Explainable Structured Machine Learning

Insights into Similarity, Graph and Transformer Models

Explainable Structured Machine Learning

Insights into Similarity, Graph and Transformer Models

vorgelegt von

M. Sc.

Oliver Eberle

ORCID: 0000-0002-6967-9950

an der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Benjamin Blankertz
Gutachter:	Prof. Dr. Klaus-Robert Müller
Gutachter:	Prof. Dr. Wojciech Samek
Gutachter:	Prof. Dr. Andreas Holzinger

Tag der wissenschaftlichen Aussprache: 22. Juli 2022

Berlin 2022

Abstract

Explainable artificial intelligence aims to make complex machine learning models interpretable. Having access to transparent prediction processes is crucial to ensure the safe, trustworthy and fair use of machine learning in science, industry and society. Unfortunately, many widely used models such as deep similarity models, graph neural networks and Transformer models, are highly non-linear and structured in ways that challenge the extraction of meaningful explanations.

The well-established layer-wise relevance propagation explanation method with its theoretical foundation in deep Taylor decomposition serves as a methodological anchor to develop explanation techniques that consider the particular model structure. Specifically, we investigate how to explain dot product similarity, graph neural network predictions and self-attention modules in Transformer models.

We observe that this can require to go beyond standard explanations in terms of input features that result in second-order and higher-order attributions. This motivates to extend existing approaches for the evaluation and visualization of explanation techniques to these new types of explanations.

In parallel to these methodological contributions, we investigate how these methods can be used in different domain applications. In particular, we apply the different explanation methods to a variety of use cases. We build and explain a similarity model designed to represent numerical content in the digital humanities to study the evolution of the history of science, revisit image classification by visualizing the relevance flow through the different processing layers and turn to natural language processing to investigate gender bias in Transformer models as well as analyze Transformer explanations during task-solving.

Throughout our experiments and analyses, we demonstrate that a careful treatment of model structure in explanation methods can improve their faithfulness, result in better explanations and enable novel insights.

Zusammenfassung

Erklärbare künstliche Intelligenz zielt darauf ab, komplexe maschinelle Lernmodelle interpretierbar zu machen. Der Zugang zu transparenten Vorhersageprozessen ist entscheidend für die sichere, vertrauenswürdige und faire Nutzung des maschinellen Lernens in Wissenschaft, Industrie und Gesellschaft. Leider sind viele weit verbreitete Modelle wie tiefe Ähnlichkeits-Modelle, neuronale Graphennetze und Transformer-Modelle äußerst nichtlinear und in einer Weise strukturiert, die die Berechnung verlässlicher Erklärungen erschwert.

Die etablierte Methode der layer-wise relevance propagation mit ihrer theoretischen Grundlage in der tiefen Taylor-Dekomposition dient hier als methodologischer Anker für die Entwicklung von Erklärungstechniken, welche besondere Eigenschaften der Modellstruktur berücksichtigen. Konkret untersuchen wir, wie Skalarprodukt-Ähnlichkeit, Berechnungen von Graph Neural Networks und Attention-Module in Transformer-Modellen erklärbar gemacht werden können.

Dies kann erfordern, über Standarderklärungen in Bezug auf Eingangsmerkmale hinauszugehen und Attributionen zweiter und höherer Ordnung zu berücksichtigen. Dies motiviert die Ausweitung bestehender Ansätze zur Evaluation und Visualisierung von Erklärungstechniken auf diese neuen Arten von Erklärungen.

Parallel zu diesen methodologischen Beiträgen untersuchen wir, wie diese Methoden in verschiedenen Anwendungsbereichen eingesetzt werden können. Insbesondere wenden wir die verschiedenen Erklärungsmethoden auf eine Vielzahl von Anwendungsfällen an. Wir entwickeln und machen ein Ähnlichkeitsmodell zur Darstellung numerischer Inhalte in den digitalen Geisteswissenschaften erklärbar, was es uns ermöglicht die Entwicklung der Wissenschaftsgeschichte zu untersuchen. Zudem untersuchen wir Bildklassifizierungs-Modelle und visualisieren den Relevanzfluss durch die verschiedenen Verarbeitungsebenen, wenden uns der Verarbeitung natürlicher Sprache zu, untersuchen geschlechtsspezifische Voreingenommenheit in Transformer-Modellen und analysieren aufgabenspezifische Transformer-Erklärungen.

In unseren Experimenten und Analysen zeigen wir, dass eine sorgfältige Behandlung von strukturierten Informationen zu besseren Erklärungen führt und neue Erkenntnisse ermöglicht.

Acknowledgements

Finishing a PhD thesis is an exciting endeavor, and I have been very fortunate to have been accompanied by many remarkably inspiring people who have made this a wonderful and formative experience!

First and foremost, I would like to express my deep gratitude to Klaus-Robert Müller. Thank you for welcoming me to the lab and for your passion, foresight, and ambition that created this fantastic research environment. I have gained a lot from your generous feedback and support throughout this time, and I truly appreciate your trust and occasional extra push to make our efforts shine.

I am particularly grateful to Grégoire Montavon; for sharing his insights into explainable AI, the many stimulating discussions, and his dedication to progress machine learning. Your ideas and feedback have taught and inspired me in many ways.

Special thanks go to Matteo Valleriani and Jochen Büttner, who have so generously shared their knowledge and passion for the history of science in many meetings and educational excursions. I would like to thank Anders Søgaard for his valuable advice, expertise and guidance in the fields of language and cognition. Additionally, I would like to thank Lior Wolf and Ameen Ali for sharing ideas and experimenting together; it was a pleasure working with you. I would also like to thank Florian, Hassan, Kristof, Jan, Jonas, Shin, Stephan, Stephanie, Thomas, Olya, and again Klaus, Matteo, Anders, Lior, Jochen and Grégoire, for the many collaborative brainstorms and solutions we experienced together.

Many thanks go to all my fellow researchers and friends of the machine learning group. I am very grateful for all the shared memories that we made together during lunch breaks, coffee gatherings, karaoke sessions, canoeing, and many other occasions. I would also like to thank our administrative team, Andrea, Cecilia and Kerstin, and Dominik, for always having all the answers and keeping our group running smoothly.

A special thanks goes to my writing club–without you, Laura and Tabea, creating this document would have been less cheerful. I thank my amazing support outside academia wholeheartedly–our relationships have contributed so much to maneuvering this adventure with joy and curiosity.

Last but not least, I would like to thank my parents for letting me and my sister pursue our ambitions and for their invaluable support throughout the years. Dankeschön!

Table of Contents

1 Introduction

Recent developments in machine learning have enabled impressive advances that are already widely used in the sciences, industry and our everyday lives. Especially deep learning has led to progress in many challenges that range from object recognition, image segmentation, reinforcement learning, machine translation and language modeling, to protein-structure prediction and quantum-chemical dynamics [1, 2, 3, 4, 5, 6].

This variety of applications motivates the use of specifically *structured* model architectures and data representations that are suited to the particular domain. For example, similarity between images is well represented by models that consist of two branches of convolutional neural networks (CNNs). In contrast, relationships between elements, e.g., words or molecules, are best encoded in the graph structure of graph neural networks (GNNs). More recently, the rise of attention mechanisms, and especially *Transformers*, which are models that stack several attention modules for the processing of sequences, have become the default architecture choice for natural language processing (NLP) and the modeling of many sequential data tasks [7].

While these highly non-linear models often achieve human or beyond human levels of performance, users and machine learning experts alike are often not able to fully understand their complex decision processes. The field of *explainable artificial intelligence* (explainable AI) aims to increase model transparency and enable the development of more trustworthy systems for the safe and robust use of machine learning [8, 9, 10, 11, 12].

Consider a machine learning model that is trained to separate dogs from cats. Given the set of cat images readily available, we would now like to understand what features of the image are most relevant for the model to infer the correct prediction. Standard explanation methods produce a heatmap over the input image

that highlights these features, and hence provide information to the user for evaluation and a better understanding of the model prediction.

In the fast-growing field of explainable AI, various approaches have been developed, ranging from self-explainable models designed to be directly interpretable [13, 14, 15], perturbation analyses that iteratively mask parts of the input to observe which are most relevant to change the prediction [16, 17, 18], the analysis of model gradients that represent the change of the model function for a given input [19, 20, 21, 22], to computing propagation-based explanations which progressively redistribute the model prediction through the layered network architecture following a set of redistribution rules [20, 23, 24, 25]. Herein supervised settings have been the standard scenario, and recently, explainable AI methods have been extended to semi-supervised and unsupervised models, including clustering [26] and anomaly detection [27, 28].

The danger of defective or unexpected model behavior has accompanied the standard approach of training large models with ever-increasing datasets. This has made it difficult to guarantee reasonable, i.e., intentional, human-understandable and causal solutions that comply with the aims and functions of explainable AI. Examples include deep neural network (DNN) models that can predict object classes in images but ground their prediction on background information [29, 30], or that can be fooled by imperceptible input changes [31, 32], language processing systems that make correct inferences using deficient heuristics [33, 34], and reinforcement learning agents that choose death repeatedly instead of mastering more challenging game levels [35]. Such defective model solutions where the model makes a correct decision using a 'wrong' or unexpected strategy are known as 'Clever Hans' strategies [30].

These examples highlight that we need reliable and robust explanations that are not prone to disguise the relevant model strategies or to produce unstable explanations. Explanation techniques are available for standard machine learning models and applications such as CNNs for object detection. But, we will observe that the complexity of specifically structured models requires careful development of explanation methods. Some of these methods motivate explanations that go beyond the common explanation in terms of input features, e.g., heatmaps over input images, resulting in explanations that highlight more complex feature interactions.

Before machine learning models can be used to extract reliable inferences, a careful evaluation of explanation methods is needed. The development of controllable evaluation scenarios and an extension of current evaluation methods to these new types of explanations are thus required. This raises the question of what properties a successful explainable model should fulfill. In the literature, numerous desired explanation properties have been proposed [36, 37, 38, 12]. These can be separated into automated evaluation approaches that assess faithfulness, sparsity, robustness or computational efficiency, and human evaluations that investigate understandability, accuracy as compared to human annotations or correlation to psychophysical signals.

This thesis considers the process from model formulation, implementation and evaluation to domain application, and finally, the generation of insights. In particular, we focus on the established and theoretically well-founded layer-wise relevance

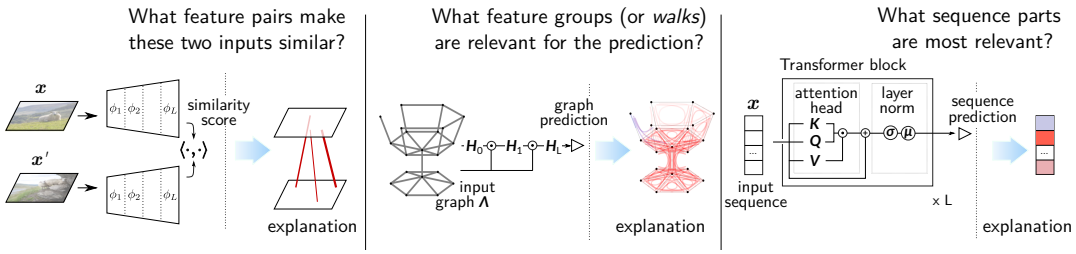


Figure 1.1: Thesis overview. Extracting explanations for models of different structures. *Left:* Interpreting similarity models using second-order explanations. *Center:* Higher-order explanations for graph neural networks. *Right:* Conservative explanations in Transformer models.

propagation (LRP) method [23, 25] as the starting point to develop propagation-based explanation techniques for similarity models, graph neural networks and Transformers. Their respective model structure requires a careful decomposition of the model prediction to the input components. The resulting layer-wise propagation rules necessary to compute relevance scores are developed and evaluated against common baseline methods. We apply these novel explanation methods to a variety of use cases: we build and explain a similarity model designed to represent numerical content in the digital humanities (DH) to study the evolution of the history of science, revisit image classification by visualizing how a class decision is formed through the different processing layers, and turn to language-processing systems to investigate gender bias in Transformer language models and analyze their explanations in comparison to human attention during task-solving.

1.1 Contributions and Structure of the Thesis

The overall structure of the thesis is summarized in Figure 1.1. We demonstrate in the following how explanations for specifically structured models, particularly similarity models, GNNs and Transformers, can be developed. The relevant fundamentals in explainable AI, including an introduction to the layer-wise relevance propagation technique, are introduced in Chapter 2. Then we continue with our findings and analyses in the following main chapters:

Chapter 3 In this chapter, we develop a method that computes explanations for similarity models. We introduce similarity as a dot product computation and describe how this model structure can be represented using second-order terms that signal interactions between features. This results in a baseline for explaining similarity based on the Hessian. Then we introduce BiLRP, a method that computes theoretically founded and robust explanations for deep similarity models by decomposing the dot product similarity to pairs of input features. We further demonstrate the effectiveness of BiLRP in experiments on similarity between natural images, video

data and historical illustrations. This leads to a use case in the digital humanities, in which we build a task-specific and interpretable similarity model that enables us to extract corpus-level insights.

Chapter 4 In this chapter, we introduce explanations for graph neural networks. We first describe the specific structure underlying typical GNN models and identify a formulation using higher-order terms that describes how the GNN structure interacts to computes its prediction. This leads us to the introduction of GNN-LRP, which is a method to compute higher-order explanations for graphs. We evaluate GNN-LRP on a set of synthetic and real-world examples, and in a use case on image recognition, GNN-LRP enables us to identify flawed model behavior.

Chapter 5 In this chapter, we investigate explanations for the widely-used Transformer architecture. We observe that the gradient in Transformers requires specific treatment to reflect the model prediction reliably and meet the desired principle of conservation. Further, we look into the structures that break conservation and find that self-attention heads and layer normalization computations are the main factors. We introduce new propagation rules to produce better explanations and evaluate their effectiveness compared to other commonly used Transformer explanations. We finally turn to two use cases in detecting biased model behavior and alignment of Transformer explanations to human task-reading.

We then conclude this thesis in Chapter 6, which includes a summary of our main findings, a discussion of their implications and an outlook on promising future directions.

The main contributions are as follows:

- **Second-Order explanations: layer-wise relevance propagation for deep similarity models. (Chapter 3)** We introduce an explanation technique that decomposes the output of (deep) similarity models. The specific model structure motivates our BiLRP explanation method that considers second-order feature contributions to highlight the relevance of feature interactions. We test our method on both toy and real-world datasets, and reveal 'Clever Hans'-type behavior using our visualization technique that connects pairs of relevant input features.
- **Building explainable similarity models for the digital humanities to extract historic insights. (Chapter 3)** We build an explainable table similarity model based on representing dense numerical content via a 'bag-of-bigrams' representation. This enables a previously not possible corpus-level analysis, which reveals spatio-temporal insights into the evolution of knowledge in early modern times.
- **Higher-order explanations for graph neural networks with GNN-LRP. (Chapter 4)** The forward propagation in GNNs requires the computation

through interaction blocks that are closely entangled to the input graph. To make GNN predictions explainable, we perform a Taylor expansion that results in higher-order feature contributions from which we extract walk-based explanations. We demonstrate the usefulness of the GNN-LRP explanation method to gain insights into synthetic growth graphs and image classification models.

- **Better explanations for Transformer models. (Chapter 5)** We demonstrate how better explanations for Transformer models can be obtained by carefully handling the non-conserving self-attention and layer normalization computations. The resulting propagation rules can largely reconstitute conservation of the model prediction score and offer the most faithful explanations in our evaluation experiments.

- **Alignment of human attention with Transformer explanations in NLP. (Chapter 5)** We investigate how well different language model attention vectors, relevance scores, and a cognitive model align to eye tracking-based human attention. We use correlation scores and a perturbation-based analysis and identify a trade-off between correlation to human attention, faithfulness and sparsity during task-solving on two NLP tasks.

1.2 Relation to Previously Published Work

The main contributions and findings of this thesis are based on the following publications:

O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani and G. Montavon. Building and Interpreting Deep Similarity Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1149-1161, 2022.

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. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

O. Eberle*, S. Brandl*, J. Pilot, A. Søgaard. Do Transformer Models Show Similar Attention Patterns to Task-Specific Human Gaze? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022. (*equal contribution)

A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller and L. Wolf. XAI for Transformers: Better Explanations through Conservative Propagation. *(accepted to ICML)*, 2022.

This thesis includes additional contents from the following papers:

> O. Eberle, J. Büttner, H. El-Hajj, G. Montavon, K.-R. Müller and M. Valleriani. From Large Data Collections to Historical Insights: A Machine Learning Analysis of Astronomical Tables in Early Modern Textbooks. *(in preparation)*, 2022.
>
> H. El-Hajj, M. Zamani, J. Büttner, J. Martinetz, O. Eberle, N. Shlomi, A. Siebold, G. Montavon, K.-R. Müller, H. Kantz and M. Valleriani. An Ever-Expanding Humanities Knowledge Graph: The Sphaera Corpus at the Intersection of Humanities, Data Management, and Machine Learning. *Datenbank Spektrum*, 2022.

All co-authors of these works have kindly agreed to the use of content, figures, and results from the works above for this thesis.

2 Fundamentals

To make modern machine learning applicable, we need to reliably extract the underlying mechanisms relevant for a decision in a way that is understandable to humans. Numerous different methods have been proposed to explain the workings of increasingly complex architectures. In the following, we will give an overview of existing explanation approaches and then focus on layer-wise relevance propagation (LRP) [23, 39], which will serve as the foundation of the introduced techniques in the following chapters. We then consider desired criteria for explanations and summarize ways to quantify them in objective automated settings and subjective human evaluation.

2.1 Explainable Artificial Intelligence

Explainable AI has provided machine learning with techniques that reveal what data features contribute the most to a model prediction. This offers an additional way to validate models beyond common evaluation procedures. While high model performance is a central goal of machine learning systems, the roles and functions of explainable AI cover many crucial aspects that go beyond test-set accuracy [8, 9, 40, 41]. Explanations are important to (i) justify the use of a specific model and by this (ii) foster trust and verifiability in machine learning, and can (iii) bring safety and security to sensitive applications, i.e., in medicine, defense, commerce or entertainment use. In parallel, explanations (iv) offer control over complex systems by providing information to identify unexpected model behavior, which can subsequently be used for model improvement. Thus, they play an important role in (v) robustifying models against data bias and dealing with data shifts, and (vi) improving their performance and efficiency during development and debugging. Explanations further provide a basis for (vii) the development of more complex model

architectures that go beyond simple linear methods and, by this, (viii) foster the discovery of novel insights and data patterns. With the extensive use of machine learning across science and society, it has become more and more important to explain the inner workings of models to (ix) ensure compliance and legislative requirements put in place by organizations and governments. These diverse demands increasingly inform the development and selection of machine learning systems, which remains challenging for complex systems.

Less complex early machine learning systems are often directly interpretable, e.g., by observing the value of coefficients in linear and logistic regression or tracing the learned rules in a decision tree. But the successful use of complex models in the sciences and industrial applications has caused a shift towards increasingly deep, non-linear architectures. This has resulted in the black-box scenario in which model decisions are not directly interpretable by humans. The emerging field of explainable AI [8, 10, 11, 12] thus aims to develop techniques that allow to comprehend the inner workings of machine learning models while preserving their performance.

There exist various definitions of understandability, interpretability, explainability, comprehensibility or transparency that are often difficult to isolate. Generally, the idea of making a model decision understandable to an end-user emerges as a common definition [11, 40]. In the following, we use these terms mentioned above interchangeably and adopt a definition focusing on the function of explainable AI systems. Among others, they refer to methods that improve trust in machine learning systems by providing insight into model decisions and why they were made, opening the underlying strategies of a model, and increasing robustness while reducing bias [42, 40].

