma \geq 0$ a hyperparameter. This search strategy can be understood as starting with the activations $\boldsymbol{a}$, and jointly decreasing them (especially the ones with positive contributions) until $\widehat{R}_{kk'}(\widetilde{\boldsymbol{a}})$ becomes zero. Zero- and first-order terms of the Taylor expansion vanish, leaving us with the interaction terms $R_{jj' \leftarrow kk'}$. Rewriting the interaction terms in closed form and aggregating messages coming from the layer above (i.e. $R_{jj'} = \sum_{kk'} R_{jj' \leftarrow kk'}$), we get the propagation rule:

$$R_{jj'} = \sum_{kk'} \frac{a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} R_{kk'} \qquad (2)$$

with $\rho(w_{jk}) = w_{jk} + \gamma w_{jk}^+$. A derivation is given in Appendix A.1 of the Supplement. This propagation rule can be seen as a second-order variant of the LRP-γ rule [62] used for explaining DNN classifiers. It has the following interpretation: A pair of neurons (j, j') is assigned relevance if the following three conditions are met:

(i) it jointly activates,
(ii) some pairs of neurons in the layer above jointly react,
(iii) these reacting pairs are themselves relevant.

In addition to linear/ReLU layers, we would like BiLRP to handle other common layers such as max-pooling and min-pooling. These two layer types can be seen as special cases of the broader class of *positively homogeneous* layers (i.e. satisfying $\forall_{\boldsymbol{a}} \forall_{t>0}: \ a_k(t\boldsymbol{a}) = t a_k(\boldsymbol{a})$). For these layers, the following propagation rule can be derived from DTD:

$$R_{jj'} = \sum_{kk'} \frac{a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}}{\sum_{jj'} a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}} R_{kk'} \qquad (3)$$

(cf. Appendix A.2 of the Supplement). This propagation rule has a similar interpretation to the one above, in particular, it also requires for (j, j') to be relevant that the corresponding neurons activate, that some neurons (k, k') in the layer above jointly react, and that the latter neurons are themselves relevant.

3.2 Theoretical Properties of BiLRP

A number of results can be shown about BiLRP. A first result relates the produced explanation to the predicted similarity. Another result lets us view the Hessian × Product method as a special case of BiLRP. A last result provides a justification for the relevance models used in Section 3.1.

Proposition 1. *For deep rectifier networks with zero biases, BiLRP is conservative, i.e.* $\sum_{ii'} R_{ii'} = y(\boldsymbol{x}, \boldsymbol{x}')$.

(See Appendix B.1 of the Supplement for a proof.) Conservation ensures that relevance scores are in proportion to the output of the similarity model.

Proposition 2. *When* $\gamma = 0$*, explanations produced by BiLRP reduce to those of Hessian × Product.*

(See Appendix B.2 of the Supplement for a proof.) The proof relies on the fact that relevance scores in linear/ReLU layers can also be expressed as $R_{jj'} = a_j a_{j'} c_{jj'}$ and $R_{kk'} = a_k a_{k'} c_{kk'}$ with

$$c_{jj'} = \sum_{kk'} (w_{jk} + \gamma w_{jk}^+) \cdot (w_{j'k} + \gamma w_{j'k}^+) \cdot \frac{a_k}{z_k} \frac{a_{k'}}{z_{k'}} c_{kk'} \qquad (4)$$

where $z_k = \sum_j a_j (w_{jk} + \gamma w_{jk}^+)$ and similarly for $z_{k'}$. For the special case $\gamma = 0$, the terms a_k/z_k and $a_{k'}/z_{k'}$ become equivalent to ReLU derivatives, and this makes Eq. (4) coincide with the equation for propagating second-order derivatives which is used to compute the Hessian. This theoretical connection also hints at a more robust behavior of BiLRP when $\gamma > 0$: In this case the discontinuity of the ReLU derivative disappears, and the propagation procedure can consequently also be interpreted as building a robustified Hessian of the similarity model. We demonstrate empirically in Sections 4 and 5 that non-zero values of γ give better explanations.

Proposition 3. *The relevance computed by BiLRP at each layer can be rewritten as* $R_{jj'} = a_j a_{j'} c_{jj'}$*, where* $c_{jj'}$ *is locally approximately constant.*

(Cf. Appendix B.3 of the Supplement.) This property supports the modeling of $c_{jj'}, c_{kk'}, \dots$ as constant, leading to easily analyzable relevance models from which the BiLRP propagation rules of Section 3.1 can be derived.

3.3 BiLRP as a Composition of LRP Computations

A limitation of a plain application of the propagation rules of Section 3.1 is that we need to handle at each layer a number of relevance scores which grows quadratically with the number of neurons. Consequently, for large neural networks, a direct computation of these propagation rules is unfeasible. However, it can be shown that relevance scores at each layer can also written in the factored form:

$$R_{kk'} = \textstyle\sum_{m=1}^{h} R_{km} R_{k'm}$$
$$R_{jj'} = \textstyle\sum_{m=1}^{h} R_{jm} R_{j'm}$$

where h is the dimension of the top-layer feature map, and where the factors can be computed iteratively as:

$$R_{jm} = \sum_k \frac{a_j \rho(w_{jk})}{\sum_j a_j \rho(w_{jk})} R_{km} \qquad (5)$$

for linear/ReLU layers, and

$$R_{jm} = \sum_k \frac{a_j [\nabla a_k]_j}{\sum_j a_j [\nabla a_k]_j} R_{km} \qquad (6)$$

for positively homogeneous layers. The relevance scores that result from applying these factored computations are strictly equivalent to those one would get if using the original propagation rules of Section 3.1. A proof is given in Appendix C of the Supplement.

Furthermore, in comparison to the $(\#\,\text{neurons})^2$ computations required at each layer by the original propagation rules, the factored formulation only requires $(\#\,\text{neurons} \times 2h)$ computations. The factored form is therefore especially advantageous when h is low. In the experiments of Section 5, we will improve the explanation runtime of our similarity models by adding an extra layer projecting output activations to a smaller number of dimensions.

Lastly, we observe that Equations (5) and (6) correspond to common rules used by standard LRP. The first one is equivalent to the LRP-γ rule [62] used in convolution/ReLU layers of DNN classifiers. The second one corresponds to the way LRP commonly handles pooling layers [24]. These propagation rules apply independently on each branch and factor of the similarity model. This implies that BiLRP can be implemented as a combination of multiple LRP procedures that are then recombined once the input layer has been reached:

$$\begin{aligned}\text{BiLRP}(y, \boldsymbol{x}, \boldsymbol{x}') = \sum_{m=1}^{h} & \text{LRP}([\phi_L \circ \cdots \circ \phi_1]_m, \boldsymbol{x}) \\ & \otimes \text{LRP}([\phi_L \circ \cdots \circ \phi_1]_m, \boldsymbol{x}')\end{aligned}$$

This modular approach to compute BiLRP explanations is shown graphically in Fig. 3. BiLRP can therefore be easily and efficiently implemented based on existing explanation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3020738, IEEE Transactions on Pattern Analysis and Machine Intelligence