2.2 Approaches to Explainable AI

Approaches to Explainable AI can be divided broadly into ante-hoc and post-hoc approaches, which we will introduce in the following sections.

2.2.1 Ante-Hoc Explanations

Ante-hoc techniques refer to models that use structures that are interpretable *before* training. Such methods are directly interpretable or self-explainable, e.g., due to their linear structure. This can also include more complex models that have been designed to be interpretable such as linear regression models, which use directly interpretable coefficients, decision trees that build an interpretable hierarchical decision structure, additive models that learn linear relationships between interpretable input features [43, 44, 45], and Bayesian networks that represent connections between features using a probabilistic graphical model offering a direct way to infer dependencies between variables [46, 47, 48]. Other works have proposed prototype methods that summarize dataset characteristics using distance measures and provide interpretable prototypical representations [49, 50, 51], and models with attention modules that provide intuitively understandable importance scores over features [52, 14, 53].

While ante-hoc models are appealing due to their built-in interpretability, this also constrains the design and transfer of such models. An additional problem is that for deeper architectures, it remains unclear how faithful their explanations are with respect to the model prediction since they typically focus on a part of the full processing pipeline, e.g., a single attention-block of the larger models or a linear readout that computes the output from a more complex, not directly interpretable model.

In practice, many influential models, e.g., convolutional neural networks, recurrent neural networks, generative adversarial networks, graph neural networks and self-attention models, are not interpretable by design and hence post-hoc explanations provide methods to make these models explainable.

2.2.2 Post-Hoc Explanations

Post-hoc approaches produce explanations after the model has been trained, providing flexibility during model design and selection [41]. Many modern machine learning models thus rely on post-hoc methods to explain their prediction strategies. For a standard explanation scenario, assume a trained (deep) model $f : \mathbb{R}^d \to \mathbb{R}$ with input $\boldsymbol{x} = (x_1, ..., x_d)$ and an output f that provides a prediction score indicating certainty for the presence of a specific feature and for which an explanation is to be computed subsequent to the prediction phase.

One approach to measure what features x_p are most relevant can be performed via perturbation of the input [16, 17, 18]. For this, each feature x_p or groups thereof are occluded and the change in output is measured producing the relevance $R_p = f(\boldsymbol{x}) - f(\boldsymbol{x}_{\setminus p})$. This process is repeated iteratively until a full explanation $\boldsymbol{R} = (R_0, ..., R_d)$ is computed. While this approach does not require a particular structure except being a function, it assumes locality of relevant features, requires repeated model re-evaluation for each perturbation, and is unable to give detailed insight into inner model processing.

Another approach to explainable AI is using surrogate functions that make complex models explainable by replacing the original model function with a local approximation around the current input data. For this, the parameters of a simple self-explainable model are optimized to produce the original model prediction as proposed in Local Interpretable Model-Agnostic Explanations (LIME) [17] and variants thereof [54, 55].

The idea of assigning attribution scores according to a feature's contribution to the observed output prediction has been formalized in the Shapley value [56]. For this, all possible groups of feature combinations are evaluated and thus, the effect of adding or removing a specific feature can be observed via marginalization. With a growing number of features, the possible combinations increase exponentially and different approximation schemes have been proposed to make computations feasible [57, 44, 58]. Based on the Shapley value, the SHAP method has offered a framework

that relies on additive feature importance and approximates Shapley values in the context of explaining classifiers [57].

Alternatively, gradient-based methods compute derivatives of the model function with respect to the input [59], for example, sensitivity analysis for which $R_p = (\partial f(\boldsymbol{x})/\partial x_p)^2$ [20] or saliency given by $R_p = (\partial f(\boldsymbol{x})/\partial x_p)$ [19]. Other popular gradient-based approaches are Gradient × Input [21], which can be seen as special case of LRP, and Integrated Gradients [22]. These methods thus give insight into which parts of the input features, e.g., pixels in images, contribute to an increased or decreased prediction score. While this generally provides a straightforward way to analyze any differentiable machine learning model, the complex structure of deep neural networks can result in unstable and noisy gradients [60, 61], as we will also discuss later in this section.

The complex structure of deep models motivates approaches that focus on a more easily analyzable model decomposition into simpler components. The decomposition of any network neuron can, for example, be performed via a deconvolution process [16] that iteratively projects activity layer-wise back to the input. Instead of observing neuron activity directly, gradients of the activations can be used as the salient signal, as proposed in Guided Backpropagation [24]. This latter approach has been widely adapted and propagation-based explanations are a key concept in computing post-hoc explanations, including Excitation Backpropagation [62] and layer-wise relevance propagation (LRP) [23].

From a theoretical perspective, a direct way to approximate a non-linear model $f(\boldsymbol{x})$ is to use a Taylor decomposition at some root $\widetilde{\boldsymbol{x}}$ [25]:

$$f(\boldsymbol{x}) = f(\widetilde{\boldsymbol{x}}) + \sum_i [\nabla f(\widetilde{\boldsymbol{x}})]_i \, (x_i - \widetilde{x}_i) + \frac{1}{2} \sum_{ii'} [\nabla^2 f(\widetilde{\boldsymbol{x}})]_{ii'} \, (x_i - \widetilde{x}_i)(x_{i'} - \widetilde{x}_{i'}) + \dots \tag{2.1}$$

and choose root points such that $f(\widetilde{\boldsymbol{x}}) = 0$ and higher-order terms vanish. Relevance scores are then directly given by the first-order terms in Equation (2.1). However, resulting relevance scores have been found to be unstable, which mirrors instabilities of deep neural networks. This includes shattered gradients, which refers to the noisy gradient behavior in deep networks [60, 63], and their vulnerability towards small and imperceptible input perturbations that strongly affect the model prediction [61].

Various propagation-based techniques have been proposed to produce more robust explanations, of which we focus on layer-wise relevance propagation in the following.

2.3 Layer-wise Relevance Propagation

Layer-wise relevance propagation (LRP) offers a framework to decompose a possibly complex deep neural network prediction and computes explanations by highlighting relevant data features [23, 25, 8]. To trace back in what ways the features x_p of input $\boldsymbol{x}$ have contributed to the model prediction $f(\boldsymbol{x})$, we are interested in finding relevance

scores R_p that connect the prediction back to the input features $f(\boldsymbol{x}) \approx \sum_p R_p$ with the sum going over input features, i.e., pixels in an image or tokens in a sequence, indexed by p. Relevance R_p indicates positive contribution for $R_p > 0$ and negative contribution for $R_p < 0$, which provides an intuitive way to decompose the network prediction.

An additional desired property requires that relevance propagation in neural networks is conservative such that $f(\boldsymbol{x}) = \sum_p R_p$ holds true. This means that the total sum of relevance remains unchanged throughout the network computations and relevance can not be generated nor reduced. Such layer-wise conservation is easy to implement and implies global conservation of relevance. For deep networks with multiple layers, this conservation can thus be applied to consecutive computations between layer l and $l+1$:

$$f(\boldsymbol{x}) = ... = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = ... = \sum_p R_p^{(0)}. \tag{2.2}$$

We denote the relevance from neurons k at layer $l+1$ as $R_k^{(l+1)}$. Then, the lower level relevance at neuron j can be computed by summing over received messages $R_{j \leftarrow k}^{(l,l+1)}$ from neurons k in the higher layer $l+1$:

$$R_j^{(l)} = \sum_k R_{j \leftarrow k}^{(l,l+1)}. \tag{2.3}$$

The relevance message is generally proportional to the ratio defined by quantities q_{jk}, which is the contribution of neuron j to the activation of neuron k and the relevance observed $R_k^{(l+1)}$:

$$R_{j \leftarrow k}^{(l)} = \frac{q_{jk}}{\sum_j q_{jk}} \cdot R_k^{(l+1)}. \tag{2.4}$$

Finally, the full relevance of neuron j is computed by pooling over all incoming messages $R_j = \sum_k R_{j \leftarrow k}$.

Depending on the network type, layer index and neuron type, different propagation rules have been proposed to compute q_{jk}. These include the $\alpha\beta$-rule that weights positive and negative contributions differently, the w^2-rule that distributes relevance according to the squared weight magnitude, or the γ-rule that favors positive over negative contributions [64]. Here, we focus on the LRP-γ rule given by:

$$R_{j \leftarrow k}^{(l)} = \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_j a_j \cdot (w_{jk} + \gamma w_{jk}^+)} \cdot R_k^{(l+1)}, \tag{2.5}$$

with lower-level neuron activation a_j, the weight w_{jk} between neuron j and k and parameter γ, which controls the preference of positive contributions using the rectified weight w_{jk}^+. Compared to naive computations of relevance scores from first-order

gradients as discussed above, this offers a more robust way to compute relevance redistribution by reducing gradient noise and using more stable gradients [64].

Notably, these redistribution rules can be integrated into the Deep Taylor Decomposition framework (DTD) [25], which considers LRP as a repeated Taylor decomposition of the relevance expressed as a function of lower activations $\boldsymbol{a}$ evaluated at some root point $\widetilde{\boldsymbol{a}}$ [25]:

$$R_k(\boldsymbol{a}) = R_k(\widetilde{\boldsymbol{a}}) + \sum_j \left[\nabla f(\widetilde{\boldsymbol{a}})\right]_j (a_j - \widetilde{a}_j) + \dots. \tag{2.6}$$

In order to arrive at a closed-form solution for equation (2.6) a relevance model $\hat{R}_k(\boldsymbol{a})$ is used to substitute $R_k(\boldsymbol{a})$. Then, one can select $\hat{R}_k$ and $\widetilde{\boldsymbol{a}}$ such that higher-order terms vanish and fulfill $\hat{R}_k(\boldsymbol{a}) = R_k(\boldsymbol{a})$ [25]. Depending on the choice of $\widetilde{\boldsymbol{a}}$, different redistribution rules can be captured and related directly to the mathematical foundation offered by the DTD framework.

In the following chapters, we will encounter settings where the model structure requires consideration of non-vanishing second-order and higher-order terms.

2.4 Developing Propagation Rules

In this thesis, we will develop propagation rules for different model architectures. We typically follow a succession of steps to arrive at an appropriate propagation procedure. While this process can vary for the specific model at hand, the following summarizes the key steps.

- Formalize the model prediction function $f(x)$ and analyze the dependencies of the input x and the general structure to compute activation scores a_j during the layer-wise computations.
- Express the relevance score, i.e., $R_k(\mathbf{a})$ for a simple feed-forward network with consecutive layers j and k, and with the observed activation vector $\mathbf{a}$ at the lower layer j.
- Perform a Taylor-Expansion of the relevance score at some root point $\widetilde{\boldsymbol{a}}$ (cf. Eq. 2.6).
- For a closed-form solution, define a relevance model to substitute the original relevance score, i.e., $R_k(\boldsymbol{a})$ with $\hat{R}_k(\boldsymbol{a})$. This relevance model locally approximates the true function and should be easier to analyze.
- Select an appropriate root point $\widetilde{\boldsymbol{a}}$ based on domain membership and such that $\widetilde{\boldsymbol{a}}$ is close to typical activations and, if possible, the Taylor expansions simplifies.

After arriving at a propagation scheme for the different model layers, their implementation can often be achieved with minimal additional software code. For

example, we can alter the forward computations to match the propagation rules and then perform layer-wise computations by calling the gradient in the backward pass. This does not affect the model predictions or its parameters and leaves the original model behavior unchanged.

2.5 Approaches to Evaluating Explainable AI

The evaluation of explainable AI systems is still considered an open problem but has recently received growing attention [65, 37, 66, 67, 41, 68, 69]. Assessing the quality of explanations has often relied merely on a qualitative inspection. This assumes that humans are able to judge what distinguishes good from bad explanations accurately. Unfortunately, this does not necessarily hold since humans can disagree or might not be able to make sense of the provided explanation [67, 70], unknowingly apply a different reasoning strategy [71, 72], or be influenced by various biases when judging evaluation quality, e.g. representation or confirmation bias [72, 73]. This highlights that when evaluating explanations, it is important to separate approaches that measure how well a method explains the model from approaches that focus on explaining a particular ground truth. For example, an explanation can be able to capture the model prediction very faithfully while not being aligned to some provided ground truth. Conversely, explanations that align well to a ground truth may not reflect the underlying model prediction properly [74, 75, 76].

The many roles and functions of explainable AI are reflected in a wide range of desired criteria an explanation could meet [37, 38, 68]. Common desiderata for explanations include fidelity, understandability, sufficiency, low construction overhead and efficiency [36], which can serve as a guidelines to judge explanations. To address and quantify different desiderata, various approaches have been proposed. In the following, we divide the evaluation of explainable AI systems into two main directions [12, 40]: objective evaluation focused on automated methods to evaluate the quality of explanations and human-centered evaluations that include human or human-annotated material during the evaluation process.

2.5.1 Objective Evaluation

Similar to the different measures used for evaluating model performance, such as cross-entropy loss or accuracy, various evaluation approaches and metrics exist for explanations. These are often designed to measure specific desiderata for explanations. Selecting the appropriate one depends on the task and domain at hand, and can be further constrained by physical limitations, legislative requirements or cognitive characteristics [74].

Common evaluation metric can be grouped into example-based, model-based and attribution-based explanations, with the latter being the most widely adopted approach [77, 68].

Example-based explanations aim to extract specific instances useful to explain the model behavior or illustrate characteristics of the data distribution itself, typically by

extracting prototypes or identifying counterfactual and adversarial instances [78, 79]. To evaluate the explanations quantitatively, the geometric structure, as characterized by the distance among the set of selected instances, has been proposed [80].

Model-based explanations cover ante-hoc approaches, for which the model can be evaluated directly, as well as post-hoc approaches, for which the model is first made explainable and then evaluated. Metrics focus on measures of the model-explanation complexity, i.e., using the model depth or size [81], the number of operations or the number of parameters used to compute a relevant local effect [82], predictor importance [83], and agreement scores between true and explained model [82].

Most widely used approaches consider attribution-based explanations, which order or measure the relevance of input features and determine a relation between input features and importance for the model prediction. We focus on common desiderata and summarize widely used evaluation approaches into the following groups [36, 69]:

- *Faithfulness* (or *fidelity*) measures the effect of a certain (e.g., highly relevant) feature on the model prediction. It is a common desired property of explanations and captures whether the explanation faithfully explains the model. Typical approaches to measure faithfulness include (pixel-)flipping [23] or region perturbation procedures [65], computing correlations between attribution strength and change in model outcome [84], and selectivity of the most relevant features to strongly affect the prediction [8].

- *Complexity* (or *sparsity*) evaluates whether the explanation is able to identify small subsets of features that explain the model prediction [85]. It thus addresses the desired property of understandability (or comprehensiveness) since a more sparse solution is typically more user-understandable. Sparsity can be measured using metrics such as Gini coefficients or entropy scores [86].

- *Sufficiency* aims to measure how well an explanation is able to provide sufficient information about the model prediction process. This is typically formalized using conditional distributions, which measure how likely a certain prediction is to be observed given some feature or set thereof as provided by the explanation [87, 88].

- *Axiomatic* rules can be defined to address specific desired properties of an explanation, usually with the goal to meet certain constraints, such as conservation, invariance properties of explanations, completeness or sensitivity [23, 22].

Furthermore, a variety of methodologies have been proposed to test adequacy of explanations. Examples include *robustness* that aims to measure how strongly explanations vary for nearby data points [89, 8, 90], *randomization* that determines if explanations are able to pass sanity checks by testing if gradual randomization of model parameters produces increasingly different explanations compared to the original [91], and *localisation* that computes to what extent an explanation matches ground truth data [92].

This variety of evaluation approaches and metrics illustrates that choosing an appropriate procedure is complex and future efforts are needed to standardize

the evaluation of explainable AI methods. We note that this even applies to the standard scenario for explanations in image classification. When explanations are specifically structured, e.g., by considering higher-order information, standard evaluation approaches may not be directly applicable, and we will demonstrate how measuring properties such as faithfulness can be adapted to this setting.

2.5.2 Human Evaluation

While automated metrics are crucial to assess explanation quality, eventually, explanations serve as a way to communicate the model decision process to human users. Designing evaluation procedures to test explanations against human ground truth or judgments requires interdisciplinary efforts of machine learning, cognitive science, and associated disciplines. In this section, we provide a brief introduction into key aspects that concern human evaluation of machine learning, and specifically explanations, and refer to following papers and reviews for further reading, i.e., [93, 38, 40].

Explanations can be designed and targeted toward different user groups, and with that, their goals and evaluation procedures can change [94]: machine learning novices including typical end-users tend to focus on model transparency, trustworthy explanations and avoiding biased predictions, computer experts are additionally interested in visualization and inspection of mechanisms as well as fine-tuning and selecting their own models, and machine learning experts focus also on making models interpretable to mitigate identified model misbehavior.

Human evaluation studies have been designed to judge various aspects of the human-machine interaction to attain these goals. To increase user confidence, trust and reliability in expert systems, deep neural network models and intelligent agents have been evaluated using human judgments [95, 40]. These qualities require an understanding of model mechanisms, which has been studied by accessing if provided explanations enable or facilitate the development of an internal mental model [96, 97, 98, 99]. Additionally, it was found that certain characteristics of an explanation are judged as preferred, e.g., humans prefer fewer and simpler explanations over more complex ones [98]. It has been studied which explanations are useful to detect wrong model strategies and thus help to debug machine learning models by users [100, 101, 102], and to determine their usefulness for task solving in work environments [96, 103].

In order to evaluate these different functions of human evaluations, a range of different approaches are commonly used. These include computing statistical analyses on closed-ended questionnaires using pre-defined scales [96, 104, 95], collecting judgments from qualified users about the accuracy of explanations and how typical the explanation is given the sample [105, 106], and asking humans to infer the predicted class from the explanation, and record their accuracy and confidence levels [107, 108, 44, 109, 110]. To test whether an explanation aids the user in building a causal understanding, the System Causability Scale aims to measure specific properties of an interaction between humans and an explanation system,

such as effectiveness, efficiency, and user satisfaction in a given scenario [111]. Other works have focused on presenting humans with the input data and ground truth class and letting them annotate the relevant parts to create a human explanation [108, 22, 112, 75], measuring psychophysical signals during task-solving to compare the model generated explanations against human rationales [113, 114, 115, 116, 117], and testing if biasing models towards human rationales improves and robustifies model performance [118, 115, 114, 119, 120].

Besides quantitative approaches, qualitative assessment of evaluation quality using open-ended questions and interviews or recordings of human-machine interaction have been conducted [40].

These studies highlight that using human signals in machine learning and, in particular, explainable AI is a promising direction that aims to promote successful human-machine interactions. Human-centered studies have addressed many aspects of model explanations, most notably focusing on their alignment to human explanations and their predictions, as well as the quantifiable benefit of explanations regarding user trust towards models. In this thesis, we will use a dataset of human task-reading patterns to compare human rationales against language model explanations.

2.6 Limitations and Challenges

Throughout this thesis, we will encounter various limitations and challenges of modern machine learning models. Here we summarize key aspects and challenges relevant to this thesis in the context of explainable AI and structured explanations.

2.6.1 Explanation Complexity

Typical explainable AI methods explain model processing in terms of model inputs, e.g., heatmaps over images or input sequences. This attribution of the prediction to the input features does not necessarily capture the full complexity of processing in a machine learning model. It thus motivates approaches that go beyond heatmap explanations. Consequently, the question of the appropriate explanation complexity arises. As a guideline, the explanation should be complex enough to capture the model and data structure appropriately without obfuscating relevant processing steps while at the same time selecting a level of abstraction that is as simple as possible and can be understood by the user. For example, pairwise feature interactions in similarity models can be considered for a more detailed understanding of which features are relevant for the similarity prediction, as we will see in Chapter 3.

Standard explanation methods aim to represent the reasoning process of a model with the goal of producing one single explanation. Alternatively, it has been argued that explanation complexity should reflect the reasoning process of human users, i.e., novices or domain experts, which result in different explanations of varying explanation complexity [121, 122, 123].

This thesis will focus on producing explanations for different types of complex models. As we will see, some specific model structures are more accurately described

as explanations of higher complexity. We will additionally see how a resulting complex explanation can be represented in simpler ways, e.g., via visualization or pooling strategies.

2.6.2 Limitations of Post-Hoc Explanations

It has been argued that post-hoc explanations can produce misleading explanations [30, 74]. This has been demonstrated by comparing various post-hoc explanation approaches and observing that a range of methods can produce different and sometimes even contradictory or paradoxical explanations for the same model [124, 74].

Not all methods answer the same question and thus different outcomes are to be expected, i.e., sensitivity methods trace which input features change the model prediction score most, whereas LRP computes which input features are most relevant for a given classification score and thus explains the actual prediction. A similar visual representation, e.g., in heatmaps, can be misleading without additional information about the underlying principles and the specific question that an explanation method addresses.

In addition, not all post-hoc explanation methods are equally robust and faithful to the model prediction. For example, it was shown that naive input gradients become increasingly variable and unreliable for deeper models [60, 63], and that their ranking in evaluation settings can differ considerably, e.g., when evaluating faithfulness [65, 125]. Hence, post-hoc explanation methods approximate a model's prediction process and their ability to capture this process has to be evaluated thoroughly.

Post-hoc explanations are designed to explain the model function and thus a successful explanation does not necessarily align with ground-truth observations. This can result in faithful explanations that contradict the user's real-world intuition. Machine learning models infer a relation between the presented training data and the respective labels. This process can identify a correlation between observation and outcome but does not indicate a causal relationship. Expressed differently, a feature can correlate strongly with a certain class, but this does not indicate this class is the cause for the feature's appearance. Such effects are known as *spurious* correlations that describe relationships not linked via a causal relationship [126, 127, 128]. As we will see in the next section, machine learning models can adopt unexpected strategies.

2.6.3 The 'Clever Hans' Effect

The various ways that modern machine learning models can bias their predictions in unexpected ways are known as 'Clever Hans' strategies [30], 'anti-causal learning' [129], or 'shortcut learning' [130]. This effect has been observed in many real-world scenarios that achieve human-level performance but fail in employing correct prediction strategies [29, 30, 35, 33, 34]. This severely hinders the safe and reliable use of machine learning. To deal with the 'Clever Hans' effect, we need to detect such misbehavior and identify strategies to overcome them in a subsequent step.

The diagnosis of 'Clever Hans' effects is a highly relevant research direction for which different solutions have been proposed. Standard machine learning pipelines evaluate model performance on test sets that are from the same data distribution as the training set, which can result in unintended model solutions if the data distribution is not sufficiently general. This motivates the use of additional out-of-domain datasets to test generalization capabilities [130] and thus a low model performance of the latter can indicate the presence of a 'Clever Hans'-type behavior. A different approach is the use of explainable AI to extract relevant features that a model prediction is based on. This offers a way for the user to inspect and control model learning strategies and flag unwanted examples. More recently, the automated and dataset-wide detection of 'Clever Hans' predictors has been explored [30, 128].

Resolving such invalid strategies is an important challenge, and promising directions concern the choice of model architecture, the training data and its distribution as well as the optimization loss function itself. But it is generally not easy to understand which changes can resolve a 'Clever Hans' effect. For example, adding more training data does not necessarily improve model generalization [131, 132]. Instead, different approaches have been proposed to either unlearn identified artifacts in an already trained model [133, 134, 135, 128], or to regularize models during training, for which an additional learning signal about the desired explanation is provided [136, 137, 138].

3

Second-order Explanations in Building and Interpreting Deep Similarity Models

Many learning algorithms such as kernel machines, nearest neighbors, clustering, or anomaly detection, are based on distances or similarities. To safely use similarities for training machine learning models or applications in downstream tasks, we would like to verify that they are bound to meaningful data patterns. We make similarities interpretable by providing detailed explanations that connect the observed similarity to the interaction between relevant features in the data. 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 multiple explanations computed using layer-wise relevance propagation (LRP), which was shown to scale to highly nonlinear models. In our experiments, we demonstrate that BiLRP robustly explains complex similarity models, e.g., built on VGG-16 deep neural network features. Then, we turn to an open problem in the digital humanities and build a model for the detailed assessment of similarity between historical documents such as astronomical tables. For this highly engineered and problem-specific similarity model, BiLRP provides transparency and brings verifiability, which enables us to draw corpus-level historical insights.

This chapter is based on the following works and partly includes already published material from:

[139] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani and G. Montavon. Building and Interpreting Deep Similarity Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1149-1161, 2022.

[140] O. Eberle, J. Büttner, H. El-Hajj, G. Montavon, K.-R. Müller and M. Valleriani. From Large Data Collections to Historical Insights: A Machine Learning Analysis of Astronomical Tables in Early Modern Textbooks. *(in preparation)*, 2022.

3.1 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 [141, 142]. Good similarity models are needed to find relevant items in databases [143, 144, 145]. Similarities (or kernels) are also the starting point of many machine learning models, including discriminative learning [146, 147], unsupervised learning [148, 149, 150, 151], and data embedding and visualization [152, 153, 154].

An important practical question is how to select the similarity model appropriately. Assembling a labeled dataset of similarities for validation can be difficult since the labeler would need to inspect meticulously multiple pairs of data points and assign exact real-valued similarity scores. As an alternative, selecting a similarity model based on performance on some proxy task can be convenient, e.g., [155, 156, 157, 158]. In both cases, however, the selection procedure is exposed to a potential lack of representativity of the training data, which can result in 'Clever Hans' effects (cf. Section 2.6.3). Here, we aim for a more direct way to assess similarity models and use explainable AI to identify the data features that support the similarity prediction.

In the following, we bring explainable AI to similarity and consider similarity models of the type: $y(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x}) \,,\, \phi_L \circ \cdots \circ \phi_1(\boldsymbol{x}') \rangle$, i.e., 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., [23, 159], cannot be directly applied since they typically assume some form of local *linearity* whereas dot products have *bilinearity*. This motivates the here used explanation method that adapts to this new setting. Our method, named 'BiLRP', is illustrated in Fig. 3.1. 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.

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 [25]. Specifically, the procedure can be expressed as a collection of second-order

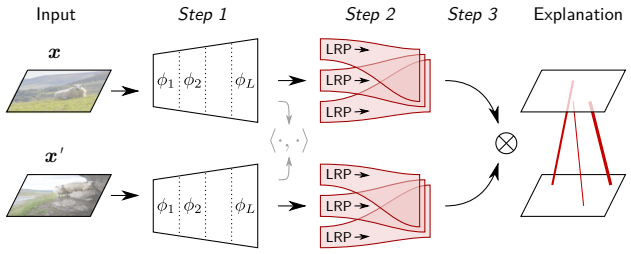


Figure 3.1: The BiLRP method for explaining similarity. Resulting explanations indicate the relevant interaction in terms of *pairs* of input features.

Taylor expansions performed in each layer. Elements of these expansions identify the exact layer-wise redistribution strategy. Additionally, BiLRP can be interpreted as building a robustified Hessian of the similarity model at every layer. This procedure allows us to 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 [160]. Our explanation method brings useful insights into the strengths and limitations of each similarity model. In addition, we illustrate how the insights brought by BiLRP can inform the development of an improved similarity model. We then move to an open problem in the digital humanities that aims to infer similarities between tables and extract detailed historical relations between them [161]. This enables us to study historical processes and knowledge evolution at scale. For this, we build a similarity model that is not built from standard pretrained features and instead uses *engineered* features that are task-specific and able to handle the high heterogeneity of historical corpora. Again, BiLRP proves useful by inspecting the similarity model and validating it from limited data. Altogether, the BiLRP method brings transparency into similarity, which is a key ingredient of machine learning, and paves the way for the systematic design and validation of similarity-based machine learning models in an efficient, fully informed, and human-interpretable manner.

3.1.1 Related Work

To gain insights into the similarity structure of large datasets, methods such as LLE [162], diffusion maps [163], or t-SNE [154] provide directly interpretable embeddings by projecting 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. We focus here on approaches that have addressed explaining individual predictions, considering joint feature interactions, and applications of similarity in machine learning models.

Individual Predictions The question of explaining individual predictions has been extensively studied in the context of machine learning classifiers. Methods based on occlusions [16, 18], surrogate functions [159, 44], and gradients [59, 19, 164, 22] have been proposed. Other approaches have used linear combinations of latent feature representations [165], or reverse propagation [23, 16] to highlight the most relevant features. Some approaches have been extended to unsupervised models, e.g., anomaly detection [166, 167] and clustering [26], and attention models have also been developed to explain tasks different from classification, such as image captioning [168] or similarity [169]. Our work goes further in this direction and explains similarity built on general neural network models by identifying relevant pairs of input features.

Joint Feature Explanations Several methods for explaining model predictions using joint features have been proposed. Some of them extract feature interactions globally [170, 171]. Other methods produce individual explanations for simple pairwise matching models applied to the input features [172], or to activation maps of a convolutional network [173]. Other methods incorporate explicit multivariate structures into the model to identify joint contributions [174], or compute pairwise interaction effects using cross derivatives [175]. Estimating the integral of the Hessian has been proposed as another method to extract joint feature explanations in nonlinear models [176]. In comparison, our BiLRP method leverages the deep layered structure of the model to robustly explain predicted similarity in terms of input features.

Applications of Similarity in Models Several works improve similarity models by leveraging prior knowledge or ground truth labels. Proposed approaches include structured kernels [177, 141, 178, 179], and siamese or triplet networks [180, 181, 182, 183, 184]. Beyond similarity, collaborative filtering [185], transformation modeling [186], and information retrieval [187] also rely on building high-quality matching models between pairs of data. We follow an orthogonal objective here since we assume an already trained, well-performing similarity model and aim to make it explainable to enhance its verifiability and enable the extraction of insights from it.

3.2 Towards Explaining Similarity

In this section, we present approaches to explain the predictions of an already trained similarity model in terms of its input features. 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:

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

and identifying the elements of the sum as the respective contributions. We observe that input features interact to produce a high or low similarity score.

In practice, more accurate models of similarity can be obtained by relaxing the linearity constraint. Consider a 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 [23] at some reference point $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')$:

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

Here, ∇^2 denotes the Hessian. The explanation is obtained by identifying the multiple terms of the expansion. As 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 close to zero leads to a simplified 'Hessian $\times$ Product' formulation:

$$y(\boldsymbol{x}, \boldsymbol{x}') = \textstyle\sum_{ii'} [\nabla^2 y(\boldsymbol{x}, \boldsymbol{x}')]_{ii'} \, x_i \, x'_{i'}, \tag{3.1}$$

where second-order contributions can be directly computed. We will use this Hessian-based formulation as a baseline method in the evaluation experiments.

3.3 Explaining Similarity with BiLRP

In the following, we introduce the BiLRP method for explaining similarities. It is based on combining second-order Taylor expansions for producing explanations in terms of pairs of input features and the layer-wise relevance propagation (LRP) [23] technique that robustly explains complex deep neural network predictions [8, 188] (cf. Section 2.3).

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

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

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. We note, the same network can also be written as a single network $y(\boldsymbol{x}, \boldsymbol{x}') = \psi_L \circ \dots \circ \psi_1(\boldsymbol{x}, \boldsymbol{x}')$ where ψ summarizes the two branches of the computation. Then, inspired by LRP, the BiLRP method applies a message passing procedure from the top layer where

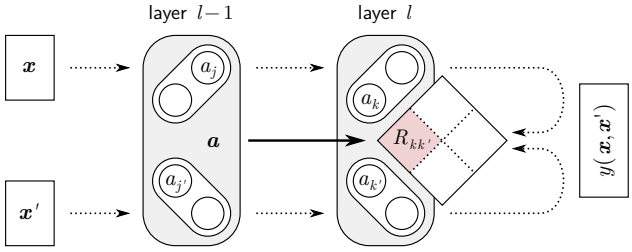


Figure 3.2: Annotated diagram to illustrate the map used by DTD to derive BiLRP propagation rules. The map connects activations at some layer to relevance in the layer above.

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.

We next describe how these messages are obtained from second-order Taylor expansions in Section 3.3.1. We discuss the theoretical properties of BiLRP in Section 3.3.2 and demonstrate how it can be interpreted as building a robustified Hessian of the similarity model. Finally, Section 3.3.3 shows how BiLRP can be computed in a way that makes use of multiple LRP computations, thereby considerably easing implementation.

3.3.1 Deriving BiLRP Propagation Rules

To build meaningful propagation rules, we use the 'deep Taylor decomposition' (DTD) [25] framework that consists of applying Taylor expansions at each layer to identify how redistribute the prediction to the layer below.

Assume we have already run a few steps of propagation starting from the output up to 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. The relation between these two quantities is depicted in Fig. 3.2. Then, DTD seeks to perform a Taylor expansion of the function $R_{kk'}(\boldsymbol{a})$ at some reference point $\widetilde{\boldsymbol{a}}$:

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

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 [25]. Assume that relevance propagated up to a certain layer can be modeled as

$$\widehat{R}_{kk'}(\boldsymbol{a}) = a_k a_{k'} c_{kk'},$$

i.e., a product of activations in the two branches of the similarity computation, multiplied by a term $c_{kk'}$ assumed to be constant and set in a way that $\widehat{R}_{kk'}(\boldsymbol{a}) = R_{kk'}$. This relevance model is also justified later in Proposition 3. Then, DTD seeks to propagate the modeled relevance to the layer below by identifying the terms of a Taylor expansion. In the following, we distinguish between (i) linear/ReLU layers, and (ii) positively homogeneous layers (e.g. min- or max-pooling). For each case, we first specify the specific relevance model, select and analyze appropriate root point choices and finally arrive at the layer propagation rule.

Linear/ReLU Layers

For linear/ReLU layers [189], the relevance model can be written as:

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

When neurons a_k and $a_{k'}$ are jointly activated (i.e. $a_k, a_{k'} > 0$), a second-order Taylor expansion of $R_{kk'}$ at some reference point $\widetilde{\boldsymbol{a}}$ is given by:

$$\begin{aligned}\widehat{R}_{kk'}(\boldsymbol{a}) = &\big(\textstyle\sum_j \widetilde{a}_j w_{jk}\big)\big(\textstyle\sum_{j'} \widetilde{a}_{j'} w_{j'k'}\big) c_{kk'} \\ &+ \textstyle\sum_j (a_j - \widetilde{a}_j) w_{jk} \big(\textstyle\sum_{j'} \widetilde{a}_{j'} w_{j'k'}\big) c_{kk'} \\ &+ \textstyle\sum_{j'} \big(\textstyle\sum_j \widetilde{a}_j w_{jk}\big)(a_{j'} - \widetilde{a}_{j'}) w_{j'k'} c_{kk'} \\ &+ \textstyle\sum_{jj'} (a_j - \widetilde{a}_j) w_{jk} (a_{j'} - \widetilde{a}_{j'}) w_{j'k'} c_{kk'}.\end{aligned}$$

BiLRP chooses the reference point $\widetilde{\boldsymbol{a}}$ to be subject to the following two constraints:

1. The point should be very close to the ReLU hinges of neurons k and k' (but still on the activated domain)
2. The point should lie on the plane $\{\widetilde{\boldsymbol{a}}(t,t') |\ t, t' \in \mathbb{R}\}$ 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 γ a hyperparameter.

We can now analyze the different terms of the expansion at this reference point and see that the zero-order term vanishes, and the first-order terms are also zero because the reference point is chosen at the *intersection* of the ReLU hinges of neurons k and k', hence the non-differentiated term is zero. The interaction terms are given by $R_{jj'\leftarrow kk'} = tt' a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'}) c_{kk'}$ with $\rho(w_{jk}) = w_{jk} + \gamma w_{jk}^+$ with the remaining product of parameters tt' still to be resolved. Because we expand a bilinear form, and since zero-order and first-order terms vanish, the constraint $\sum_{jj'} R_{jj'\leftarrow kk'} = R_{kk'}$ must be satisfied. This constraint allows us to resolve the product tt', leading to the following closed-form expression for the interaction terms:

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

This propagation rule is also consistent with the case where a_k or $a_{k'}$ are zero and where no relevance needs to be redistributed. Aggregate relevance scores for the layer below are obtained by summing over neurons in the higher layer:

$$\begin{aligned} R_{jj'} &= \textstyle\sum_{kk'} R_{jj'\leftarrow kk'} \\ &= \sum_{kk'} \frac{a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} R_{kk'}. \end{aligned} \tag{3.2}$$

This last equation is the propagation rule used by BiLRP to propagate relevance in linear/ReLU layers. This propagation rule can be seen as a second-order variant of the LRP-γ rule [64] used for explaining DNN classifiers. From this rule it follows that 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.

Positively Homogeneous Layers

When a_k and $a_{k'}$ are positively homogeneous functions of their input activations, i.e., min- and max-pooling layers, the relevance model can be expressed in terms of the Hessian:

$$\begin{aligned} \widehat{R}_{kk'}(\boldsymbol{a}) &= a_k a_{k'} c_{kk'} \\ &= (\textstyle\sum_j a_j [\nabla a_k]_j)(\sum_{j'} a_{j'} [\nabla a_{k'}]_{j'}) c_{kk'} \\ &= \textstyle\sum_{jj'} a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'} c_{kk'}. \end{aligned}$$

The last form can also be interpreted as the interaction terms of a Taylor expansion of $\widehat{R}_{kk'}$ at $\widetilde{\boldsymbol{a}} = \varepsilon \boldsymbol{a}$ with ε almost zero. Zero-order and first-order terms of the expansion

vanish, and interaction terms can be rewritten in a propagation-like manner as:

$$R_{jj' \leftarrow kk'} = \frac{a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}}{\sum_{jj'} a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}} R_{kk'},$$

and finally, we arrive at the BiLRP propagation rule for positively homogeneous layers

$$R_{jj'} = \sum_{kk'} \frac{a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}}{\sum_{jj'} a_j a_{j'} [\nabla^2 a_k a_{k'}]_{jj'}} R_{kk'}. \tag{3.3}$$

This propagation rule has a similar interpretation to the previous case. 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.

3.3.2 Theoretical Properties of BiLRP

An important property of LRP [23] is conservation, i.e., the relevance scores assigned to the input features sum to the prediction output[1]. Similar results can be obtained for BiLRP.

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

Proof. We first show conservation when propagating with Eq. (3.2) in a linear/ReLU layer:

$$\begin{aligned}\sum_{jj'} R_{jj'} &= \sum_{jj'} \sum_{kk'} \frac{a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} R_{kk'} \\ &= \sum_{kk'} \frac{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} R_{kk'} = \sum_{kk'} R_{kk'}.\end{aligned}$$

The same conservation property can be shown for the propagation rule in Eq. (3.3). Because these rules are applied repeatedly at each layer, we get the chain of equalities where we observe that conservation also holds globally:

$$\sum_{ii'} R_{ii'} = \cdots = \sum_{jj'} R_{jj'} = \sum_{kk'} R_{kk'} = \cdots = y(\boldsymbol{x}, \boldsymbol{x}').$$

A result due to [21] is that application of a special case of LRP (referred by [64] as LRP-0, or LRP-γ with $\gamma = 0$) at each layer of the network produces an explanation that is equivalent to Gradient $\times$ Input. A similar result can be shown for BiLRP.

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

[1] In LRP, exact conservation requires using non-dissipative propagation rules (e.g. LRP-0 and LRP-γ), as well as avoiding contribution of biases (e.g. by training a model with biases set to zero).