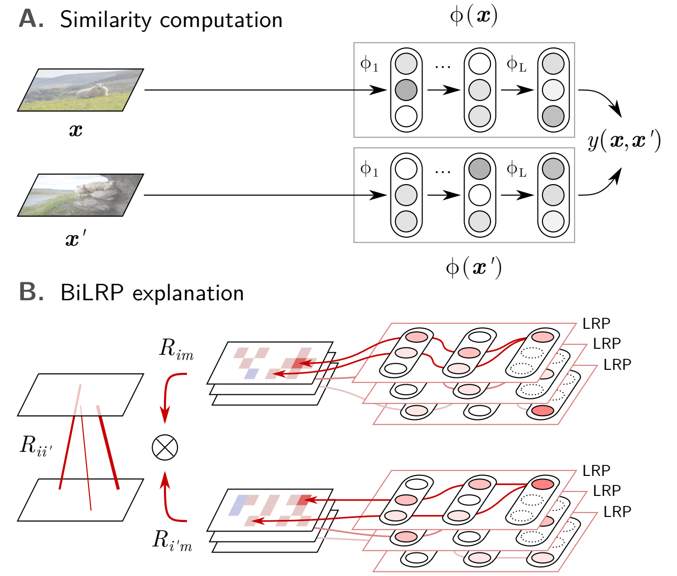


Fig. 3. Illustration of our approach to compute BiLRP explanations: **A.** Input examples are mapped by the neural network up to the layer at which the similarity model is built. **B.** LRP is applied to all individual activations in this layer, and the resulting array of explanations is recombined into a single explanation of predicted similarity.

software. We note that the modular approach described here is not restricted to LRP. Other explanation techniques could in principle be used in the composition. Doing so would however lose the interpretation of the explanation procedure as a deep Taylor decomposition.

4 BiLRP vs. Baselines

This section tests the ability of the proposed BiLRP method to produce faithful explanations. In general, ground-truth explanations of ML predictions, especially nonlinear ones, are hard to acquire [22], [63]. Thus, we consider an *artificial* scenario consisting of:

(i) a hardcoded similarity model from which it is easy to extract ground-truth explanations,
(ii) a neural network trained to reproduce the hardcoded model exactly on the whole input domain.

Because the hardcoded model and the neural network become exact functional copies after training, explanations for their predictions should be the same. Hence, this gives us ground-truth explanations to evaluate BiLRP against baseline methods.

The hardcoded similarity model takes two random sequences of 6 digits as input and counts the number of matches between them. The matches between the two sequences form the ground truth explanation. The neural network is constructed and trained as follows: Each digit forming the sequence is represented as vectors in $\mathbb{R}_+^{10}$. To avoid a too simple task, we set these vectors to be correlated. Vectors associated to the digits in the sequence are then concatenated to form an input $\boldsymbol{x} \in \mathbb{R}_+^{6\times 10}$. The input goes through two hidden layers of size 100 and one top layer of size 50 corresponding to the feature map. We train the network for 10000 iterations of stochastic gradient descent to minimize the mean square error between predictions and ground-truth similarities, and reach an error of 10^{-3}, indicating that the neural network solves the problem perfectly.

Because there is currently no well-established method for explaining similarity, we consider three simple baselines and use them as a benchmark for evaluating BiLRP:

– 'Saliency': $R_{ii'} = (x_i x'_{i'})^2$
– 'Curvature': $R_{ii'} = ([\nabla^2 y(\boldsymbol{x}, \boldsymbol{x}')]_{ii'})^2$
– 'Hessian × Product': $R_{ii'} = x_i x'_{i'}\,[\nabla^2 y(\boldsymbol{x}, \boldsymbol{x}')]_{ii'}$

Each explanation method produces a scoring over all pairs of input features, i.e. a $(6\times 10)\times(6\times 10)$-dimensional explanation. The latter can be pooled over embedding dimensions (cf. Appendix D of the Supplement) to form a 6×6 matrix connecting the digits from the two sequences. Results are shown in Fig. 4. The closer the produced connectivity pattern to the ground truth, the better the explanation method. High scores are shown in red, low scores in light red or white, and negative scores in blue.

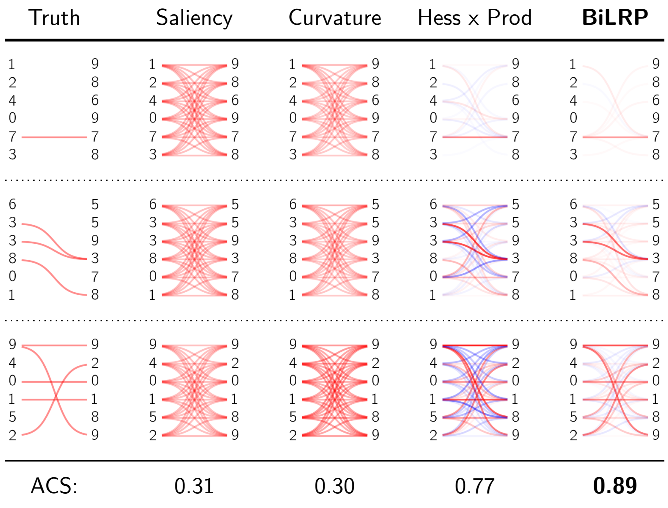


Fig. 4. Benchmark comparison on a toy example where we have ground-truth explanation of similarity. BiLRP performs better than all baselines, as measured by the average cosine similarity to the ground truth.

We observe that the 'Saliency' baseline does not differentiate between matching and non-matching digits. This is explained by the fact that this baseline is not output-dependent and thus does not know the task. The 'Curvature' baseline, although sensitive to the output, does not improve over saliency. The 'Hessian × Product' baseline, which can be seen as a special case of BiLRP with $\gamma = 0$, matches the ground truth more accurately but introduces some spurious negative contributions. BiLRP, through a proper choice of parameter γ (here set to 0.09) considerably reduces these negative contributions.

This visual inspection is validated quantitatively by considering a large number of examples and computing the average cosine similarity (ACS) between the produced explanations and the ground truth. An ACS of 1.0 indicates perfect matching with the ground truth. 'Saliency' and 'Curvature' baselines have low ACS. The accuracy is strongly improved by 'Hessian × Product' and further improved by BiLRP. The effect of the parameter γ of BiLRP on the ACS score is shown in Fig. 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3020738, IEEE Transactions on Pattern Analysis and Machine Intelligence

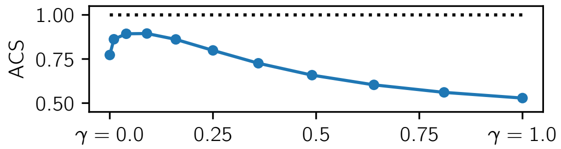


Fig. 5. Effect of the BiLRP parameter γ on the average cosine similarity between the explanations and the ground truth.

We observe that the best parameter γ is small but non-zero. Like for standard LRP, the explanation can be further fine-tuned, e.g. by setting the parameter γ different at each layer or by considering a broader set of LRP propagation rules [64], [62].