Proof. Relevance scores in linear/ReLU layers can be rewritten as $R_{jj'} = a_j a_{j'} c_{jj'}$ and $R_{kk'} = a_k a_{k'} c_{kk'}$ and observing that for $\gamma = 0$, we have $\rho(w_{jk}) = w_{jk}$, the propagation from one layer to another can be written for Eq. (3.2) as:

$$\begin{aligned} c_{jj'} &= \sum_{kk'} w_{jk} w_{j'k'} \frac{a_k}{\sum_j a_j w_{jk}} \frac{a_{k'}}{\sum_{j'} a_{j'} w_{j'k'}} c_{kk'} \\ &= \sum_{kk'} w_{jk} w_{j'k'} 1_{a_k>0} 1_{a_{k'}>0} c_{kk'} \\ &= \sum_{kk'} [\nabla a_k]_j [\nabla a_{k'}]_{j'} c_{kk'}, \end{aligned}$$

and similarly for Eq. (3.3) as:

$$\begin{aligned} c_{jj'} &= \sum_{kk'} [\nabla^2 a_k a_{k'}]_{jj'} c_{kk'} \\ &= \sum_{kk'} [\nabla a_k]_j [\nabla a_{k'}]_{j'} c_{kk'}. \end{aligned}$$

For the considered class of functions, this relation is equivalent to the formula for propagating second-order derivatives (cf. [190]), where $c_{jj'}$ and $c_{kk'}$ denote $[\nabla^2 y]_{jj'}$ and $[\nabla^2 y]_{kk'}$ respectively. Hence, we finally arrive at the quantity $c_{ii'} = [\nabla^2 y]_{ii'}$ and therefore $R_{ii'} = x_i x'_{i'} c_{ii'}$ is equivalent to 'Hessian $\times$ Product'. 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 3.5 and 4.5 that non-zero values of γ give better explanations.

We highlight in the following the product structure of relevance scores produced by BiLRP at each layer. The modeling of $c_{jj'}, c_{kk'}, \ldots$ as constant leads to easily analyzable relevance models from which the BiLRP propagation rules an be derived.

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.*

Proof. In the top layer, we have $c_{kk'} = 1_{\mathrm{id}(k)=\mathrm{id}(k')}$, which is constant and where 'id' is a function returning the neuron index in its respective branch. Applying an inductive argument, assume that at some layer, $c_{kk'}$ is locally approximately constant, we would like to show that the same holds for $c_{jj'}$ in the layer below. Relevance scores in Eq. (3.2) can be rewritten as $R_{jj'} = a_j a_{j'} c_{jj'}$ with

$$c_{jj'} = \sum_{kk'} \rho(w_{jk}) \rho(w_{j'k'}) \frac{\left(\sum_j a_j w_{jk}\right)^+ \left(\sum_{j'} a_{j'} w_{j'k'}\right)^+}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} c_{kk'}.$$

The term $c_{jj'}$ depends on a_j and $a_{j'}$ only through nested sums, which can be seen as diluting the effect of these activations, and the term $c_{kk'}$ that we have assumed as a starting point to be locally approximately constant. Similarly,

for Eq. (3.3), the redistributed relevance can be written in product form using $c_{jj'} = \sum_{kk'} [\nabla^2 a_k a_{k'}]_{jj'} c_{kk'}$. This time, $c_{jj'}$ depends on local activations through a combination of a nested sum and a second-order differentiation, with the same diluting effect as above, and the term $c_{kk'}$ which is locally approximately constant. Overall, in both cases, the weak dependency of $c_{jj'}$ on local activations provides support for treating this term as constant in the relevance model used by DTD.

3.3.3 BiLRP as a Composition of LRP Computations

A limitation of a direct application of the propagation rules of Section 3.3.1 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, the relevance scores at each layer can also written in the factored form:

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

where h is the dimension of the top-layer feature map. The proof relies on the insight that the dot product similarity at the top layer can be rewritten in a factored form, and from which the above relevance scores can then be factorized [139]. Then, the factors can be computed iteratively as:

$$R_{jm} = \sum_k \frac{a_j \rho(w_{jk})}{\sum_j a_j \rho(w_{jk})} R_{km} \tag{3.4}$$

for linear/ReLU layers, and

$$R_{jm} = \sum_k \frac{a_j [\nabla a_k]_j}{\sum_j a_j [\nabla a_k]_j} R_{km} \tag{3.5}$$

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 3.3.1.

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 4.5, 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 (3.4) and (3.5) correspond to common rules used by standard LRP. The first one is equivalent to the LRP-γ rule [64] used in convolution/ReLU layers of DNN classifiers. The second one corresponds to the way LRP commonly handles pooling layers [23]. These propagation rules apply independently on each branch of the similarity model. Thus, BiLRP can

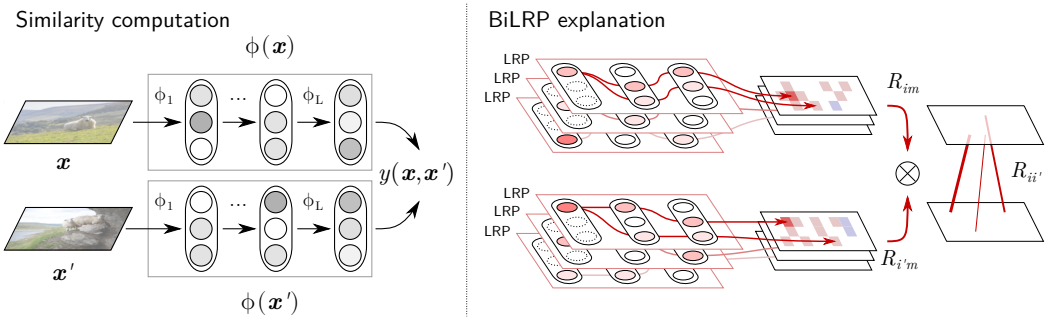


Figure 3.3: Illustration of our approach to compute BiLRP explanations. *Left*: In the forward pass of the neural network, input examples are mapped until the layer at which the similarity model is built. *Right*: One pass of LRP is computed for each neuron activation in this layer during the backward pass. The resulting relevance arrays from each branch are recombined into a single explanation of predicted similarity that contains the relevance scores for each feature interaction.

be implemented as a combination of a series of LRP computations that are then recombined at the input layer:

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

This modular approach to compute BiLRP explanations is presented in Fig. 3.3. BiLRP can therefore be easily and efficiently implemented based on existing explanation software. We note that the modular approach described here is not restricted to LRP and other explanation techniques could be used in the composition to compute the respective relevance maps. This would however lose the interpretation of the explanation procedure as a deep Taylor decomposition.

3.4 Visualization of BiLRP Explanations

To make the additional information of explanations that capture feature interactions human-interpretable, we have developed visualization approaches described in the following. The relevant features that contribute to the similarity prediction can be described by the polarity and magnitude between a pair of features as well as their respective location. In scenarios for which the number of features is small, BiLRP explanations can be visualized directly by representing relevant feature interactions using connecting lines between features. Polarity can be indicated by coloring the line in red for positive relevance scores, and in blue for negative scores. The magnitude of these scores can be rendered using an opacity parameter α. However, the number of possible connections grows quadratically with the number of features and we use the following two approaches, coarse-graining and a specific rendering approach, to reduce the visual explanation complexity and to make them easier to comprehend.

3.4.1 Coarse-Grained Explanations

With increasing input dimensions d, the resulting BiLRP explanation of size d^2 can grow quite large. In practice, similarity does not necessarily need to be attributed to every single pair of pixels or input dimensions. A coarse-grained explanation in terms of groups of features jointly representing a super-pixel, a character, or a word, is often sufficient. Let $(\mathcal{I}_1, \mathcal{I}_2, \dots)$ and $(\mathcal{I}'_1, \mathcal{I}'_2, \dots)$ be two partitions of features for the two input examples $\boldsymbol{x}$ and $\boldsymbol{x}'$. These partitions form the coarse-grained structure in terms of which we would like to produce an explanation. Coarse-grained relevance scores are then given by:

$$R_{\mathcal{I}\mathcal{I}'} = \sum_{i\in\mathcal{I}} \sum_{i'\in\mathcal{I}'} R_{ii'}.$$

When the original explanation is conservative, it can be verified that the same holds for the coarse-grained explanation ($\sum_{\mathcal{I}\mathcal{I}'} R_{\mathcal{I}\mathcal{I}'} = \sum_{\mathcal{I}\mathcal{I}'} \sum_{ii'\in\mathcal{I}\mathcal{I}'} R_{ii'} = \sum_{ii'} R_{ii'}$).

3.4.2 Rendering Explanations

When rendering all lines in the explanation tensor, it can be difficult to separate the typically many low-relevant lines from the few highly-relevant ones. In order to further reduce the visual complexity of the explanation scores, we set $\alpha = 0$ for zero or near zero relevance scores, and $\alpha = 1$ for the largest scores and render the explanation as described in Algorithm 1.

Algorithm 1 Rendering of BiLRP explanations

$R_{\mathcal{I}\mathcal{I}'} \leftarrow \sum_{i\in\mathcal{I}} \sum_{i'\in\mathcal{I}'} R_{ii'}$ (coarse-graining)
$R_{\mathcal{I}\mathcal{I}'} \leftarrow R_{\mathcal{I}\mathcal{I}'} / \sqrt[4]{\mathbb{E}[R^4_{\mathcal{I}\mathcal{I}'}]}$ (normalization)
$R_{\mathcal{I}\mathcal{I}'} \leftarrow R_{\mathcal{I}\mathcal{I}'} - \text{clip}(R_{\mathcal{I}\mathcal{I}'}, [-l, l])$ (sparsification)
$\Delta = h - l$
$R_{\mathcal{I}\mathcal{I}'} \leftarrow \text{clip}(R_{\mathcal{I}\mathcal{I}'}, [-\Delta, \Delta])/\Delta$ (thresholding)
for $R_{\mathcal{I}\mathcal{I}'} \neq 0$ **do**
  $\alpha = |R_{\mathcal{I}\mathcal{I}'}|^p$ (set opacity)
  if $R_{\mathcal{I}\mathcal{I}'} > 0$ **then**
    connect$(\mathcal{I}, \mathcal{I}', \text{red}, \alpha)$
  else
    connect$(\mathcal{I}, \mathcal{I}', \text{blue}, \alpha)$
  end if
end for

The procedure pools relevance scores on super-pixels, normalizes them, shrinks them so that only a limited number of connections need to be plotted, thresholds them so that they fit into a finite color space, and raises them to some power p. The parameter l controls the level of sparsification and we tune it mostly for computational reasons. The parameter h forces all scores beyond a certain range to be plotted to the maximum opacity value. The parameter p lets the explanation focus on all or

the highest relevance scores. A large value for p makes the visualization more easily interpretable, however contributions to similarity that are spread to a larger group of input features can become visually imperceptible.

3.5 Evaluation of BiLRP

This section tests the ability of the BiLRP method to produce faithful explanations. Generally, ground-truth explanations of machine learning predictions, especially nonlinear ones, are hard to acquire [8, 66]. Thus, we consider a synthetic scenario to evaluate whether the correct feature interactions are highlighted by the explanation method. We use a hard-coded similarity model and train a neural network to match the predictions of the former exactly. The structure of the hard-coded model is chosen such that extract ground-truth explanations in the form of relevant interacting features can be obtained. After training, the extracted explanations from the neural network should match for a faithful explanation method. This setting allows us to obtain ground-truth explanations to evaluate BiLRP against baseline methods.

Setup The hard-coded similarity model takes two random sequences of 6 digits as input and counts the number of matches between them. The matching elements between the two sequences form the ground truth explanation. The neural network receives an input matrix $\boldsymbol{x} \in \mathbb{R}_+^{6\times 10}$ with each row representing the encoding of a digit in the form of a vector $\mathbb{R}_+^{10}$. We introduce correlation between the digit vectors to make the task more difficult and avoid undesired solutions, e.g., memorizing input sequences. The input is then fed through two hidden layers of size 100 and a top layer of size 50 that computes the output feature map. The network is trained for $10,000$ iterations of stochastic gradient descent to minimize the mean square error between predictions and ground-truth similarities. After training, the neural network can solve the task perfectly with a final error of 10^{-3}.

Benchmark Methods 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$\times$Product': $R_{ii'} = x_i x'_{i'} \, [\nabla^2 y(\boldsymbol{x}, \boldsymbol{x'})]_{ii'}$

Results 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 to form a 6×6 matrix connecting the digits from the two sequences as introduced in Section 3.4.1. Results are shown in Fig. 3.4. A better matching between the ground truth data and the produced connectivity pattern indicates a better explanation method, which we measure using the average cosine

similarity (ACS). High scores are shown in red, low scores in light red or white, and negative scores in blue.

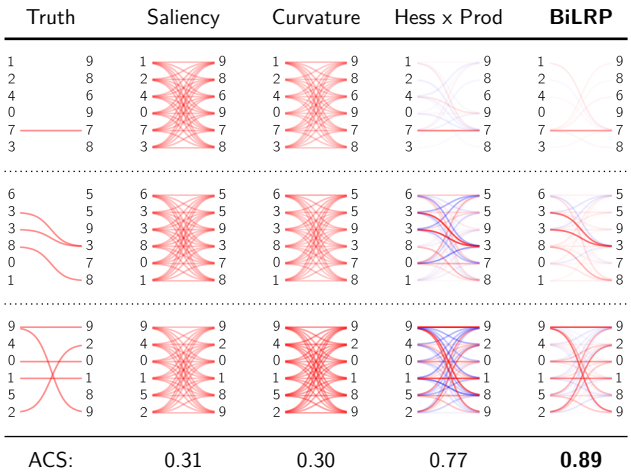


Figure 3.4: Benchmark comparison on synthetic data, which offers ground-truth similarity explanations. The ability of an explanation method to match this ground truth is measured using the average cosine similarity (ACS).

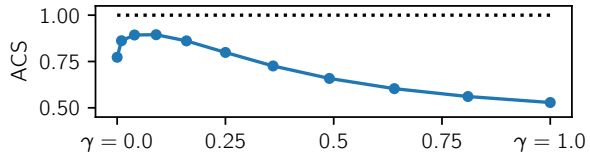


Figure 3.5: Effect of the BiLRP parameter γ on the average cosine similarity between the explanations and the ground truth.

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. Although sensitive to the output, the 'Curvature' baseline does not improve over Saliency. The 'Hessian $\times$ 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. By choosing an appropriate BiLRP parameter γ (here set to 0.09), these negative contributions are considerably reduced.

This visual inspection is validated quantitatively by considering a large number of examples and computing the ACS between the produced explanations and the ground truth with an ACS of 1.0 indicating a perfect match with the ground truth. 'Saliency' and 'Curvature' baselines have low ACS. The accuracy is strongly improved by 'Hessian $\times$ Product' and further improved by BiLRP.

The effect of the parameter γ of BiLRP on the ACS score is shown in Fig. 3.5. 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 [191, 64].

3.6 Interpreting Deep Similarity Models

Next, we will use BiLRP to gain insight into more complex similarity models that are built on the well-established VGG-16 convolutional neural network [160] that was trained on the task of object classification.[2]

Setup We build the similarity model on the output activation at a particular processing layer and compute a dot product on the neural network activations by computing

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

using layer 31 that corresponds to the last layer of features before the classification step. The mapping from input to layer 31 is a sequence of convolution/ReLU 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 add a random projection layer after the last 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 $z^{\mathcal{B}}$-propagation rule, which specifically handles the pixel domain [25]. Finally, we apply an 8×8 pooling on the output of BiLRP to reduce the size of the explanations. Visualization parameters are given in Appendix A.1.

Results Figure 3.6 (A-F) shows our BiLRP explanations on a selection of image pairs taken from the Pascal VOC 2007 dataset [192] 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 for a cat. 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 is 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 a sheep's head. In Example E, while we expect the matching to occur between the two large animals in the image, the true reason for the similarity is

[2] Demonstration code for implementing BiLRP is available at: `https://github.com/oeberle/BiLRP_explain_similarity`

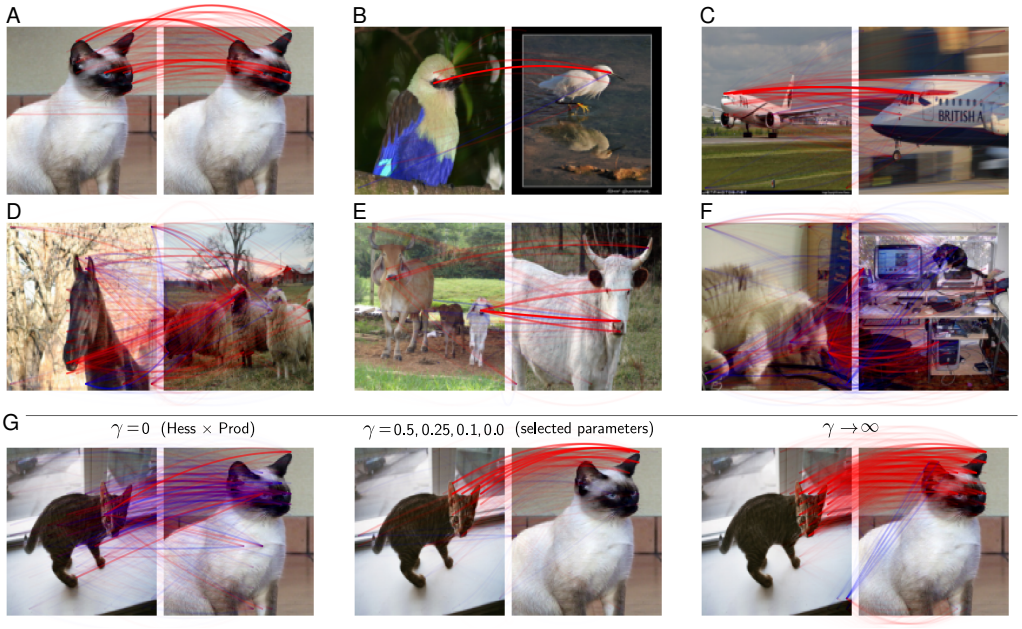


Figure 3.6: Application of BiLRP to a dot product similarity model built on VGG-16 features at layer 31. *Top:* BiLRP explanations on different pairs of input images from the Pascal VOC 2007 dataset. Red and blue colors indicate positive and negative contributions to the similarity. *Bottom:* Effect of the BiLRP parameter γ on the explanation.

a small white calf in the right part of the first image. In example F, the scene is cluttered, does not let appear any meaningful similarity structure, and we observe that while a cat appears in both images these are not highlighted by the explanation. 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. 3.6 (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. In addition, we demonstrate the effect of different rendering parameters p as presented in Figure 3.7. Here, we select $p = 2$, which results in explanations of visual complexity that are visually not too sparse yet not overly complex.

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

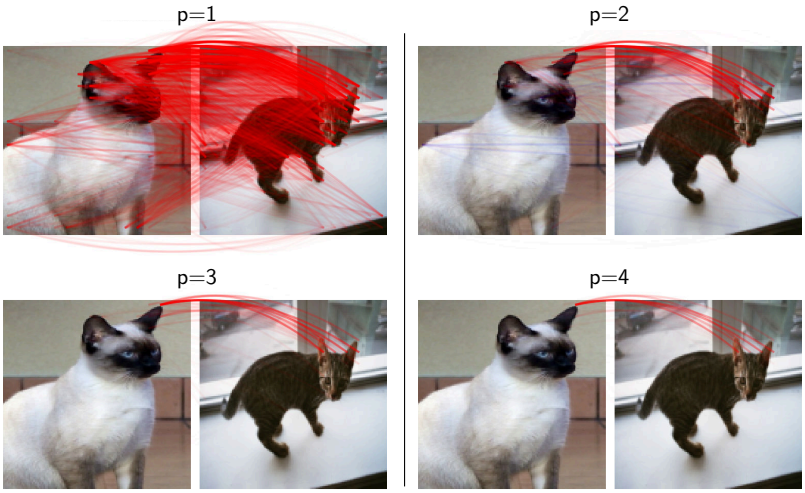


Figure 3.7: Effect of the parameter p on the rendering of the explanation. The higher the parameter p, the sparser the explanation.