5 Interpreting Deep Similarity Models

Our next step will be to use BiLRP to gain insight into practical similarity models built on the well-established VGG-16 convolutional neural network [28]. We take a pretrained version of this network and build the similarity model

$$y(\boldsymbol{x}, \boldsymbol{x}') = \langle \text{VGG}_{:31}(\boldsymbol{x}), \text{VGG}_{:31}(\boldsymbol{x}') \rangle,$$

i.e. a dot product on the neural network activations at layer 31. This layer corresponds to the last layer of features before the classifier. The mapping from input to layer 31 is a sequence of convolution/ReLU layers, and max-pooling layers. It is therefore explainable by BiLRP. However, the large number of dimensions entering in the dot product computation (512 feature maps of size $\frac{w}{32} \times \frac{h}{32}$ where w and h are their dimensions), makes a direct application of BiLRP computationally expensive. To reduce the computation time, we append to the last layer a random projection layer that maps activations to a lower-dimensional subspace. In our experiments, we find that projecting to 100 dimensions provides sufficiently detailed explanations and achieves the desired computational speedup. We set the BiLRP parameter γ to $0.5, 0.25, 0.1, 0.0$ for layers 2–10, 11–17, 18–24, 25–31 respectively. For layer 1, we use the $z^{\mathcal{B}}$-rule, that specifically handles the pixel-domain [27]. Finally, we apply a 8×8 pooling on the output of BiLRP to reduce the size of the explanations.

Figure 6 (A-F) shows our BiLRP explanations on a selection of images pairs taken from the Pascal VOC 2007 dataset [65] and resized to 128×128 pixels. Positive relevance scores are shown in red, negative scores in blue, and score magnitude is represented by opacity. Example A shows two identical images being compared. BiLRP finds that eyes, nose, and ears are the most relevant features to explain similarity. Example B shows two different images of birds. Here, the eyes are again contributing to the high similarity. In Example C, the front part of the two planes are matched.

Examples D and E show cases where the similarity is not attributed to what the user may expect. In Example D, the horse's muzzle is matched to the head of a sheep. In Example E, while we expect the matching to occur between

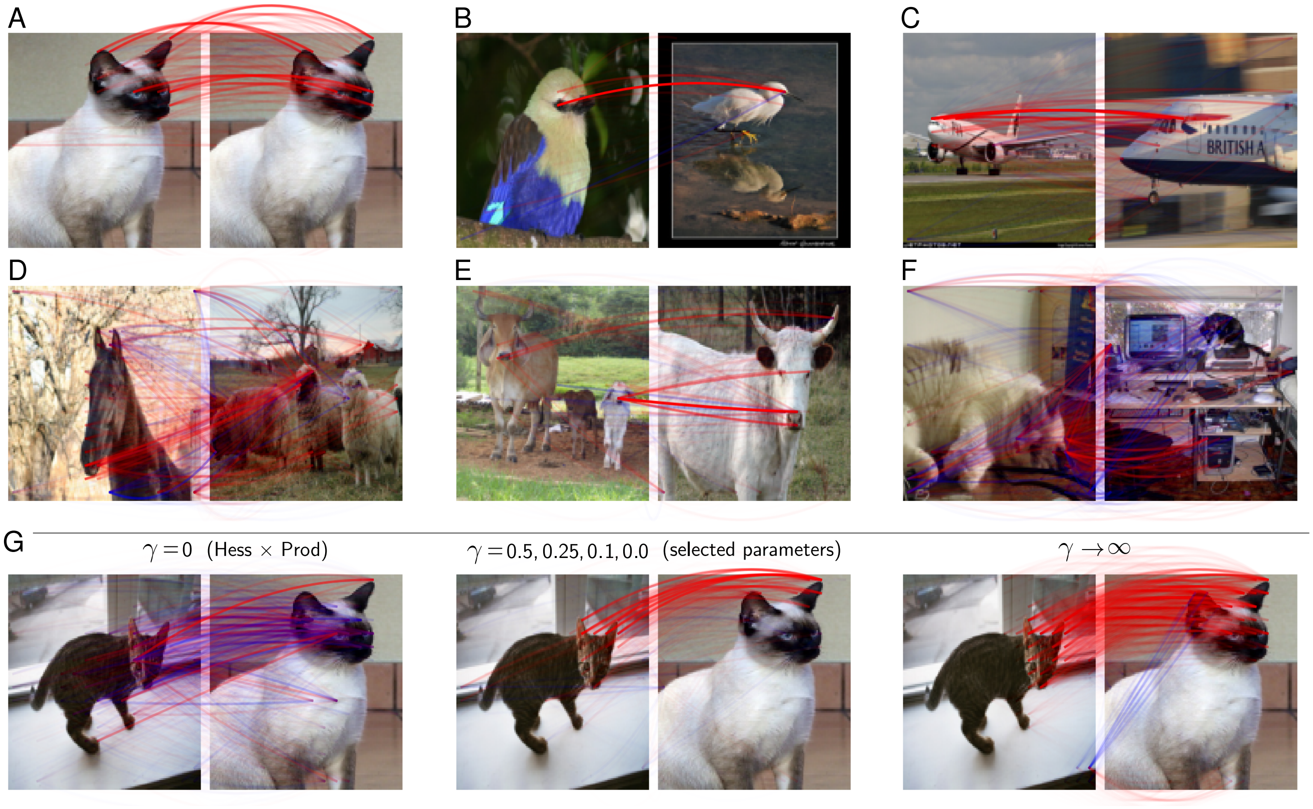

A
B
C
BRITISH A
D
E
F
G
$\gamma = 0$ (Hess × Prod)
$\gamma = 0.5, 0.25, 0.1, 0.0$ (selected parameters)
$\gamma \to \infty$

Fig. 6. Application of BiLRP to a dot-product similarity model built on VGG-16 features at layer 31. *Top:* BiLRP explanations on different pairs of input images from the Pascal VOC 2007 dataset. Red and blue color indicate positive and negative contributions to the similarity. (Details of the rendering procedure are given in Appendix E of the Supplement.) *Bottom:* Effect of the BiLRP parameter γ on the explanation.

the two large animals in the image, the true reason for similarity is a small white calf in the right part of the first image. In example F, the scene is cluttered, and does not let appear any meaningful similarity structure, in particular, the two cats are not matched. We also see in this last example that a substantial amount of negative relevance appears, indicating that several joint patterns contradict the similarity score.

The effect of the parameter γ on the explanation is shown in Fig. 6 (G). A low value of γ gives noisy explanations with many negative scores. A high value of γ produces explanations that are mainly positive but also less selective for the exact patterns of similarity. Intermediate values of γ produce the best explanations.

Overall, the BiLRP method gives insight into the strengths and weaknesses of a similarity model, by revealing the features and their relative poses/locations that the model is able or not able to match.

5.1 How *Transferable* is the Similarity Model?

Deep neural networks, through their multiple layers of representation, provide a natural framework for multi-task/transfer learning [66], [67]. DNN-based transfer learning has seen many successful applications [68], [69], [70]. In this section, we consider the problem of transferring a *similarity* model to some task of interest. We will use BiLRP to compare different similarity models, and show how their transferability can be assessed visually from the explanations.

We take the pretrained VGG-16 model and build dot product similarities at layers $5, 10, 17, 24, 31$ (i.e. after each max-pooling layer):

$$y^{(5)}(\boldsymbol{x}, \boldsymbol{x}') = \langle \mathrm{VGG}_{:5}(\boldsymbol{x}), \mathrm{VGG}_{:5}(\boldsymbol{x}') \rangle,$$
$$\vdots$$
$$y^{(31)}(\boldsymbol{x}, \boldsymbol{x}') = \langle \mathrm{VGG}_{:31}(\boldsymbol{x}), \mathrm{VGG}_{:31}(\boldsymbol{x}') \rangle$$

Like in the previous experiment, we add to each feature representation a random projection onto 100 dimensions in order to make explanations faster to compute. In the following experiments, we consider transfer of similarity to the following three datasets:

- 'Unconstrained Facial Images' (UFI) [71],
- 'Labeled Faces in the Wild' (LFW) [72],
- 'The Sphaera Corpus' [29], [73].

The first two datasets are face identification tasks. In identification tasks, a good similarity model is needed in order to reliably extract the closest matches in the training data [53], [74]. The third dataset is composed of 358 scanned academic textbooks from the 15th to the 17th century containing texts, illustrations and tables related to astronomical studies. Again, similarity between these entities is important, as it can serve to consolidate historical networks [56], [75], [76].

Faces and illustrations are fed to the neural network as images of size 64×64 pixels and 96×96 pixels respectively. We choose for each dataset a pair composed of a test example and the most similar training example. For each pair, we compute the BiLRP explanations. Results for the similarity model at layer 17 and 31 are shown in Fig. 7.

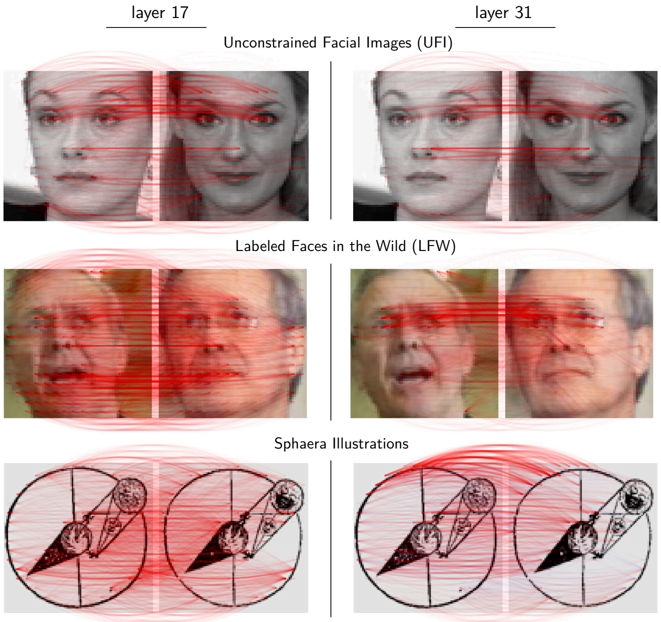

layer 17
layer 31
Unconstrained Facial Images (UFI)
Labeled Faces in the Wild (LFW)
Sphaera Illustrations

Fig. 7. Application of BiLRP to study how VGG-16 similarity transfers to various datasets.

We observe that the explanation of similarity at layer 31 is focused on a limited set of features: the eyes or the nose on face images, and a reduced set of lines on the Sphaera illustrations. In comparison, explanations of similarity at layer 17 cover a broader set of features. These observations suggest that similarity in highest layers, although being potentially capable of resolving very fine variations (e.g. for the eyes), might not have kept sufficiently many features in other regions, in order to match images accurately.

To verify this hypothesis, we train a collection of linear SVMs on each dataset where each SVM takes as input activations at a particular layer. On the UFI dataset, we use the original training and test sets. On LFW and Sphaera, data points are assigned randomly with equal probability to the training and test set. The hyperparameter C of the SVM is selected by grid search from the set of values $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ over 4 folds on the training set. Test set accuracies for each dataset and layer are shown in Table 1.

TABLE 1
Accuracy of a SVM built on different layers of the VGG-16 network and for different datasets.

		layer				
dataset	# classes	5	10	17	24	31
UFI	605	0.45	0.57	**0.62**	0.54	0.19
LFW	61	0.78	0.86	**0.92**	0.89	0.75
Sphaera	111	0.93	0.96	**0.98**	0.97	0.96

These results corroborate the hypothesis initially constructed from the BiLRP explanations: Overspecialization of top layers on the original task leads to a sharp drop of accuracy on the target task. Best accuracies are instead obtained in the intermediate layers.

5.2 How *Invariant* is the Similarity Model?

To further demonstrate the potential of BiLRP for characterizing a similarity model, we consider the problem of assessing its invariance properties. Representations that incorporate meaningful invariance are particularly desirable as they enable learning and generalizing from fewer data points [77], [78], [79].

Invariance can however be difficult to measure in practice: On one hand, the model should respond equally to the input and its transformed version. On the other hand, the response should be selective [80], [81], i.e. not the same for every input. In the context of neural networks, a proposed measure of invariance that implements this joint requirement is the local/global firing ratio [81]. In a similar way, we consider an invariance measure for similarity models based on the local/global similarity ratio:

$$\textsc{Inv} = \frac{\big\langle y(\boldsymbol{x},\boldsymbol{x}')\big\rangle_{\text{local}}}{\big\langle y(\boldsymbol{x},\boldsymbol{x}')\big\rangle_{\text{global}}} \tag{7}$$

The expression $\langle\cdot\rangle_{\text{local}}$ denotes an average over pairs of transformed points (which our model should predict to be similar), and $\langle\cdot\rangle_{\text{global}}$ denotes an average over all pairs of points.

We study the layer-wise forming of invariance in the VGG-16 network. We use for this the 'UCF Sports Action' video dataset [82], [83], where consecutive video frames readily provide a wealth of transformations (translation, rotation, rescaling, etc.) which we would like our model to be invariant to, i.e. produce a high similarity score. Videos are cropped to square shape and resized to size 128 × 128. We define $\langle\cdot\rangle_{\text{local}}$ to be the average over pairs of nearby frames in the same video ($\Delta t \leq 5$), and $\langle\cdot\rangle_{\text{global}}$ to be the average over all pairs, also from different videos. Invariance scores obtained for similarity models built at various layers are shown in Table 2.

TABLE 2
Invariance measured by Eq. (7) at various layers of the VGG-16 network on the UCF Sports Action dataset.

	layer				
	5	10	17	24	31
Inv	2.30	2.31	2.43	2.87	**4.00**

Invariance increases steadily from the lower to the top layers of the neural network and reaches a maximum score at layer 31. We now take a closer look at the invariance score in this last layer, by applying the following two steps:

(i) The invariance score is decomposed on the pairs of video frames that directly contribute to it, i.e. through the term $\langle\cdot\rangle_{\text{local}}$ of Eq. (7).

(ii) BiLRP is applied to these pairs of contributing video frames in order to produce a finer pixel-wise explanation of invariance.