3.6.1 How *Transferable* is the Similarity Model?

Through their layered structure, deep neural networks extract feature representations at different processing steps that provide a natural framework for multitask and transfer learning [193, 194]. DNN-based transfer learning has seen many successful applications [195, 196, 197]. In this section, we investigate 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 ability to transfer feature representations can be assessed from an inspection of BiLRP explanations.

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

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

Like in the previous experiment, we add to each feature representation a random projection onto 100 dimensions to make explanations faster to compute. In the following experiments, we use similarity in the context of different identification tasks. The two datasets 'Unconstrained Facial Images' (UFI) [198] and 'Labeled Faces in the Wild' (LFW) [199] are used in a face identification task, for which a good similarity model is needed to extract the closest matches in the training data reliably [181, 200]. Our third dataset focuses on historical illustrations from 'The Sphaera Corpus' [161, 201]. This material is composed of 358 scanned academic textbooks from the 15th to the 17th century and contains texts, illustrations and tables related to astronomical studies. The similarity between these entities is of

interest, as it can serve to consolidate historical networks and enable the automated analysis of historical material at scale [184, 202, 203]. We will focus on similarity between historic illustrations and return later to the Sphaera corpus in an extended use case study on computational tables in Section 3.8.

Input images of 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, and for each pair, we compute the BiLRP explanations.

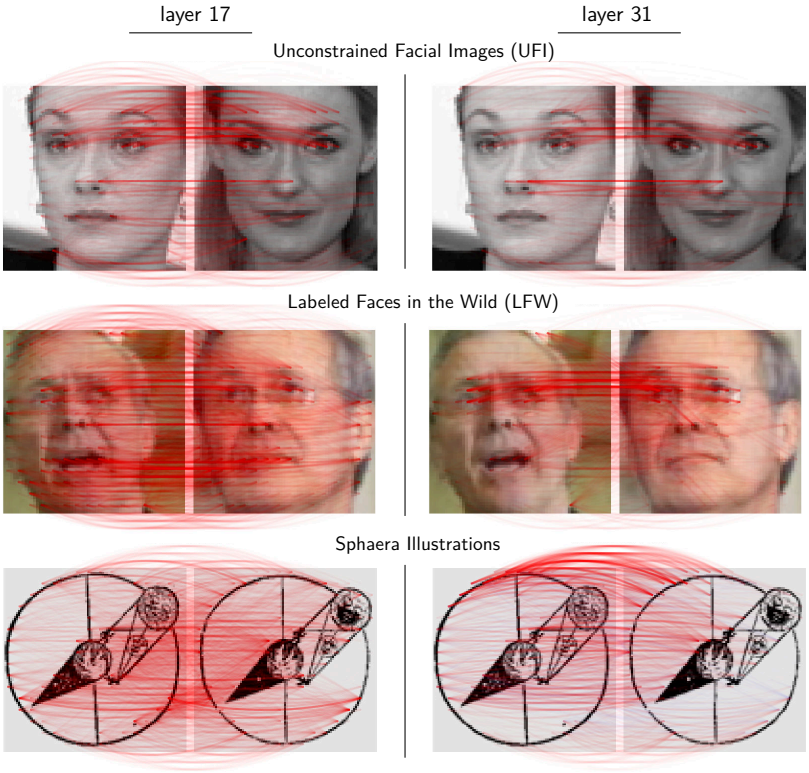


Figure 3.8: Application of BiLRP to study how similarity built on VGG-16 features transfers to various datasets. The resulting explanations of the similarity score are shown for different processing steps at layers 17 and 31 of the VGG-16 model.

Results We present results for the similarity model at layer 17 and 31 in Fig. 3.8 and 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 is built on features from the highest layers that are potentially capable of capturing very fine variations, e.g., for the eyes, might not have kept sufficiently many other features 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 3.1.

Table 3.1: Accuracy of an 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 support the hypothesis initially constructed from the BiLRP explanations. The Overspecialization of top layers on the original task leads to a sharp drop in accuracy on the target task. Best accuracies are instead obtained using features from the intermediate processing layers.

3.6.2 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 [204, 205, 206]. 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 [207, 208], 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 [208].

Setup We consider an invariance measure for similarity models based on the local/global similarity ratio:

$$\textsc{Inv} = \frac{\langle y(\boldsymbol{x}, \boldsymbol{x}') \rangle_{\text{local}}}{\langle y(\boldsymbol{x}, \boldsymbol{x}') \rangle_{\text{global}}}. \tag{3.6}$$

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 the 'UCF Sports Action' video dataset [209, 210], 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 a 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.

Table 3.2: Invariance measured by Eq. (3.6) 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**

Results We present invariance scores obtained for similarity models built at various layers in Table 3.2. 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. (3.6).

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

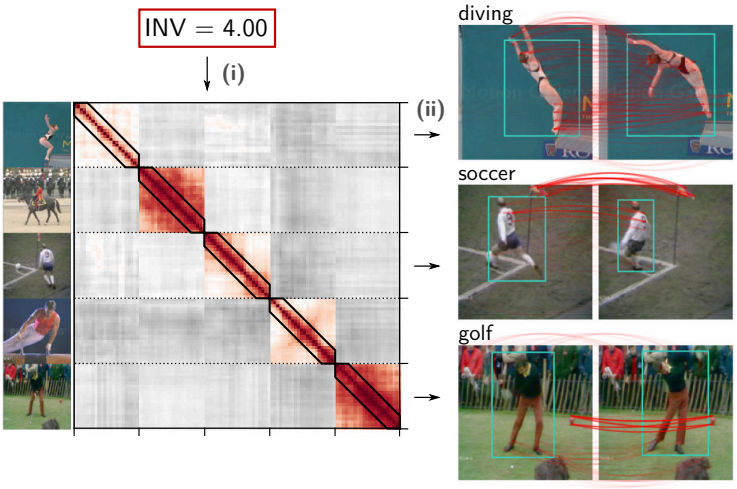


Figure 3.9: Explanation of measured invariance at layer 31. *Left*: The similarity matrix associated to a selection of video clips. The diagonal band outlined in black contains the pairs of examples in $\langle\cdot\rangle_{\text{local}}$. *Right*: BiLRP explanations for selected pairs from the diagonal band.

With this two-step analysis, we take a detailed look at the similarity structure that underlies the invariance score for a selection of videos and pairs of video frames, as shown in Fig. 3.9. The first example shows a diver rotating counterclockwise as she leaves the platform. We observe that the contribution to invariance is attributed to the different parts of the rotating body in expected ways. The second example shows a soccer player performing a corner kick. While part of the invariance is attributed to the player moving from right to left, a considerable amount of relevance is also attributed in an unexpected manner to the static corner flag behind the 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 seen as another example for 'Clever Hans' behavior [30]. The high nominal model performance on a given task can mislead the user to believe that the model works as intended. Similar 'Clever Hans' effects can be observed beyond video data, e.g., when applying the similarity model to historical illustrations in the Sphaera corpus. Figure 3.10 shows two pairs of illustrations whose content is equivalent up to a rotation and for which our model predicts a high similarity score. In both cases, BiLRP reveals that the high similarity is not a result of correctly matching the rotated patterns but instead capturing the interaction of fixed elements at the center and the border of the image.

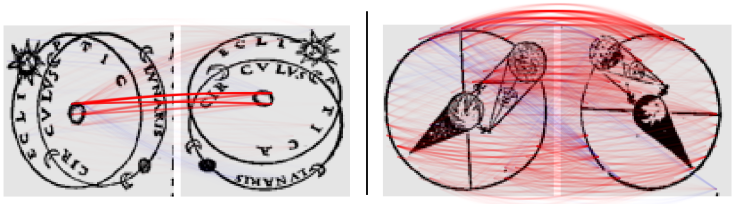


Figure 3.10: Pairs of illustrations from the Sphaera corpus explained with BiLRP. The high similarity originates mainly from matching fixed features in the image rather than capturing the rotating elements.

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.

3.7 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 3.7.1 how the interpretable feedback provided by BiLRP can be used to fix a flawed similarity model. Then, we turn to a use case in Section 3.8, where we build a domain-specific similarity model in the

digital humanities, which is both predictive and explainable with BiLRP, and that enables the discovery of historical insights.

3.7.1 Fixing a 'Clever Hans' Similarity Model

In the example of Fig. 3.10, BiLRP has revealed a Clever Hans effect of the similarity model. We observe that the model assigns 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.

A simple mitigation strategy to force rotation invariance into the model, is to compute the similarity score for all flips and 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 a 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. 3.11.

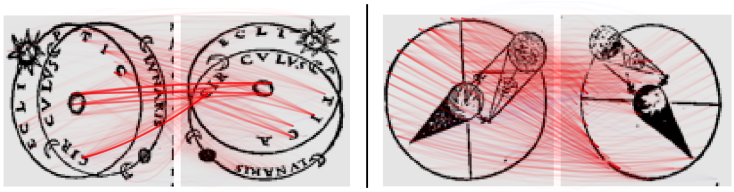


Figure 3.11: Pairs of illustrations from the Sphaera corpus and BiLRP explanation for the *improved* similarity model. Similarity captures rotating elements such as letters.

Compared to the original model (Fig. 3.10), 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 specific analysis through many pairs of data points and their corresponding explanations to systematically turn explanatory feedback into model improvements, as discussed in Section 2.6.3.

3.8 Use Case: Extracting Historical Insights in a Corpus-Level Analysis of Astronomical Tables

We now turn to another challenge in the digital humanities that addresses the automated processing of historical computational tables. These could so far only be very selectively addressed due to the meticulous analysis process and the limited number of experts needed to comprehend a single table. The high data heterogeneity precludes the use of similarity models that are built from standard pre-trained feature extractors. Instead, we use an *engineered* similarity model that can address the peculiarities of historical data.

We use this approach to assess similarity between numeric tables extracted from highly heterogeneous historical textbooks. We consider scanned numeric tables from the *Sphaera* Corpus [161]. The tables in the corpus typically report astronomical measurements or calculations of the positions of celestial objects in the sky. We take a closer look at two exemplary table pages later in Fig. 3.13 (left).

Traditionally, historical research questions are investigated by considering single case studies involving close reading and evaluating predefined sets of hypotheses on limited data. To trace the history of science at scale, we are interested in analyzing the evolution of scientific knowledge and its mechanisms, such as homogenization and divergence of knowledge. But, the possibilities to analyze how science could evolve in geographical and temporal dimensions are currently limited. The sheer number of historical sources that would have to be analyzed in detail exceeds human possibilities.[3] Instead, we develop a machine learning-assisted approach. For this historical analysis, we require a well-predicting model that is able to robustly handle the high data heterogeneity and for which we can also verify that meaningful and reliable data features support the obtained predictions.

3.8.1 Heterogeneity in the *Sphaera* Corpus

The challenges posed by heterogeneous data are a key limiting factor to automate processes and generally, heterogeneity can be characterized by a lack of uniform composition across samples in a dataset and makes up more than 90% of big data [211]. It is also a characteristic feature of historical corpora, which can be attributed to the printing process that has resulted in various irregularities and the more recent and non-standardized digitization process across libraries and research institutions. We illustrate some examples of the heterogeneity present in the Sphaera corpus using digit and non-digit patches in Figure 3.12. Other sources of heterogeneity include the very different states of preservation that have resulted in damaged, folded, wrinkled, stained or de-saturated pages. The print process has also introduced noise due to

[3]The *Spheara* corpus contains $\approx 10,000$ table pages for which an expert would need to carefully inspect each of the individual digits composing the table, and possibly their location in the table. A manual assessment of similarity would require a meticulous examination of each table content from which similarity scores can subsequently be computed, or $10,000 \times 10,000$ manual pairwise table comparisons for an optimal result.

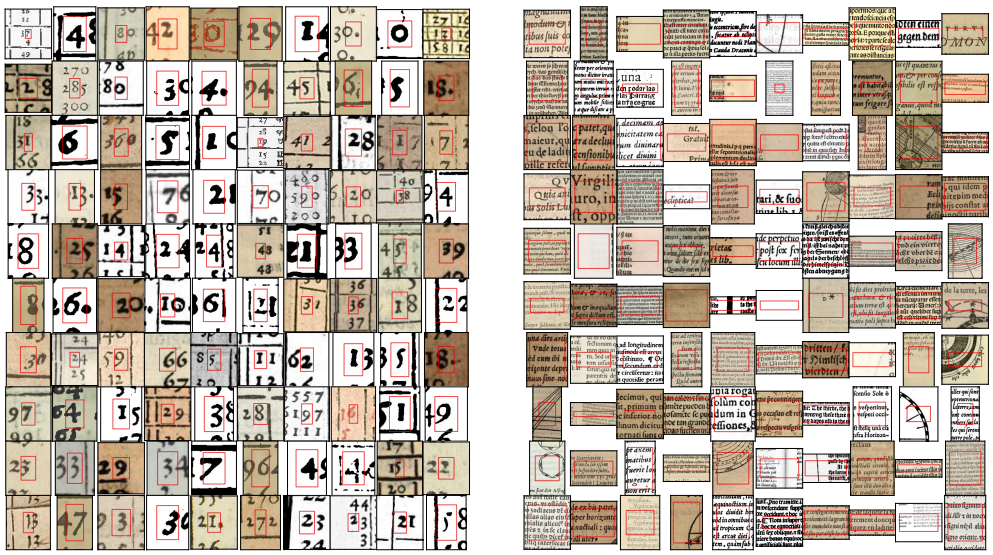


Figure 3.12: Examples of Sphaera corpus patches that illustrate the large variability in historical printing. All patches are extracted from the scanned material before any pre-processing is applied. *Left*: Digit patches with human-annotated bounding boxes. *Right*: Non-digit patches used as a contrastive learning signal. Patches are extracted via randomly sampling regions from non-table pages.

the tedious task of typesetting tables with many numbers, resulting in unwanted variations between identical tables. Many more sources of heterogeneity in the Sphaera corpus can be identified, which precludes using standard machine learning solutions. For example, end-to-end training pipelines assume that large amounts of training data from sufficiently variable sources are available to handle more complex data robustly. Instead, we use an 'atomization-recomposition' approach that is able to make inferences using very few annotated material at a lower data complexity.

3.8.2 Atomization-Recomposition Approach

In an initial atomization step, the complex superposition of many features is broken up into the smallest semantically meaningful unit, e.g., single digits. These representations act as the basic building blocks which offer the possibility to handle heterogeneity, robustness and invariance properties at an intermediate data complexity. After the atomization model is trained to detect these building blocks robustly, it can be used to recompose more complex and task-relevant features.

Atomization

To handle the high number of different printers and publishers who printed astronomic tables in the Sphaera corpus and the corresponding number of different font types

and sizes in combination with the heterogeneous quality of the digitized sources we use a digit recognition model. For this, we build a convolutional neural network and apply ReLU layers between the convolutional layers, except at the final layer. The encoder network outputs 10 activation maps $\{\boldsymbol{a}_j(\boldsymbol{x})\}_{j=1}^{10}$ for the digits 0–9. For each digit j the network is trained to output a Gaussian blob positioned at the digit location in map $\{\boldsymbol{a}_j(\boldsymbol{x})\}$. At the final layer we subtract a small bias term $b = 0.1$ before the ReLU layer in order to attenuate background activity. To model variations in scan orientation and size, we identify the page scaling factor s and rotation θ, for which the single-digit activation maps are maximally activated (sum of activations). The model is built from linear/ReLU and positive homogeneous layers to remain explainable with BiLRP while retaining the necessary representation power. Additional details on the architecture are given in Appendix A.2.1.

Recomposition

After handling the different sources of heterogeneity, we are now able to extract single-digit activation maps robustly. These will serve as the basis to recompose more complex and task-relevant features, which in our case are compositions of single-digits such as bigrams. We compute bigram maps by applying an element-wise 'min' operation

$$\boldsymbol{a}_{jk}^{(\tau)}(\boldsymbol{x}; s, \theta) = \min\left\{\boldsymbol{a}_j(\boldsymbol{x}; s, \theta), \tau(\boldsymbol{a}_k(\boldsymbol{x}; s, \theta))\right\},$$

which signals the presence of bigrams $jk \in 00$–99 and can be seen as a continuous 'AND' [166] operation. In addition, we build features that detect isolated single digits $j \in \cdot 0 \cdot$–$\cdot 9 \cdot$ with $\cdot$ indicating that no digit activity is present in the neighborhood.

The function τ represents a translation operation shifting activation maps by δx. We use multiple shifts as candidate alignments and identify digit compositions by applying a spatial max-pooling layer:

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

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. This results in a total of 110 feature maps, from which we extract a summarized vector representation by globally pooling the evidence for each feature. This implements translation invariance regarding the exact feature location. The final representation is computed either by sum-pooling over the bigram activation layers or an additional peak detection step that extracts a discrete feature count map from the activation maps (cf. Appendix A.2.1.2).

3.8.3 Verifying the 'Bigram Network' with BiLRP

We next verify our approach on the task of predicting table similarity using BiLRP. Examples of commonly used validation procedures include precision-recall curves

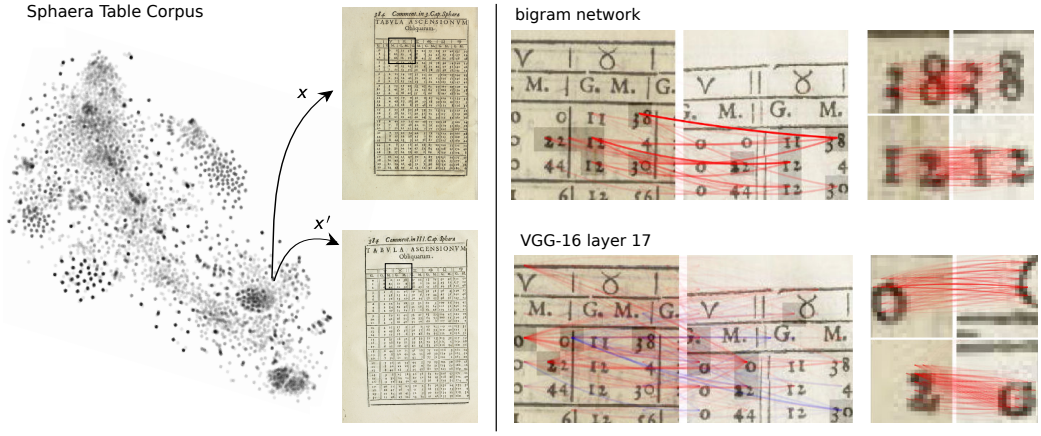


Figure 3.13: Explainable AI and machine learning in the digital humanities. *Left*: T-SNE visualization of the collection of tables from the Sphaera Corpus [161] from which we extract two tables with identical content. *Right*: Detailed BiLRP explanations of predicted similarities between the two input tables for our bigram approach and the pretrained object recognition model VGG-16.

or the ability to solve a proxy task (e.g., table classification) from the predicted similarities. These approaches require end-to-end label information, which is difficult to obtain for the type of data considered here. Furthermore, when the labeled data is not sufficiently representative, these procedures are potentially affected by the 'Clever Hans' effect.

In the following, we will use the explanatory feedback offered by BiLRP to verify that the model indeed uses the desired numerical features to predict similarity. 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 4.5. We show examples of our analysis in Fig. 3.13 (right).

The bigram network similarity model correctly matches pairs of digits in the two tables. Furthermore, these relevant interactions 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 task-irrelevant aspects, such as table borders. In addition, BiLRP offers a way to extract relevant feature

interactions at different granularity, e.g., on a more coarse level by pooling relevance over pixel locations that results in patch-level explanations or a very detailed explanation on the level of interacting individual pixels, as shown in Fig. 3.13 (right). Hence, for this particular table similarity task, BiLRP can establish the superiority of the bigram network over VGG-16.