This two-step analysis is shown in Fig. 8 for a selection of videos and pairs of video frames.

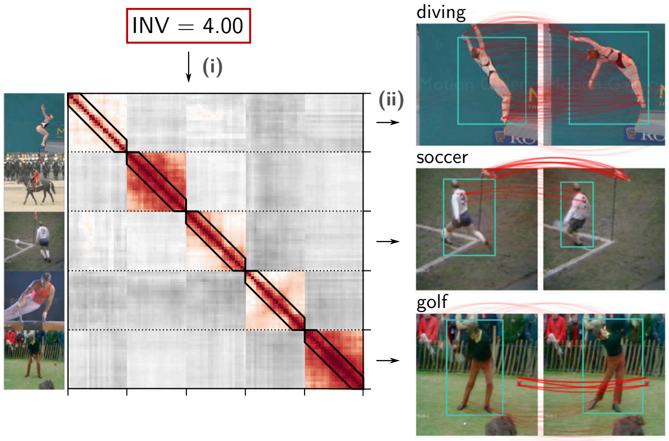


Fig. 8. Explanation of measured invariance at layer 31. *Left:* Similarity matrix associated to a selection of video clips. The diagonal band outlined in black contains the pairs of examples in $\langle\cdot\rangle_{\text{local}}$. *Right:* BiLRP explanations for selected pairs from the diagonal band.

The first example shows a diver rotating counterclockwise as she leaves the platform. Here, the contribution to invariance is meaningfully attributed to the different parts of the rotating body. The second example shows a soccer player performing a corner kick. Part of the invariance is attributed to the player moving from right to left, however, a sizable amount of it is also attributed in an unexpected manner to the static corner flag behind the soccer player. The last example shows a golf player as he strikes the ball. Again, invariance is unexpectedly attributed to a small red object in the grass. This small object would have likely been overlooked, even after a preliminary inspection of the input images.

The reliance of the invariance measure on unexpected objects in the image (corner flag, small red object) can be viewed as a 'Clever Hans' effect [19]: the observer assesses how 'intelligent' (or invariant) the model is, based on looking at the outcome of a given experiment (the computed invariance score), instead of investigating the decision structure that leads to the high invariance score. This effect may lead to an overestimation of the invariance properties of the model.

Similar 'Clever Hans' effects can also be observed beyond video data, e.g. when applying the similarity model to illustrations in the Sphaera corpus. Figure 9 shows two pairs of illustrations whose content is equivalent up to a rotation, and for which our model predicts a high similarity.

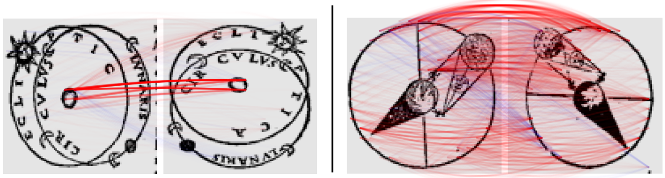


Fig. 9. Pairs of illustrations from the Sphaera corpus, explained with BiLRP. The high similarity originates mainly from matching fixed features in the image rather than capturing the rotating elements.

Once more, BiLRP reveals in both cases that the high similarity is not due to matching the rotated patterns, but mainly fixed elements at the center and at the border of the image respectively.

Overall, we have demonstrated that BiLRP can be useful to identify unsuspected and potentially undesirable reasons for high measured invariance. Practically, applying this method can help to avoid deploying a model with false expectations in real-world applications. Our analysis also suggests that better *explanation-based* invariance measures could be designed in the future, potentially in combination with optical flows [84], in order to better distinguish between the matching structures that should and should not contribute to the invariance score.

6 Building Better Similarity Models

In this section we discuss how to produce better and more useful similarity models with the help of BiLRP. First, we show in Section 6.1 how the interpretable feedback provided by BiLRP can be used to fix a flawed similarity model. Then, we engineer in Section 6.2 a domain-specific similarity model which is both predictive and explainable with BiLRP.

6.1 Fixing a 'Clever Hans' Similarity Model

In the example of Fig. 9, BiLRP has revealed a Clever Hans effect of the similarity model: The model would assign high similarity between rotated images *not* by matching the rotated elements, but by matching the few elements that are invariant to such rotation. With this particular decision structure, the model will likely not generalize well to a broader set of images.

To force rotation invariance into the model, a simple fix is to compute the similarity score for all flips/rotations τ, τ' of the two input images, and output the maximum similarity score:

$$y^{(\text{new})}(\boldsymbol{x}, \boldsymbol{x}') = \max_{\tau, \tau'} \; y(\tau(\boldsymbol{x}), \tau'(\boldsymbol{x}'))$$

Note that τ, τ' can be expressed as linear operation on their input, and the maximum function is also locally linear. With these simple transformations, BiLRP remains applicable and the explanation is obtained in this case by applying BiLRP to the flips/rotations corresponding to the highest similarity score. Explanations of similarities predicted by the improved model are shown in Fig. 10.

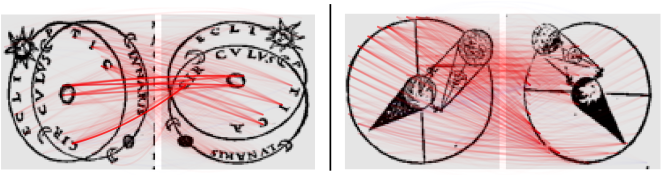


Fig. 10. Pairs of illustrations from the Sphaera corpus and BiLRP explanation for the *improved* similarity model. Similarity captures rotating elements such as letters.

Compared to the original model (Fig. 9), some of the rotating patterns are now being matched, for example, the sequence of letters 'tic' in the first pair of images.

However, this simple enhancement does not resolve *all* weaknesses of the similarity model. In the second pair of images, we observe that the actual image content, e.g. the planet's triangular shadow, remains largely unattended. Therefore, further enhancements of the similarity model (e.g. extracting additional features from the images) are needed. Comprehensively fixing a similarity model would require a way to screen through many pairs of data points and their corresponding explanations (e.g. using the SpRAy visualization technique [19]), and then, a mechanism to systematically turn explanatory feedback into model improvements.

6.2 Engineering an Explainable Similarity Model

An alternative approach is to build specific similarity models that do not rely on generic pretrained features, and are instead *engineered* to address the peculiarities of the problem at hand. We use this engineered approach to address another open and significant problem in the digital humanities: assessing similarity between numeric tables in historical textbooks. We consider scanned numeric tables from the Sphaera Corpus [29]. Tables contained in the corpus typically report astronomical measurements or calculations of the positions of celestial objects in the sky. Examples of such tables are given in Fig. 11 A. Producing an accurate model of similarity between astronomical tables would allow to further consolidate historical networks, which would in turn allow for better inferences.

The similarity prediction task has so far proven challenging: Unlike natural images, faces, or illustrations, which are all well represented by existing pretrained convolutional neural networks, table data usually requires ad-hoc approaches [85], [86]. In particular, we need to specify which aspects of the tables (e.g. numbers, style, or layout) the similarity model should support. Furthermore, end-to-end similarity labels are expensive to obtain, and it is easier to produce annotations for table content directly, e.g. single digit labels. With these intermediate labels, an ad-hoc training approach is needed. Lastly, it is also essential that the produced model retains explainability in order to verify that the knowledge built into the model is effectively used. Therefore, the model must retain the basic structures that make explanation techniques such as BiLRP applicable.

6.2.1 The 'Bigram Network'

We propose a novel 'bigram network' to predict table similarity. Our network can be learned from a limited number of single-digit annotations and is designed to encourage the prediction to be based on relevant numerical features. Also, it is only composed of linear/ReLU and positive homogeneous layers so that it remains explainable with BiLRP. The proposed bigram network consists of two parts:

The first part is a standard stack of convolution/ReLU layers taking a scanned table $\boldsymbol{x}$ as input and producing 10 activation maps $\{\boldsymbol{a}_j(\boldsymbol{x})\}_{j=1}^{10}$ detecting the digits 0–9. The map $\boldsymbol{a}_j(\boldsymbol{x})$ is trained to produce small Gaussian blobs at locations where digits of class j are present. The convolutional network is trained on a few hundreds of single digit labels along with their respective image patches. We also incorporate a comparable amount of negative examples (from non-table pages) to correctly handle the absence of digits.

The second part of the network is a hard-coded sequence of layers that extracts task-relevant information from the

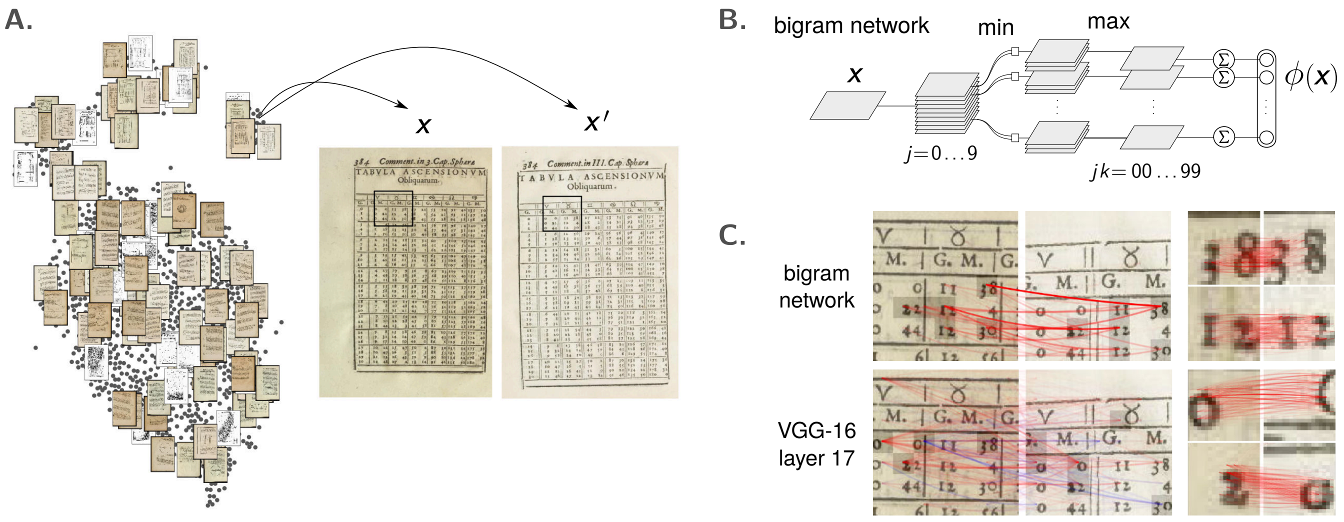


Fig. 11. **A.** Collection of tables from the Sphaera Corpus [29] from which we extract two tables with identical content. **B.** Proposed 'bigram network' supporting the table similarity model. **C.** BiLRP explanations of predicted similarities between the two input tables.

single-digit activation maps. The first layer in the sequence performs an element-wise 'min' operation:

$$\boldsymbol{a}_{jk}^{(\tau)}(\boldsymbol{x}) = \min\big\{\boldsymbol{a}_j(\boldsymbol{x}), \tau(\boldsymbol{a}_k(\boldsymbol{x}))\big\}$$

The 'min' operation be interpreted as a continuous 'AND' [38], and tests at each location for the presence of bigrams $jk \in$ 00–99. The function τ represents some translation operation, and we apply several of them to produce candidate alignments between the digits forming the bigrams (e.g. horizontal shifts of 8, 10, and 12 pixels). We then apply the max-pooling layer:

$$\boldsymbol{a}_{jk}(\boldsymbol{x}) = \max_{\tau}\big\{\boldsymbol{a}_{jk}^{(\tau)}(\boldsymbol{x})\big\}.$$

The 'max' operation can be interpreted as a continuous 'OR', and determines at each location whether a bigram has been found for at least one candidate alignment. Finally, a global sum-pooling layer is applied spatially:

$$\phi_{jk}(\boldsymbol{x}) = \big\|\boldsymbol{a}_{jk}(\boldsymbol{x})\big\|_1$$

It introduces global translation invariance into the model and produces a 100-dimensional output vector representing the sum of activations for each bigram. The bigram network is depicted in Fig. 11 B.

From the output of the bigram network, the similarity score can be obtained by applying the dot product $y(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$. Furthermore, because the bigram network is exclusively composed of convolution/ReLU layers and standard pooling operations, similarities built at the output of this network remain fully explainable by BiLRP.

6.2.2 Validating the 'Bigram Network' with BiLRP

We come to the final step which is to validate the 'bigram network' approach on the task of predicting table similarity. Examples of common validation procedures include precision-recall curves, or the ability to solve a proxy task (e.g. table classification) from the predicted similarities. These validation procedures require end-to-end label information, which is however difficult to obtain for this type of data. Furthermore, when the labeled data is not sufficiently representative, these procedures are potentially affected by the 'Clever Hans' effect [19].

In the following, we will show that BiLRP, through the explanatory feedback it provides, offers a much more data efficient way of performing model validation. We take a pair of tables $(\boldsymbol{x}, \boldsymbol{x}')$, which a preliminary manual inspection has verified to be similar. We then apply BiLRP to explain:

(i) the similarity score at the output of our engineered task-specific 'bigram network',
(ii) the similarity score at layer 17 of a generic pretrained VGG-16 network.

For the bigram network, the BiLRP parameter γ is set to 0.5 at each convolution layer. For the VGG-16 network, we use the same BiLRP parameters as in Section 5. The result of our analysis is shown in Fig. 11 C.

The bigram network similarity model correctly matches pairs of digits in the two tables. Furthermore, matches are produced between sequences occurring at different locations, thereby verifying the structural translation invariance of the model. Pixel-level explanations further validate the approach by showing that individual digits are matched in a meaningful manner. In contrast, the similarity model built on VGG-16 does not distinguish between the different pairs of digits. Furthermore, part of the similarity score is supported by aspects that are not task-relevant, such as table borders.—Hence, for this particular table similarity task, BiLRP can clearly establish the superiority of the bigram network over VGG-16.