This assessment could be obtained from a *single* pair of tables. If, instead, we had applied a validation technique that relies fully 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 assessing reasonable model behavior is especially important in the digital humanities or other scientific domains, where ground-truth labels are often scarce and expensive to obtain.

3.8.4 Generating Corpus-Level Historical Insights

After having extracted representations for the full *Sphaera* Table Corpus we are now in the position to analyze corpus-level trends at scale. In the following, we describe two examples that we have identified: temporal shifts and geographical singularities in printing.

Temporal Shifts in Printing

Many interesting phenomena in the history of science are linked to the evolution of knowledge over time as scientific insights are being transformed, diverge into new directions, and knowledge is extended or forgotten [212, 213, 214]. In the following, we aim to investigate such effects in the Sphaera Table corpus and perform an over-time analysis of the full corpus statistics.

Setup The books of the Sphaera corpus that contain at least one page of tables were printed during a period of 153 years (1494-1647), over which publication rates changed considerably. We use a sampling-based temporal analysis to deal with this imbalanced distribution of available table pages over time. For each time step t_i we assign a sampling probability to each book page from a truncated normal distribution $\mathcal{N}(t_i, \sigma^2)$ which sets probabilities for data points outside the interval $(t_i - \sigma, t_i + \sigma)$ to zero. At every step we sample $N = 80$ data points, determine their cluster membership label, construct the cluster count histogram of size $1 \times k$ and compute the entropy $H(p_{cl}) = -\sum_k p_{cl,k} \log(p_{cl,k})$ of the cluster probability vector $p_{cl} \in \mathbb{R}^{1\times k}$. By this we measure to what extent new material can be grouped into previous clusters, i.e., reproducing already established semantics versus populating new areas in the embedding space, i.e., adding novel scientific knowledge, by computing the entropy of the table distribution over clusters. We apply different digit density thresholds by selecting all tables from a time step that have at least $\{0, 100, .., 300\}$ digits.

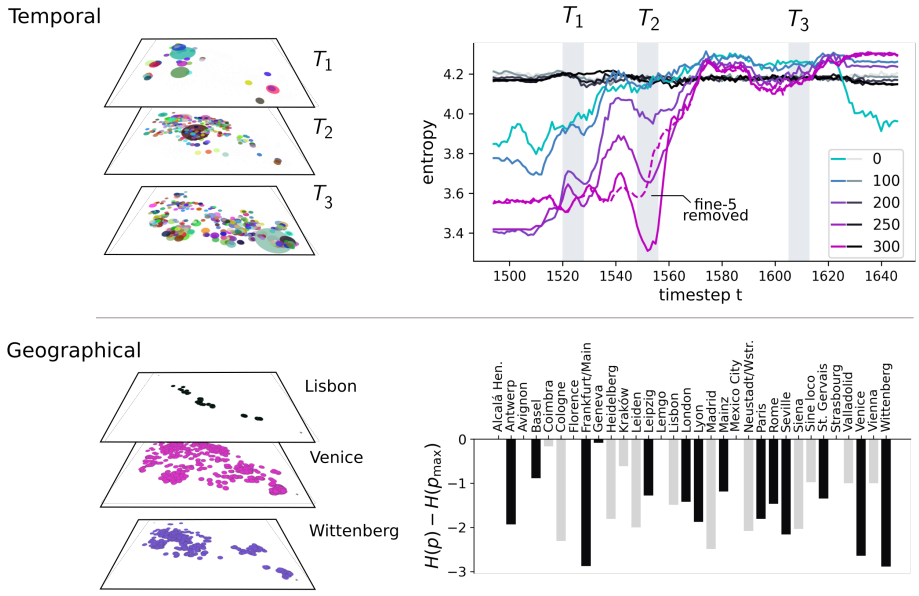


Figure 3.14: Corpus-level analysis of the Sphaera table pages. *Top*: Temporal evolution of knowledge displayed by computing the entropy of cluster membership vectors (number of tables in each cluster) for each time step. Gray to black lines correspond to a random embedding baseline, colored lines correspond to the data from our corpus. Different colors indicate a filtering threshold on the digit density per page, i.e., all pages containing at least 100 digits. The clusters are shown as t-SNE visualization for three time intervals indicating active clusters and cluster disk diameter is proportional to cluster size. We observe a marked drop in entropy for tables with extensive numerical content between 1540 and 1560. This drop disappears after removing the *fine-5* group, a subset of tables that occur in Finé books that we identified as the dominant factor driving the entropy change. *Bottom*: Geographical analysis of knowledge distribution for each print location in alphabetical order using relative entropy. Low-output cities (<=100 tables) are colored in light gray. For three selected cities, t-SNE visualization of the distribution of the printed tables is provided.

Results The resulting entropy evolution is shown in Figure 3.14 (top). We observe two particularly notable features that we describe in to following. First, around t=1550 we find that the entropy is minimized and modulated by the digit density. A subsequent historical investigation of the clustering distributions that were responsible for the strongest absolute change in entropy between $t = \{1540, ..., 1560\}$ reveals that this dip is likely the result of an exceptional episode, the printing of essentially the same work five times between the years 1551 and 1555 by the same printer. This work, Oronce Finé's Sphere, contains many particularly high-density tables. If the analysis is repeated with these five books artificially removed from the corpus, the dip disappears as shown by the dashed magenta line in Figure 3.14 (top), which suggests that it is indeed due to this particular episode in the printing history. This provides proof of concept that we are able to identify singularities in the history of

printing by our method of analysis, such as the unusually frequent reissue of a specific book in a short period. Overall, our findings suggest that focusing on high-density tables is necessary to reveal interesting temporal entropy changes at the full corpus level. This can be explained by the fact that low-density tables carry less specific mathematical information that does not vary greatly over time and instead often contain more basic tables such as enumerated lists or table of contents.

Second, we observe a trend of increasing entropy until roughly 1570, when saturation sets in. Lower entropy suggests that scientific knowledge encoded in numerical tables stayed closer to previously published material, while higher entropy signals the addition of semantically new tables. The books of the corpus, which all focus on the same core knowledge, are printed in more and more places and reached an ever-widening audience in this period. This effect can be described as knowledge homogenization. Novel contents are added to this existing core of knowledge in different ways, which signals processes of innovation during the first 100 years of the period considered.

These processes could be identified from the entropy curves for the investigated corpus of numerical tables. We next turn to a second analysis of effects related to specific print locations.

Geographical Singularities in Printing

To study the varying knowledge production expressed by the tables printed, we use the measure *relative entropy* for each of the 32 different book printing centers that are distributed all over Europe.

Setup We calculate the difference between the observed cluster entropy $H(p)$ to the maximum attainable entropy at this location $H(p_{\max})$. The latter quantifies for each city the entropy of a hypothetical, uninformed and uniformly distributed production process without memory of its print history and without outside influences. In this scenario, none of the printed tables is expected to be similar to any other. Thus, the relative entropy can be understood as a measure of the redundancy created by the actual process of content production and distribution in print as compared to this hypothetical process for each location.

Results Our analysis in Figure 3.14 (bottom) shows that relative entropy varies strongly between print locations and that the minimum is reached for the cities of Frankfurt am Main and Wittenberg. This result indicates that astronomic tables printed in the treatises produced in Wittenberg and Frankfurt are more homogeneous and, therefore that textbooks in general were more similar to each other than those produced in other print locations. In Frankfurt, we identified that low relative entropy can be attributed to the fact that a great part of its book production was constituted by many reprints of the same editions. However, in Wittenberg the case is different. It is known that the main Protestant Reformers, Martin Luther and Philipp Melanchthon, meticulously designed and supervised the curriculum of

the Wittenberg University [215] and that they worked in close contact with the different printers and publishers that had moved their businesses to Wittenberg after the Reformation [216]. Thus, we can conclude that the homogeneity of the works produced in Wittenberg was due to political control of scientific knowledge executed by the Reformers. This interpretation is consistent with and backed by the fact that printed material on astronomic studies developed in the same period, mostly in Wittenberg, remained constant for many decades and was highly influential all over Europe [217, 218].

To summarize, we have presented how explainable AI can aid in enabling novel scientific insights into domains for which sufficiently annotated material and end-to-end learning are not feasible. Our atomization-recomposition approach is built with transparency in mind that allows expert users, in our case historians, to verify the extracted representations. Our corpus-level analyses have resulted in two historically plausible singularities that describe knowledge evolution along temporal and geographical axes.

3.9 Summary and Discussion

In this chapter, we have developed explanations that reflect the specific structure present in similarity models. As demonstrated, it is important to get a detailed and human-interpretable explanation of the predicted similarity before using it to train a practical machine learning model. We have introduced a theoretically well-founded method to explain similarity in terms of pairs of input features. The proposed BiLRP method can be expressed as a composition of LRP computations and brings explanations to the novel scenario of explaining similarity. 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' 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.

Limitations BiLRP inherits several properties from LRP, including its theoretical connection to the deep Taylor decomposition and robustness. This lack of robustness is a typical limitation of methods relying on the model derivatives, especially when the function to be analyzed is a deep neural network [63, 8]. The use of more robust propagation rules, such as LRP-γ can alleviate some of these effects resulting in better explanations as presented in Section 3.5. LRP also inherits its dependence on the specific model implementation and in our description of BiLRP, we have made assumptions on the type of network layers, e.g., positively homogeneous functions. While this includes many standard layers, in particular Linear/ReLU, Max-Pooling and Min-Pooling layers, this sets constraints on the architecture choices for which our analysis, e.g., on conservation, holds.

Nevertheless, in practice, the BiLRP procedure developed here can serve as a general strategy that consists of an initial computation of relevance contributions from each branch at every output neuron and a subsequent construction of the full relevance tensors that contain the feature-wise interaction scores. However, some BiLRP properties such as conservation and the direct connection to the Hessian formulation may not hold for other relevance propagation schemes.

For high-dimensional output feature maps, the LRP computation from each feature neuron required by BiLRP, can become computationally expensive. In our experiments on natural images using VGG-16, we have applied a random projection layer to reduce complexity considerably. While both theoretical and experimental findings support the efficiency and effectiveness of random projections [219, 220, 221, 222], the dimensionality reduction introduces some noise in the explanation since different random projections may lead to different results. We observed in our experiments that we could attain the desired speed-up of computations by selecting a projection dimension of 100, which resulted in consistent and sufficiently detailed explanations. Promising projection approaches for future work include principle component analysis, clustering of local activations or sparse projections of the model output, which can serve to further reduce representation complexity in meaningful ways and alleviate computational costs.

Evaluating whether fine-grained similarity explanations accurately explain ground truth data, especially in real-world scenarios, is an open problem. We initially explored to test this on natural image datasets (e.g., MS COCO [223]) but observed that this was not feasible due to a number of reasons discussed next. Inspired by heatmap evaluations that compute overlap between explanation and segmentation maps or bounding boxes for the considered class label [65, 224], we experimented with quantifying the observed similarity flow as computed by BiLRP between bounding boxes. For this, we used BiLRP explanations between same-class image pairs and computed how much relevance one bounding box receives from the other and vice versa. Similar to previous findings [224], we observed that this worked well for pairs of closely cropped objects without complex image backgrounds but, in general, did not provide a robust way for ground truth evaluation and we did eventually not further pursue these experiments. As reported in several experimental findings [29, 225, 30], image classification DNNs often ground their predictions not on the object alone but also on the background and surrounding scene. This makes a segmentation-based evaluation in real-world settings already difficult for standard heatmap explanations and even more infeasible for the case of evaluating feature interactions. In addition, explanations typically do not match segmentation masks since they highlight only the relevant image features and object parts for the prediction instead of the full object [65].

Assessing a method's ability to explain the model prediction faithfully is commonly performed using a feature flipping analysis and observing the impact on the prediction. In settings that go beyond first-order explanations, this requires a procedure to add or remove the isolated interaction of features. In CNNs this is challenging since a masking of pixels will also affect other interactions due to the overlapping of

neighboring receptive fields. In the next chapter, we will further investigate this in the context of higher-order explanations for GNNs.

This currently impedes the evaluation of explanations that consider feature interactions on real-world data. Until specific ground-truth annotations of similarity become available, we consider synthetic datasets a well-suited approach to judging explanation quality.

Similarity computations play a crucial role in machine learning. However, a high and plausible similarity score may not necessarily be grounded in the expected features. Our developed BiLRP approach has allowed us to compute detailed explanations for similarity scores by identifying relevant interactions between pairs of features. It provides an example of a model structure that motivates explanations beyond first-order terms. It thus enables verification of similarity predictions widely used in downstream tasks, such as information retrieval and visualization, and supports improving machine learning models.

4

Higher-order Explanations in Graph Neural Networks

Graph neural networks (GNNs) are a widely used approach to represent and make predictions on data that is best represented via graphs. Since the layer-wise computations directly act on the input graph, GNNs preclude the use of explainable AI approaches designed for standard neural networks. In the following, we consider the graph explanation as a group of relevant features that result from a *higher-order* expansion of the graph prediction. These higher-order contributions can be computed via specific back-propagation schemes such as layer-wise relevance propagation (LRP) and present a novel approach to explaining GNNs. The resulting GNN-LRP explanation method is evaluated using specifically developed perturbation schemes that are designed to use this additional information. Our results demonstrate the benefit of going beyond attributions on input features alone and the comparable or better performance compared to existing graph explanation approaches. We further present how our method can be used to extract explanations for binary graph classification and study the evolution of feature assembly throughout processing in a deep neural network for image recognition.

This chapter is based on the following work and partly includes already published material from:

[226] 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. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

4.1 Introduction

A broad variety of important problems can be represented using a graph structure, ranging from parsing trees in language processing, to social interaction networks or molecular graph structures. In order to efficiently represent and infer predictions from such structured data, GNNs [227, 228, 229] have been proposed and provide a modeling framework for various tasks [230, 231, 232, 233, 234, 235]. To ensure the robust and safe use of machine learning in engineering, science and society, explainable AI develops methods that make model predictions and their inner workings interpretable [10, 37, 12]. In GNNs, the model computations are tightly entangled with the input graph and thus computing explanations solely at the level of the input may not be sufficient to represent the complexity of the model.

This chapter focuses on how this structure can be used to extract theoretically well-founded graph explanations that faithfully reflect the complexity of the internal model mechanism. More specifically, the layered GNN computations can be analyzed using a *higher-order* Taylor expansion, which results in an attribution of the prediction in terms of collections of edges. These can be seen as walks into the input graph, e.g., expressed via a sequence of nodes to be traversed, which offers a novel way to interpret the GNN prediction beyond node or edge attributions.

We find that the higher-order expansion can be represented using first-order terms, which can be computed using a modified backward propagation procedure as illustrated in Figure 4.1 (left). This enables the use of the well-studied LRP method [23] to compute walk-based explanations on graphs and results in our GNN-LRP explanation method.

To evaluate these new types of graph explanations to previous approaches and methods, we develop graph perturbation schemes inspired by pixel-flipping [65] to quantify the faithfulness of identified features by each explanation method with respect to the model prediction.

We demonstrate in the following how the GNN-LRP approach can be used to gain insights into image classifiers that we view as a GNN operating on pixel lattices. The resulting explanations produce detailed and reliable explanations of the internal model processes that offer a new and insightful view on the inner computations. This highlights the usefulness of our method by considering the full GNN procedure from input to output prediction.

4.2 Related Work

In the following, we give an overview of related work for our graph explanation method. In particular, we focus on work that considers explanations using higher-order information and then summarize explanation techniques specifically designed for GNNs.

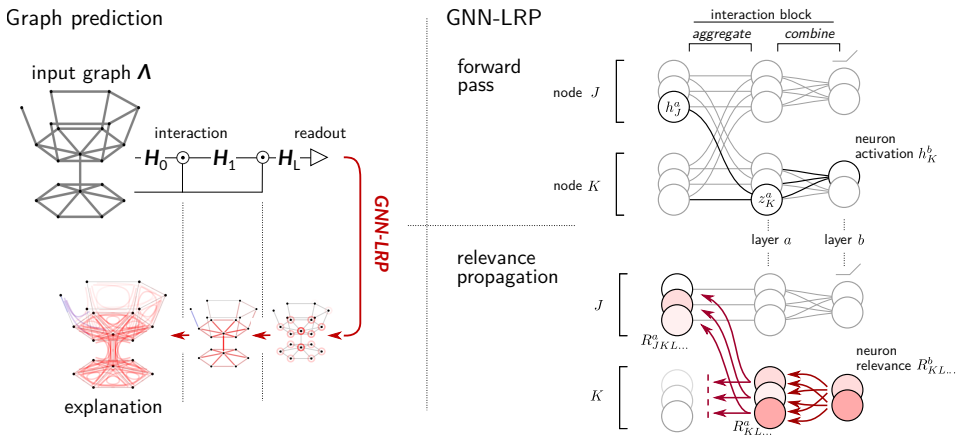


Figure 4.1: Overview of computing graph explanations using GNN-LRP. *Left*: After the forward prediction, the explanation procedure starts at the GNN output and layer-by-layer computes the graph walks relevant for the prediction. *Right*: A detailed GNN-LRP illustration shows an annotated forward pass through a single aggregate-combine interaction block. Nodes J and K in consecutive layers a and b highlight how relevance scores are obtained during the subsequent relevance propagation step. Relevance scores are only considered for existing walks in the input graph.

Higher-Order Explanations As demonstrated in the previous Chapter 3, second-order information can be used to attribute model predictions to pairs of input features, e.g., using the model's Hessian [139, 176, 236]. This presents a natural way to go beyond the sole identification of the most important features and in addition, offers a more detailed view on model computations by highlighting the relevant pairs of interacting features. We have seen that this is especially suited to bilinear computations such as dot product similarity. Second-order and higher-order explanations have also been incorporated by using an explicit sum-of-interactions structure [174]. Similarly, the extraction of higher-order feature interaction has been proposed via an iterative algorithm, which inspects neural network weights at the different layers [170].

Explaining Graph Neural Networks To make GNNs explainable, most methods have focused on attributions to nodes or edges of the input graph. Explanation techniques such as Grad-CAM or Excitation Backprop have been extended to the GNN model and use an attribution on nodes to explain the graph prediction [237]. Furthermore, identifying relevant subgraphs has been used as a general strategy to find an interpretable representation of the GNN prediction [238, 239, 240]. Similarly, the PGExplainer [241] and GNNExplainer [242] extract the relevant subgraph using an optimization procedure that maximizes the mutual information between a GNN prediction and subgraph candidates. In language processing, graph convolutional networks (GCNs) have been explained in terms of input nodes and edges using LRP explanations [243]. In a different approach, reinforcement learning has been

proposed to generate graphs that maximize a specific class prediction, which highlights characteristic graph patterns [244]. Other related approaches have focused on generating causal GNN explanations [245, 246]. Most similar, to our approach are the PGMExplainer method [247], which learns a probabilistic graphical model that measures the probability for different higher-order feature interactions in the model, and the GraphMask approach [248] that learns binary edge masks for each layer to test, which edges are most relevant for the prediction. Both methods are optimized to represent the prediction strategy of a GNN via an optimization criterion, which can result in unstable solutions and introduces additional complexity to the model.

In the following, we aim to extend explanations beyond attribution to the input graph's nodes, edges, or subgraphs. We develop a method that extracts higher-order feature attribution via a backward propagation procedure that computes explanations in terms *sequences* of features.

4.3 Explaining GNNs using Higher-Order Explanations

The following section introducs GNNs before we turn to the development of our graph explanations using a Taylor-expansion of the graph model. We will introduce two explanation techniques, namely GNN-GI and GNN-LRP that make use of higher-order attributions, and finally present how specific propagation rules can be developed and implemented in practice.