We stress that this assessment could be readily obtained from a *single* pair of tables. If instead we would have applied a validation technique that relies only on similarity scores, significantly more data would have been needed in order to reach the same conclusion with confidence. This sample efficiency of BiLRP (and by extension any successful explanation technique) for the purpose of model validation is especially important in digital humanities or other scientific domains, where ground-truth labels are typically scarce or expensive to obtain.

7 Conclusion

Similarity is a central concept in machine learning that is precursor to a number of supervised and unsupervised machine learning methods. In this paper, we have shown that it can be crucial to get a human-interpretable explanation of the predicted similarity before using it to train a practical machine learning model.

We have contributed a theoretically well-founded method to explain similarity in terms of pairs of input features. Our method called BiLRP can be expressed as a composition of LRP computations. It therefore inherits its robustness and broad applicability, but extends it to the novel scenario of similarity explanation.

The usefulness of BiLRP was showcased on the task of understanding similarities as implemented by the VGG-16 neural network, where it could predict transfer learning capabilities and highlight clear cases of 'Clever Hans' [19] predictions. Furthermore, for a practically relevant problem in the digital humanities, BiLRP was able to demonstrate with very limited data the superiority of a task-specific similarity model over a generic VGG-16 solution.

Future work will extend the presented techniques from binary towards n-ary similarity structures, especially aiming at incorporating the different levels of reliability of the input features. Furthermore we will use the proposed research tool to gain insight into large data collections, in particular, grounding historical networks to interpretable domain-specific concepts.

Acknowledgements

This work was funded by the German Ministry for Education and Research as BIFOLD – Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref. 01IS18037A), and the German Research Foundation (DFG) as Math+: Berlin Mathematics Research Center (EXC 2046/1, project-ID: 390685689). This work was partly supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-00451, No. 2017-0-01779).

References

[1] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller, "Engineering support vector machine kernels that recognize translation initiation sites," *Bioinformatics*, vol. 16, no. 9, pp. 799–807, 2000.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

[3] A. Nierman and H. V. Jagadish, "Evaluating structural similarity in XML documents," in *WebDB*, 2002, pp. 61–66.

[4] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classification," in *ISMIR*, 2005, pp. 628–633.

[5] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 983–996, 1998.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[7] B. Schölkopf and A. J. Smola, *Learning with Kernels: support vector machines, regularization, optimization, and beyond*, ser. Adaptive computation and machine learning series. MIT Press, 2002.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[9] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[12] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[14] L. van der Maaten and G. E. Hinton, "Visualizing non-metric similarities in multiple maps," *Machine Learning*, vol. 87, no. 1, pp. 33–55, 2012.

[15] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *ICML*, ser. ACM International Conference Proceeding Series, vol. 69. ACM, 2004.

[16] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, 2006.

[17] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[18] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[19] 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*, vol. 10, p. 1096, 2019.

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

[21] Z. C. Lipton, "The mythos of model interpretability," *Commun. ACM*, vol. 61, no. 10, pp. 36–43, 2018.

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

[23] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.

[24] 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*, vol. 10, no. 7, p. e0130140, 07 2015.

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

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *ICCV*. IEEE Computer Society, 2017, pp. 618–626.

[27] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

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

[29] M. Valleriani, F. Kräutli, M. Zamani, A. Tejedor, C. Sander, M. Vogl, S. Bertram, G. Funke, and H. Kantz, "The emergence of epistemic communities in the sphaera corpus: Mechanisms of knowledge evolution," *Journal of Historical Network Research*, vol. 3, pp. 50–91, 2019.

[30] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[31] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, Jul. 2006.

[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV (1)*, ser. Lecture Notes in Computer Science, vol. 8689. Springer, 2014, pp. 818–833.

[33] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *ICLR (Poster)*. OpenReview.net, 2017.

[34] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017, pp. 4765–4774.

[35] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *ICLR (Workshop Poster)*, 2014.

[36] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "SmoothGrad: removing noise by adding noise," *CoRR*, vol. abs/1706.03825, 2017.

[37] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 3319–3328.

[38] J. Kauffmann, K.-R. Müller, and G. Montavon, "Towards explaining anomalies: A deep Taylor decomposition of one-class models," *Pattern Recognition*, p. 107198, 2020.

[39] B. Micenková, R. T. Ng, X. Dang, and I. Assent, "Explaining outliers by subspace separability," in *ICDM*. IEEE Computer Society, 2013, pp. 518–527.

[40] J. Kauffmann, M. Esders, G. Montavon, W. Samek, and K.-R. Müller, "From clustering to cluster explanations via neural networks," *CoRR*, vol. abs/1906.07633, 2019.

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 2048–2057.

[42] M. Zheng, S. Karanam, T. Chen, R. J. Radke, and Z. Wu, "Learning similarity attention," *CoRR*, vol. abs/1911.07381, 2019.

[43] M. Tsang, D. Cheng, and Y. Liu, "Detecting statistical interactions from neural network weights," in *ICLR (Poster)*. OpenReview.net, 2018.

[44] T. Cui, P. Marttinen, and S. Kaski, "Recovering pairwise interactions using neural networks," *CoRR*, vol. abs/1901.08361, 2019.

[45] S. Leupold, "Second-order Taylor decomposition for explaining spatial transformation of images," Master's thesis, Technische Universität Berlin, 2017.

[46] M. Simon, E. Rodner, T. Darrell, and J. Denzler, "The whole is more than its parts? from explicit to implicit pose normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 749–763, 2020.

[47] 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," in *KDD*. ACM, 2015, pp. 1721–1730.

[48] J. D. Janizek, P. Sturmfels, and S. Lee, "Explaining explanations: Axiomatic feature interactions for deep networks," *CoRR*, vol. abs/2002.04138, 2020.

[49] C. Watkins, "Dynamic alignment kernels," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 39–50.

[50] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller, "A new discriminative kernel from probabilistic models," *Neural Computation*, vol. 14, no. 10, pp. 2397–2414, 2002.

[51] T. Gärtner, "A survey of kernels for structured data," *SIGKDD Explorations*, vol. 5, no. 1, pp. 49–58, 2003.

[52] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *NIPS*. Morgan Kaufmann, 1993, pp. 737–744.

[53] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR (1)*. IEEE Computer Society, 2005, pp. 539–546.

[54] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *CVPR*. IEEE Computer Society, 2014, pp. 1386–1393.

[55] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *SIMBAD*, ser. Lecture Notes in Computer Science, vol. 9370. Springer, 2015, pp. 84–92.

[56] B. Seguin, C. Striolo, I. diLenardo, and F. Kaplan, "Visual link retrieval in a database of paintings," in *ECCV Workshops (1)*, ser. Lecture Notes in Computer Science, vol. 9913. Springer, 2016, pp. 753–767.

[57] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *WWW*. ACM, 2017, pp. 173–182.

[58] R. Memisevic and G. E. Hinton, "Learning to represent spatial transformations with factored higher-order Boltzmann machines," *Neural Computation*, vol. 22, no. 6, pp. 1473–1492, 2010.

[59] K. Tzompanaki and M. Doerr, "A new framework for querying semantic networks." in *Proceedings of Museums and the Web 2012: the international conference for culture and heritage on-line*, 2012.

[60] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 342–350.

[61] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, ser. JMLR Proceedings, vol. 15. JMLR.org, 2011, pp. 315–323.

[62] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI*, ser. Lecture Notes in Computer Science. Springer, 2019, vol. 11700, pp. 193–209.

[63] H. Zhang, J. Chen, H. Xue, and Q. Zhang, "Towards a unified evaluation of explanation methods without ground truth," *CoRR*, vol. abs/1911.09017, 2019.

[64] S. Lapuschkin, A. Binder, K.-R. Müller, and W. Samek, "Understanding and comparing deep neural networks for age and gender classification," in *IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1629–1638.

[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[66] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[67] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *CVPR*. IEEE Computer Society, 2014, pp. 1717–1724.

[68] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," in *KDD*. ACM, 2015, pp. 1475–1484.

[69] G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[70] Y. Gao and K. M. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Comp.-Aided Civil and Infrastruct. Engineering*, vol. 33, no. 9, pp. 748–768, 2018.

[71] L. Lenc and P. Král, "Unconstrained facial images: Database for face recognition under real-world conditions," in *MICAI (2)*, ser. Lecture Notes in Computer Science, vol. 9414. Springer, 2015, pp. 349–361.

[72] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[73] M. Valleriani, "Prolegomena to the study of early modern commentators on Johannes de Sacrobosco's tractatus de sphaera," in *De sphaera of Johannes de Sacrobosco in the Early Modern Period: The Authors of the Commentaries*. Springer Nature, 2019, pp. 1–23.

[74] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10, 000 classes," in *CVPR*. IEEE Computer Society, 2014, pp. 1891–1898.

[75] F. Kräutli and M. Valleriani, "CorpusTracer: A CIDOC database for tracing knowledge networks," *DSH*, vol. 33, no. 2, pp. 336–346, 2018.

[76] S. Lang and B. Ommer, "Attesting similarity: Supporting the organization and study of art image collections with computer vision," *Digital Scholarship in the Humanities*, vol. 33, no. 4, pp. 845–856, 04 2018.

[77] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, 2013.

[78] 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*, vol. 3, no. 5, p. e1603015, 2017.

[79] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nature Communications*, vol. 9, no. 1, Sep. 2018.

[80] F. Anselmi, L. Rosasco, and T. Poggio, "On invariance and selectivity in representation learning," *Information and Inference*, vol. 5, no. 2, pp. 134–158, May 2016.

[81] I. J. Goodfellow, Q. V. Le, A. M. Saxe, H. Lee, and A. Y. Ng, "Measuring invariances in deep networks," in *NIPS*. Curran Associates, Inc., 2009, pp. 646–654.

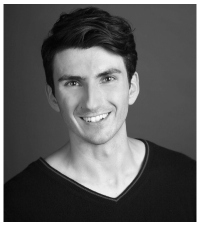
[82] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[83] K. Soomro and A. Zamir, "Action recognition in realistic sports videos," *Advances in Computer Vision and Pattern Recognition*, vol. 71, pp. 181–208, 01 2014.

[84] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*. IEEE Computer Society, 2015, pp. 2758–2766.

[85] M. Husson, "Remarks on two dimensional array tables in latin astronomy: a case study in layout transmission," *Suhayl. Journal for the History of the Exact and Natural Sciences in Islamic Civilisation*, vol. 13, pp. 103–117, 2014.

[86] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "DeepDeSRT: Deep learning for detection and structure recognition of tables in document images," in *ICDAR*. IEEE, 2017, pp. 1162–1167.

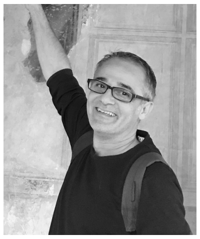


Oliver Eberle received a Joint M.Sc. in Computational Neuroscience from Technische Universität Berlin and Humboldt Universität zu Berlin in 2017. He is currently pursuing a Ph.D. in the Machine Learning Group at TU Berlin and his research focuses on explainable machine learning and natural language processing.

Jochen Büttner Jochen Büttner received his physics diploma from the Freie Universität Berlin in 1998 and his Ph.D. degree in history from the Humboldt University zu Berlin in 2009. Currently he is a Research Associate at the Max Planck Institute for the History of Science in Berlin. In the context of the Berlin Center for Machine Learning (BZML) he is at present exploring the potential of the application ML approaches in the history of science.

Florian Kräutli Florian Kräutli leads the Digital Humanities activities at the Max Planck Institute for the History of Science in Berlin. His research focuses on digital methods for knowledge production in the Humanities, specializing in knowledge representation and visualization. He obtained a PhD in this area at the Royal College of Art, London. Previously he completed an MSc in Cognitive Computing at Goldsmiths, University of London, focusing on philosophy of perception and artificial intelligence, and trained as a designer at the Design Academy Eindhoven.

Klaus-Robert Müller (M'12) has been a professor of computer science at Technische Universität Berlin since 2006; at the same time he is co-directing the Berlin Big Data Center. He studied physics in Karlsruhe from 1984 to 1989 and obtained his Ph.D. degree in computer science at Technische Universität Karlsruhe in 1992. After completing a postdoctoral position at GMD FIRST in Berlin, he was a research fellow at the University of Tokyo from 1994 to 1995. In 1995, he founded the Intelligent Data Analysis group at GMD-FIRST (later Fraunhofer FIRST) and directed it until 2008. From 1999 to 2006, he was a professor at the University of Potsdam. He was awarded the Olympus Prize for Pattern Recognition (1999), the SEL Alcatel Communication Award (2006), the Science Prize of Berlin by the Governing Mayor of Berlin (2014), and the Vodafone Innovations Award (2017). In 2012, he was elected member of the German National Academy of Sciences-Leopoldina, in 2017 of the Berlin Brandenburg Academy of Sciences and also in 2017 external scientific member of the Max Planck Society. In 2019 he became ISI Highly Cited Researcher. His research interests are intelligent data analysis and machine learning with applications in neuroscience (specifically brain-computer interfaces), physics and chemistry.



Matteo Valleriani Matteo Valleriani is Research Group Leader in Dept. I, Honorary Professor at the Technische Universität Berlin, Professor for Special Appointments at the Faculty of Humanities at Tel Aviv University, and Principal Investigator of the Project "Images and Configurations in Corpora of University Textbooks" at the Berlin Center for Machine Learning. In his research, he investigates processes of 1) emergence of scientific knowledge in relation to its practical, social, and institutional dimensions, and 2) homogenization of scientific knowledge in the framework of Cultural Heritage Studies. A further focus of his research is on the epistemic function of visual material in scientific research and in the framework of processes of knowledge transformation.

Grégoire Montavon received a Masters degree in Communication Systems from École Polytechnique Fédérale de Lausanne, in 2009 and a Ph.D. degree in Machine Learning from the Technische Universität Berlin in 2013. He is currently a Research Associate in the Machine Learning Group at TU Berlin. His research interests include interpretable machine learning and deep neural networks.