4.3.1 Graph Neural Networks

GNNs [227, 249] are specific types of neural networks that receive a graph $\mathcal{G}$ that is formalized by a set of nodes $\mathcal{N}$ and their edge connections $\mathcal{E}$ as an input. This graph structure can be summarized by the connectivity or adjacency matrix $\mathbf{\Lambda}$. In a typical GNN, this *input graph* is not only used in the fist layer but it appears also at later processing layers, which motivates to go beyond a standard explanation in terms of input features. This general formulation of GNNs via adjacencies and interaction blocks allows for a flexible adaptation of various graph structures, including directed, undirected, spatial, temporal, labeled, unlabeled, and many other graph structures [250, 251, 252].

A graph neural network can be constructed by a sequence of *interaction blocks* that are the different layers in our model. Every block $t = 1 \dots T$ computes a graph representation $\boldsymbol{H}_t \in \mathbb{R}^{n \times d_t}$, where n is the number of nodes in the input graph and d_t is the dimensionality of the node embedding. The representation in a given block is computed via an initial *aggregate* function that receives information from nodes in the neighborhood and a subsequent *combine* function that processes this information into aggregated node representations [253].

These two processing steps connect the representations $\boldsymbol{H}_{t-1}$ and $\boldsymbol{H}_t$ of consecutive blocks as:

$$\text{aggregate:} \quad \boldsymbol{Z}_t = \boldsymbol{\Lambda}\boldsymbol{H}_{t-1} \tag{4.1}$$

$$\text{combine:} \quad \boldsymbol{H}_t = \big(\mathcal{C}_t(\boldsymbol{Z}_{t,K})\big)_K, \tag{4.2}$$

with adjacency matrix $\boldsymbol{\Lambda}$ of size $n \times n$, $\boldsymbol{Z}_{t,K}$ the row of $\boldsymbol{Z}_t$ associated to node K, and $\mathcal{C}_t$ representing the 'combine' function.

Starting from some initial state $\boldsymbol{H}_0 \in \mathbb{R}^{n \times d_0}$, these two steps are repeated for each processing layer, and a final readout function g to compute the desired GNN prediction is applied:

$$f(\boldsymbol{\Lambda}; \boldsymbol{H}_0) = g(\boldsymbol{H}_T(\boldsymbol{\Lambda}, \boldsymbol{H}_{T-1}(\boldsymbol{\Lambda}, \dots \boldsymbol{H}_1(\boldsymbol{\Lambda}, \boldsymbol{H}_0)))). \tag{4.3}$$

4.3.2 Higher-order Explanations in GNNs

In a standard approach, we would derive an explanation of the graph prediction using a Taylor expansion of $f(\boldsymbol{H}_0)$ that only considers the dependency of the GNN on the initial state $\boldsymbol{H}_0$. As mentioned before, the input graph occurs repeatedly as an input in later processing steps, and thus this approach solely focuses on the relevance of a node in the input layer. Thereby, it ignores its effect on later processing layers and does not capture more complex relations between nodes.

Instead of computing the decomposition with respect to the initial state $\boldsymbol{H}_0$ we will next consider the 'true' input $\boldsymbol{\Lambda}$. As we have seen, it appears as a multiplicative term in the aggregate step and we will identify these interactions using a *higher-order* Taylor expansion of $f(\boldsymbol{\Lambda})$.

Assuming that $f(\boldsymbol{\Lambda})$ is smooth on the relevant input domain, we can compute a T-order Taylor expansion at some reference point $\widetilde{\boldsymbol{\Lambda}}$:

$$f(\boldsymbol{\Lambda}) = \sum_{\mathcal{B}} \underbrace{\frac{1}{\alpha_{\mathcal{B}}!} \frac{\partial^T f}{\partial \lambda_{\mathcal{E}_1} \dots \partial \lambda_{\mathcal{E}_T}}\bigg|_{\widetilde{\boldsymbol{\Lambda}}} \cdot \Delta_{\mathcal{E}_1} \cdot \ldots \cdot \Delta_{\mathcal{E}_T}}_{R_{\mathcal{B}}} + \dots \tag{4.4}$$

with $\Delta_{\mathcal{E}} := (\lambda_{\mathcal{E}} - \widetilde{\lambda}_{\mathcal{E}})$ and where we define $\alpha_{\mathcal{B}}! := \prod_{\mathcal{E}} \alpha_{\mathcal{B},\mathcal{E}}!$ with $\alpha_{\mathcal{B},\mathcal{E}}$ being the number of edge occurrences $\mathcal{E}$ in the bag-of-edges $\mathcal{B}$. The sum goes over all bags $\mathcal{B}$ of T edges. Thus, these terms in the sum represent the joint effect of multiple edges on the GNN prediction that we are interested in. With '$+\dots$' we denote the non-expanded terms of lower or higher edge than T.

Equation 4.4 presents a well-founded and general formulation of how the GNN structure interacts to produce the prediction. We observe that this formulation in terms of bag-of-edges requires the computation of higher-order derivatives, which calls for specific computation software and does not scale favorably to more complex

architectures. Hence, we introduce the concept of a walk-based explanations with *walk* $\mathcal{W} = (\dots, J, K, L, \dots)$ that represents an *ordered* sequence of nodes J, K, L between consecutive blocks. In the following, we focus on this representation using interactions between nodes to describe the layer-wise propagation process.

Furthermore, it can be shown that the non-expanded terms vanish for piecewise multilinear positively homogeneous functions such as GNNs with ReLU nonlinearity and without biases: choosing the reference point $\widetilde{\mathbf{\Lambda}} = s\mathbf{\Lambda}$ in the limit of $s \to 0$ leads to the conservation property $\sum_{\mathcal{B}} R_{\mathcal{B}} = f(\mathbf{\Lambda})$, and that the bag-of-edges relevance $R_{\mathcal{B}}$ can be represented as a sum over walks $R_{\mathcal{B}} = \sum_{\mathcal{W} \in \mathcal{B}} R_{\mathcal{W}}$ [226]. The latter walk-based representation produces detailed information on how the different edges interact along the block layers, and allows for a straight-forward implementation following the sequence of computation layers. Using this node-based formulation, we can now compute the relevance of a walk directly by:

$$R_{\mathcal{W}} = \frac{\partial}{\partial \dots} \left(\frac{\partial}{\partial \lambda^\star_{JK}} \left(\frac{\partial \dots}{\partial \lambda^\star_{KL}} \cdot (\lambda^\star_{KL} - \widetilde{\lambda}^\star_{KL}) \right) (\lambda^\star_{JK} - \widetilde{\lambda}^\star_{JK}) \right) \cdot \dots \tag{4.5}$$

with $\lambda^\star_{JK}$ being the connection between nodes J and K in consecutive blocks, '$\dots$' denotes placeholders for the leading and trailing walk nodes, and root points chosen such that walks in $\mathbf{\Lambda}$ coincide with walks in $\mathbf{\Lambda} - \widetilde{\mathbf{\Lambda}}$ (cf. [226] for a full derivation). This nesting of the individual blocks enables us to analyze each block iteratively from the last network layer back to the input features, and to propose the computation as a series of nested 'Gradient $\times$ Input (GI)' computations $R_{JKL\dots} = [\nabla R_{KL\dots}(\widetilde{\mathbf{\Lambda}}^\star)]_J \cdot \lambda_{JK}$. This graph explanation technique results in our 'GNN-GI' baseline that we will use in our experiments. We will also observe, that this baseline presents a special case of our more general 'GNN-LRP' method.

Furthermore, the analysis of each interaction block can itself be challenging. For example, the interaction blocks used in the graph isomorphism network (GIN) model [254], can be composed of multiple layers that make it difficult if not impossible to choose an accurate root point $\widetilde{\mathbf{\Lambda}}^\star$ for the Taylor expansion.

We now consider an extension of deep Taylor decomposition (DTD) [25] which consists of replacing the Taylor expansion of the layered model by several Taylor expansions performed at each layer (cf. Section 2.3). For standard neural networks, DTD leads to the robustified LRP method [23, 64] in comparison to naive gradient-based approaches. In the following we will use this approach and apply LRP to the individual GNN interaction blocks. This results in the 'GNN-LRP' graph explanation technique.

From DTD to GNN-LRP

In the following, we will focus on the graph convolutional network (GCN) [250]. In a GCN, each interaction block is composed of a linear aggregate function with positive

adjacencies, followed by a linear/ReLU combine function:

$$\begin{aligned} \text{aggregate:} \quad & z_K^a = \textstyle\sum_J \lambda_{JK} h_J^a \\ \text{combine:} \quad & h_K^b = \max\left(0, \textstyle\sum_a z_K^a w_{ab}\right) \end{aligned}$$

Here, h_J^a denotes the activation of some neuron with index a inside the node J. The notation $\sum_a$ represents a sum over all neurons a composing a node plus a hardcoded neuron '0' with activation $z_K^0 = 1$ and with bias w_{0b}. We further simplify the notation by omitting the star symbol for variable λ_{JK}. An annotated diagram of the GCN prediction and relevance procedure is given in Figure 4.1 (right).

In the following, we first redistribute relevance layer-wise to the intermediate representation $\boldsymbol{z}_K$ before then distributing relevance to the adjacencies $\boldsymbol{\lambda}_K$. The additional granularity introduced by considering neurons instead of only graph nodes results in additional relevance attribution to these neurons (e.g., relevance $R_{KL\ldots}^b$).

As part of DTD, we define the following relevance model $\widehat{R}_{KL\ldots}^b(\boldsymbol{z}_K) = h_K^b(\boldsymbol{z}_K)\, c_{KL\ldots}^b$. It is a product of the neuron activation (which is a function of the intermediate representation), and a term that is constant and set in a way that the relevance model matches the true relevance $R_{KL\ldots}^b$ locally. We can now follow the typical LRP-procedure and attribute the relevance score to neurons of the intermediate representation using a first-order Taylor expansion at some specific root point $\widetilde{\boldsymbol{z}}_K$ for each output neuron b:

$$R_{KL\ldots}^{a\leftarrow b} = \left.\frac{\partial \widehat{R}_{KL\ldots}^b}{\partial z_K^a}\right|_{\widetilde{\boldsymbol{z}}_K} \cdot (z_K^a - \widetilde{z}_K^a). \tag{4.6}$$

We then sum over all relevance contributions from neurons in the layer above, i.e. $R_{KL\ldots}^a = \sum_b R_{KL\ldots}^{a\leftarrow b}$. We next attribute these relevance scores to the adjacencies λ_{JK} in the aggregate step. Again, we define a relevance model $\widehat{R}_{KL\ldots}^a = z_K^a(\boldsymbol{\lambda}_K)\, c_{KL\ldots}^a$ and compute the first-order terms of a Taylor expansion:

$$R_{JKL\ldots}^a = \left.\frac{\partial \widehat{R}_{KL\ldots}^a}{\partial \lambda_{JK}}\right|_{\widetilde{\boldsymbol{\lambda}}_K} \cdot (\lambda_{JK} - \widetilde{\lambda}_{JK}). \tag{4.7}$$

Now, for the first Taylor expansion we set the root points $\widetilde{\boldsymbol{z}}_K$ in Eqs. (4.6) and (4.7) to the intersection of the line $\{\boldsymbol{z}_K - s\boldsymbol{z}_K \odot (\mathbf{1} + \gamma \mathbf{1}_{\boldsymbol{w}_b \succeq \mathbf{0}}) \mid s \in \mathbb{R}\}$ and the ReLU hinge, with γ being a factor that favors root points towards more positive neuron activations $\boldsymbol{z}_K$.

Inserting this root point in Eq. (4.6) results in the relevance messages $R_{KL\ldots}^{a\leftarrow b} = z_K^a\, w_{ab}\, s\, (1 + \gamma 1_{w_{ab} \geq 0})\, c_{KL\ldots}^b$ with w_{ab} denoting the weight connects neuron a and neuron b. By resolving the parameter s and pooling relevance messages coming from the multiple output neurons, we obtain the propagation rule:

$$R_{KL\ldots}^a = \sum_b \frac{z_K^a (w_{ab} + \gamma w_{ab}^+)}{\sum_a z_K^a (w_{ab} + \gamma w_{ab}^+)} R_{KL\ldots}^b. \tag{4.8}$$

Similarly, for propagation in the aggregate layer, we find root point $\widetilde{\boldsymbol{\lambda}}_k = \mathbf{0}$ and injecting this root point in Eq. (4.7) results in the following propagation rule:

$$R^a_{JKL\ldots} = \frac{\lambda_{JK} h^a_J}{\sum_J \lambda_{JK} h^a_J} R^a_{KL\ldots}. \tag{4.9}$$

Finally, we arrive at the GNN-LRP relevance propagation scheme to compute walk-based explanation:

$$R^a_{JKL\ldots} = \sum_b \frac{\lambda_{JK} h^a_J w^{\uparrow}_{ab}}{\sum_{J,a} \lambda_{JK} h^a_J w^{\uparrow}_{ab}} R^b_{KL\ldots}, \tag{4.10}$$

with $w^{\uparrow} = w + \gamma w^{+}$. The propagation rule can be seen as a generalization of the LRP-γ rule [64] to the GCN. In parallel to the connection between LRP and Gradient $\times$ Input, for $\gamma \to 0$ explanations produced by GNN-LRP become equivalent to those of the proposed GNN-GI baseline.

This framework to derive GNN-LRP rules can be applied to any graph network resulting in GNN-LRP procedures for various GNN architectures as we summarized in [226]. It requires that the GNN model can be formulated as successive computation of aggregate and combine steps as given in Equations (4.1) and (4.2).

4.3.3 Implementation of GNN-LRP

In order to implement GNN-LRP, we can apply a set of implementation tricks using a set of forward/backwards hooks. These modify the LRP gradient computations along isolated walks through the graph. For example, GNN-LRP for a GCN can be easily implemented by rewriting the combine function of each interaction block as:

$$\begin{aligned}
\boldsymbol{P}_t &\leftarrow \boldsymbol{Z}_t \boldsymbol{W}_t^{\uparrow} \\
\boldsymbol{Q}_t &\leftarrow \boldsymbol{P}_t \odot [\rho(\boldsymbol{Z}_t \boldsymbol{W}_t) \oslash \boldsymbol{P}_t]_{\text{cst.}} \\
\boldsymbol{H}_t &\leftarrow \boldsymbol{Q}_t \odot \boldsymbol{M}_K + [\boldsymbol{Q}_t]_{\text{cst.}} \odot (\mathbf{1} - \boldsymbol{M}_K),
\end{aligned}$$

where $[\cdot]_{\text{cst.}}$ detaches the quantity to which it applies from the gradient, $\boldsymbol{M}_K$ represents a mask that selects node K, and $\odot$ and $\oslash$ refer to the element-wise multiplication and division respectively. The variables $\boldsymbol{P}_t$ and $\boldsymbol{Q}_t$ denote intermediate hidden representations. The automatic differentiation capabilities of machine learning software in which standard layers such as convolution or pooling are already predefined allow the implementation of GNN-LRP even for more complex GNN architectures with minimal changes necessary to the code.

4.4 Evaluation of GNN-LRP

In the following, we test the GNN-LRP method on various types of GNNs and several graph prediction tasks. We begin with a qualitative inspection of explanations comparing different explanation baselines and then turn towards the quantitative evaluation of these approaches.[1]

4.4.1 Qualitative Evaluation

In order to validate the extracted explanations, we investigate a user-controllable two-class synthetic problem that we call 'BA-growth'.

Setup For this, we modulate the growth parameter in Barabási-Albert graphs, [255] and assign 'class 1' for growth parameter 1 or 'class 2' for higher growth parameters that result in a growth behavior in which new nodes are attached preferably to low-degree nodes. We consider a graph isomorphism network (GIN), which have been found to be a powerful GNN architecture and that differ from standard GCNs by using a multilayer perceptron for the 'combine' step [254]. In the following, we use two interaction blocks with each consisting of a two-layer network with 32 neurons per node at each layer to aggregated node features into node embeddings. The initial state $\boldsymbol{H}_0$ is a matrix of size $n \times 1$ filled with ones, which indicates that nodes do not have intrinsic information. The GIN receives as input the connectivity matrix $\boldsymbol{\Lambda} = \widetilde{\boldsymbol{A}}/2$ where $\widetilde{\boldsymbol{A}}$ is the adjacency matrix augmented with self-connections. The GIN is trained on this task until convergence, where it reaches an accuracy above 95 %. More details on the model and its training are given in Appendix A.3.1. After training, we take an input graph from class 1 and explain the GIN prediction using GNN-LRP. We use the LRP parameter $\gamma = 2$ and $\gamma = 1$ in each layer of the first and second interaction blocks, respectively.

We compare GNN-LRP to a selection of other GNN explanation methods:

- *Pope et al.* [237]: The method views the GNN as a function of the initial state $\boldsymbol{H}_0$ and performs an attribution of the GNN output on nodes in $\boldsymbol{H}_0$. This framework lets the user choose the technique to perform attribution on $\boldsymbol{H}_0$. In our benchmark, we use the techniques Gradient × Input (GI) and LRP.
- *GNN-GI*: This simple baseline replaces the LRP steps in the GNN-LRP procedure by Gradient × Input steps and serves as a baseline for higher-order graph explanations. It can also be seen as a special case of GNN-LRP with parameter $\gamma = 0$.
- *GNNExplainer* [242]: The method runs an optimization procedure that identifies a selection of edges that maximize the model output prediction. This procedure thus identifies a mask $\boldsymbol{M} = \sigma(\boldsymbol{R})$ with σ denoting the logistic sigmoid function

[1] Our code implementation of GNN-LRP and the experimental analyses are available at: `https://git.tu-berlin.de/thomas_schnake/paper_gnn_lrp`

that maximizes the prediction $f(\boldsymbol{M} \odot \boldsymbol{\Lambda})$. Finally, the explanation is directly given by $\boldsymbol{R}$.

Results Explanations produced by each method are shown in Fig. 4.2. The method by Pope et al. [237] highlights relevant nodes for the prediction. It is difficult to determine from the explanation whether the highlighted nodes are relevant by themselves or if they are relevant in relation to their neighbors. The GNN-GI baseline and the GNN-LRP method provide a higher level of granularity, which allows distinguishing between the contribution of an individual node and its interactions with other nodes. Compared to GNN-LRP, the GNN-GI baseline tends to be less selective, with spurious positive or negative relevance, and is generally more noisy (we will return to this effect later in Section 4.5). While the GNNExplainer [242] highlights similar relevance patterns as GNN-LRP, its explanations are less detailed. This first qualitative analysis suggests that GNN-LRP is the only method in our benchmark that computes explanations with the desired robustness and a high level of detail.

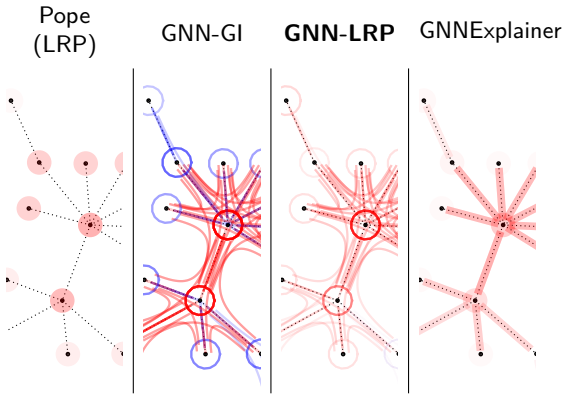


Figure 4.2: Comparison of different explanation techniques on one example from the BA-growth dataset. GNN-LRP and GNN-GI produce more detailed explanations than the other methods, and GNN-LRP explanations appear more robust than GNN-GI.

4.4.2 Quantitative Evaluation

Standard methods to evaluate graph explanations use attributions at the input level without considering higher-order information or including the structure of the graph specifically. To make use of this specific structure, we will consider approaches to evaluating graph explanations that focus on identifying relevant subgraphs or groups thereof.

Subgraph Selection Conceptually, in order to judge the performance of the different explanations, we are interested in identifying a sequence of subgraphs that is created by incrementally adding or removing nodes. Generally, finding an optimal

feature ordering is computationally intractable for graphs with many numbers of features. Instead, we consider a *local* approximation that selects the next feature that most strongly affects the model prediction. This approach offers the flexibility to be applied to both node-based as well as edge-based explanations and can also be applied to make use of higher-order information present in walk-based explanations.

Evaluation metrics In the following, we use this approach in two evaluation task settings: First, we consider the task of 'model activation'. We start with an empty subgraph $\mathcal{S}$ and evolve the sequence of subgraphs by adding at each step the node to the subgraph that maximally affects the relevance score $R_{\mathcal{S}}$. After adding each node to the graph, we observe the output of the GNN for the true class $f(\mathcal{S})$. We measure the area under this activation curve (AUAC) which is higher the more faithful the explanation method is. A faithful explanation method thus can correctly identify a subgraph that produces a high GNN output for the true class.

Instead of iteratively growing the subgraph, in the 'model pruning' task, we build a sequence by removing nodes from the graph. Here we are interested in quantifying what explanation produces a sequence of subgraphs that minimally affects the model output. Again, using the local approximation of the optimal sequence of subgraphs, we remove the next least relevant node at each step. We record the model output when pruning $\mathcal{G}$ according to the feature ordering, use the resulting subgraph as input to the model and observe the difference in model output between the original and pruned graph $|f(\mathcal{S}) - f(\mathcal{G})|$. The resulting area under the pruning curve (AUPC) is used to judge the different explanations methods. A low AUPC indicates an effective ordering since a small absolute relevance score should not strongly affect the model prediction.

Setup We next compute AUAC and AUPC scores to asses the explanation method's ability to explain the model. For the synthetic BA-growth dataset, we use GCN, GIN, and spectral network models with each consisting of two interaction blocks and neuron embeddings of 128, 32 and 32 at each layer. In addition, we consider two real-world datasets. We use the Stanford Sentiment Treebank (SST) [256] dataset for sentiment classification in movie reviews. The syntactic sentence information is encoded via a graph with nodes corresponding to word tokens. Finally, we interpret the VGG-16 pre-trained image recognition network [160] as a graph neural network operating on a lattice of size 14×14 that starts at convolutional block 3. Each node here represents the collection of activations at a specific spatial location. Additional details for each network and data setting are given in Appendix A.3.

Results Results for the activation task are presented in Table 4.1 (left). We find that on the *BA-growth* dataset GNN-LRP outperforms all other explanations methods on average. The nearest competitors are Pope et al. [237] (together with LRP), and the GNNExplainer [242]. This result further supports our qualitative analysis at the beginning of Section 4.4. For the pruning task, we show results in

Table 4.1: Evaluation of GNN explanation methods across datasets and models using activation and pruning tasks. Best performing methods are shown in bold. *Left*: Activation task scores as measured using the area under the activation curve (AUAC). Higher AUAC is better. *Right*: Pruning task scores as measured using the area under the pruning curve (AUPC). Lower AUPC is better.

	Activation					
	P [237] (GI)	P [237] (LRP)	GNN-GI (ours)	GNN-LRP (ours)	GNNExpl [242]	Random
BA-growth, GCN	2.54	3.02	2.93	**3.52**	3.32	1.05
BA-growth, GIN	2.75	3.49	3.04	**3.84**	3.71	1.18
BA-growth, Spec	0.29	**1.91**	0.23	1.85	1.65	0.09
SST, GCN	26.07	26.35	26.40	26.65	**27.07**	20.47
VGG-16 [160]	9.46	13.18	12.03	**14.04**	—	7.90

	Pruning					
	P [237] (GI)	P [237] (LRP)	GNN-GI (ours)	GNN-LRP (ours)	GNNExpl [242]	Random
BA-growth, GCN	3.61	3.08	2.84	**2.54**	2.80	4.55
BA-growth, GIN	3.93	3.12	3.24	**2.78**	3.08	5.06
BA-growth, Spec	3.37	2.35	2.93	2.63	**2.24**	3.89
SST, GCN	31.37	31.00	30.08	**29.11**	29.42	32.89
VGG-16 [160]	10.02	6.67	7.78	**6.05**	—	11.47

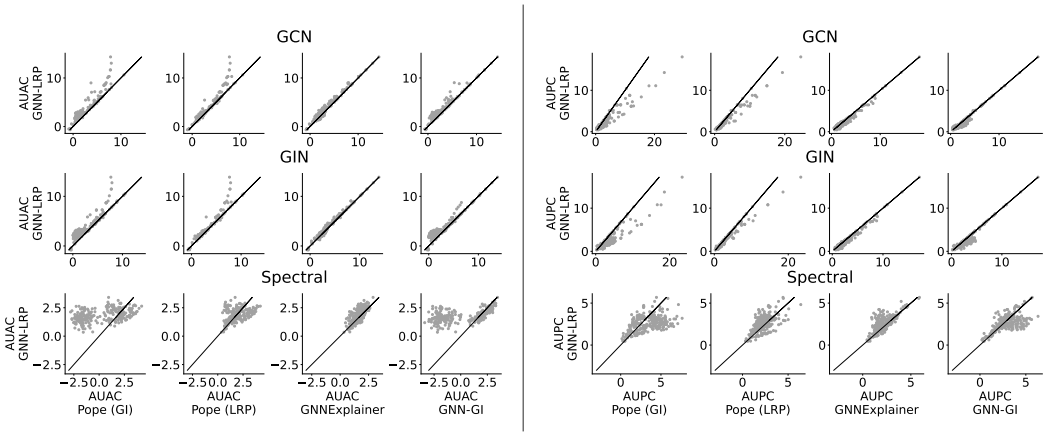


Figure 4.3: GNN-LRP comparison against other explanation methods. We compare the distribution of individual AUAC and AUPC scores from the BA-growth dataset on GCN, GIN and spectral network models. The x-axis shows the score for the comparison method and the y-axis for GNN-LRP. *Left*: AUAC comparison for the activation task. Points above the diagonal indicate better performance of GNN-LRP. *Right*: AUPC comparison for the pruning task. Points below the diagonal indicate better performance of GNN-LRP.

Table 4.1 (right) and find that GNN-LRP outperforms most of the other baseline methods in each of the experimental setups. It is overall the most effective method at affecting the model output minimally for the least relevant features.

We take a closer look at the distribution of AUAC and AUPC scores and observe that GNN-LRP provides systematically better explanations, as shown by the majority of points above the diagonal for the activation task in Fig. 4.3 (left). For the pruning task, we observe a similar pattern in Fig. 4.3 (right), with most GNN-LRP points achieving lower scores below the diagonal compared to other methods. For the spectral network, the distribution is generally less clear, and we observe that methods

tend to identify different explanations, which can result in better performance on some samples but fail for others, as for example for the Pope (GI) model. Overall, GNN-LRP performs comparable or better than the other explanation approaches.

In addition, we test the sensitivity of GNN-LRP towards its hyperparameter γ as shown in Figure 4.4. For the activation task, we observe that any choice of parameter $\gamma \geq 1$ delivers comparably high AUAC performance. When setting $\gamma = 0$, which results in the GNN-GI method, we observe the lowest performance scores. Regarding the choice of γ in the pruning task, we observe that larger values of γ achieve superior performance and that any choice of parameter $\gamma > 2$ results in low and very similar AUPC scores. These findings again support the increased robustness of the modified gradient computation when using non-zero values of γ.

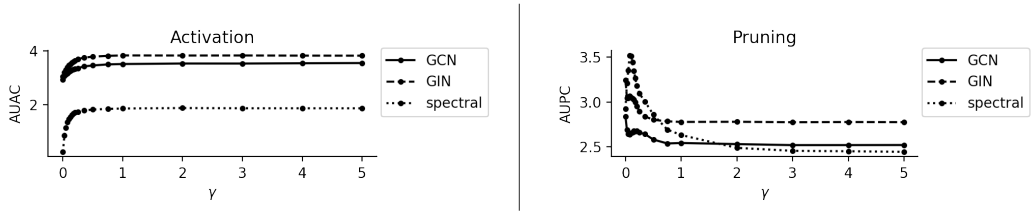


Figure 4.4: Effect of the parameter γ on GNN-LRP. Results are shown for GCN, GIN and spectral network models trained on the BA-growth dataset. *Left*: AUAC scores for the activation task. *Right*: AUPC scores for the pruning task.

Metric comparison Similar to our AUAC and AUPC metric, *fidelity* and *sparsity* have been proposed to evaluate GNN explanations [237, 257]. We compare the different metrics using the GCN model trained on the BA-growth dataset. In Figure 4.5, we show the averaged curves of the activation and pruning task, along the results of the fidelity metric when removing the most positive (fidelity$^+$) and most negative (fidelity$^-$) relevant features. We observe that the performance of the interpretation methods aligns with the other evaluation metrics and that GNN-LRP achieves a performance that is comparable or better than that of other models. This indicates that achieving a good performance in any of these metrics requires an explanation to be sparse *and* faithful. Additionally, activation and pruning tasks also reflect if the explanation method differentiates properly between important and redundant graph features, which is similar to what the *sparsity* measure does. Thus, the activation and pruning tasks reflect the trend captured by a variety of existing evaluation methods and present a robust and informative summary to assess the quality of explanations.

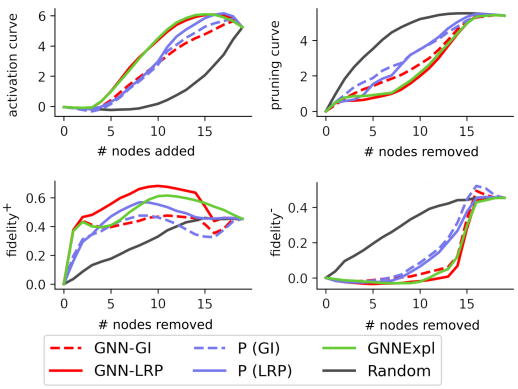


Figure 4.5: Comparison of evaluation metrics. Evaluation of explanation methods on the GCN model trained on the BA-growth dataset, when adding the most relevant nodes (activation) or removing the least relevant ones (pruning). In the top row, we compute the activation/pruning curve, and the bottom row reports the fidelity$^+$ and fidelity$^-$ scores when iteratively masking out the most/least relevant features.

4.5 Use Case: Revisiting Image Classification

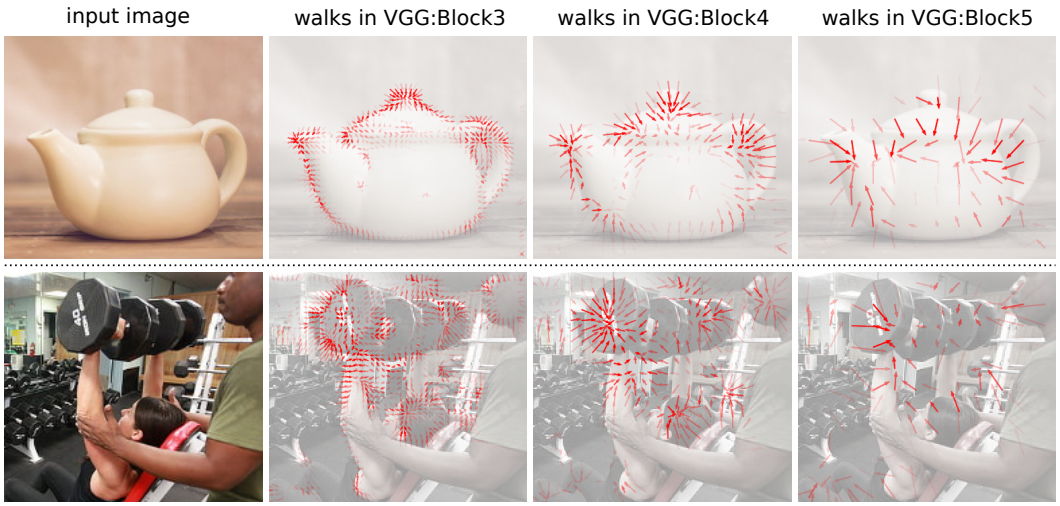
In parallel to the emergence of high-performing image recognition networks, explainable AI has adopted their use as the standard scenario to develop and evaluate explanations. The resulting heatmaps provide information about the importance of input image pixels and have been studied intensively [19, 16, 23, 17, 21, 22]. In this following use case, we revisit explanations for image classification and observe new ways to explain the inner model processing.

Setup A convolutional neural network (CNN) can be seen as a particular GNN operating on lattices of pixels. CNN predictions have so far mainly been explained using heatmaps highlighting pixels that are the most relevant for a given prediction [16, 23, 22]. Heatmaps are a useful representation summary of the decision structure, but they do not reveal the more complex strategies of a network that have been used to progressively build the prediction layer after layer. By viewing CNNs as graph neural networks and extracting relevant walks in the resulting pixel lattice, we demonstrate that the GNN-LRP method can provide explanations to comprehend these internal model strategies better. We consider the well-established VGG-16 network [160]. It consists of a collection of blocks interleaved by pooling layers, where each block is composed of a sequence of convolution and ReLU layers. We use the pretrained version of the VGG-16 network without batch normalization.

Efficient Computation The VGG-16 neural network is deep and the number of possible walks grows exponentially with neural network depth. Hence, we marginalize explanations to only consider the position of the walk at the input and at the output of a block. This is easily achieved by using a mask-based implementation and removing all masks except those at the input and output of the block. We then compute explanations for block 3, 4, and 5, respectively. To cope with the large spatial lattices in each block, we make use of a multi-mask strategy, which further accelerates computations by exploiting the local connectivity of nodes caused by the receptive field size of 7 in each of the VGG-16 blocks. We then identify selections of

nodes $(K)_{K\in\mathcal{K}}$ at a given layer, such that their receptive fields in the layer below are disjoint and thus, multiple walks can be processed in parallel by choosing the mask to be a grid with stride 7. This allows us to collect all relevant walks at the given block using 49 backward passes.

We analyze two exemplary images[2] that the VGG-16 network predicts as 'teapot' and 'dumbbell', respectively. We set the LRP parameter to $\gamma = 0.5$ in block 3, reduce the γ value in each subsequent block by a factor of two, and choose $\gamma = 0$ in the top-level classifier.



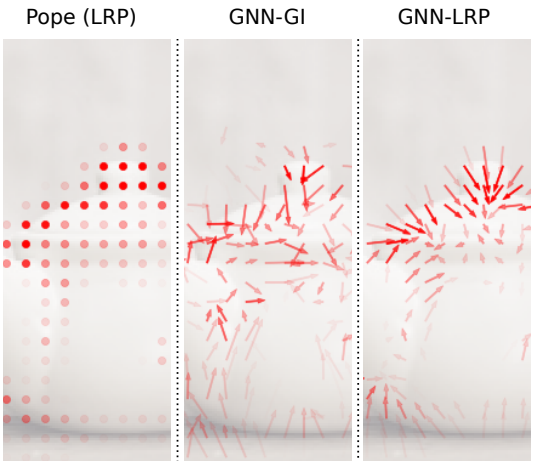


Figure 4.6: *Top:* Relevant walks in the pixel lattice explaining the prediction of the VGG-16 network on two input images that are correctly classified as 'teapot' and 'dumbbell', respectively. In each vector field, arrows connect block input nodes to the relevance-weighted average position of the block output nodes. *Left:* Detailed view of relevance flow comparing GNN-LRP with different explanation techniques on Block 4.

Results The resulting explanations are shown in Figure 4.6 (top) for the two images at multiple blocks of the VGG-16 network. For the first image, Block 3

[2]Images are from `https://www.piqsels.com/en/public-domain-photo-fjjsr` and `https://www.piqsels.com/en/public-domain-photo-fiffy`, rescaled and cropped to the relevant region to produce images of size 224×224, which are the standard input size for VGG-16.

detects local edges in the teapot, then, in Block 4, the walks converge to center points of specific parts of the teapot, e.g. the handle and the spout and the knob, and finally, the walks converge in Block 5 to the center of the teapot, which can be interpreted as composing the different parts of the teapot. For this exemplary image, we further observe in Fig. 4.6 (left) the advantageous properties of GNN-LRP compared to more basic explanation methods. The GNN-GI baseline also produces a vector field; however, it is significantly noisier than the one produced by GNN-LRP. The method by Pope et al. [237] robustly highlights relevant nodes at the input of the given block, but it does not reveal where exactly these features are being transported to for use in the subsequent block.

For the second image, we investigate a known 'Clever Hans' strategy where the network classifies images as 'dumbbell' by detecting both the dumbbell and the arm that holds it [258]. Using GNN-LRP we observe that Blocks 3 and 4 detect the arm and the dumbbell separately, and then Block 5 composes them into a single 'dumbbell-arm' concept, as shown by the walks for both objects converging to some center point near the wrist. These insights could not have been obtained from a standard pixel-wise heatmap explanation.

Overall, our GNN-LRP method can be used to comprehensively inspect the prediction of an image classifier beyond what would be possible with a standard pixel-wise heatmap explanation. This deeper explanation capability allows us to understand the detailed structure of image classifications better, and shed more light on anecdotal 'Clever Hans' effects observed in the context of a widely-used image classification model.

4.6 Summary and Discussion

In this chapter, we have focused on developing explanations for GNNs, which are a widely applicable and popular model choice for making predictions on problems best represented using graphs. The close entanglement of the input graph with the layered GNN computations makes the explanation of GNNs a challenging problem.

Our resulting GNN-LRP method reflects this by using groups of graph features that are derived from higher-order Taylor expansions and that go beyond solely considering input features. We have seen that GNN-LRP can be seen as a more general and robust case of our proposed GNN-GI baseline. In quantitative evaluation experiments, we have demonstrated that the additional explanation depth that arises from the nested interaction structure between GNN and input graph outperforms other graph explanation methods. We have introduced the activation and pruning task scenarios to measure a method's ability to explain the model prediction. Our relevance-based ordering of subgraph sequences extends standard approaches such as pixel-flipping to include higher-order information. The resulting AUAC and AUPC scores from our evaluation experiments on different GNN models and data have demonstrated the effectiveness of GNN-LRP.

The high quality and usefulness of GNN-LRP was further demonstrated in a use case in image classification. GNN-LRP explanations have enabled us to gain insight into the inner mechanisms of convolutional blocks and to trace the emergence of undesired 'Clever Hans' behavior in the popular VGG-16 model. A careful analysis of the GNN structure has thus resulted in robust and detailed graph explanations.

Limitations Computing higher-order interactions in GNN-LRP explanations, requires a full forward and backward pass for each possible walk through the graph model. The computational efficiency of the method is thus limited by the number of layers and the resulting exponential number of walks. For suitable applications, the required computations can be significantly reduced by coarse-graining of multiple graph nodes resulting in a pooling of walks that can be implemented efficiently into the masking approach. Alternatively, a partial computation of graph walks may be sufficient, e.g., by computing walks only for selected layers or defining a relevance threshold that skips less relevant walks. Recent work has shown that GNN-LRP can be computed more efficiently in linear dependence of the number of graph layers using a subgraph attribution approach [259].

Furthermore, GNN-LRP requires the implementation of tailored propagation rules, which also applies to other explanation methods such as GraphMask [248]. Some explanation techniques, including GNNExplainer [242] and PGExplainer [241], are based on a direct evaluation of the model function or its gradient, while the GNN-LRP method requires access to the internal representation at each layer to implement appropriate propagation rules. We have aimed to keep the additional implementation effort minimal by introducing the efficient implementation approaches introduced in Section 4.3.3.

To asses whether GNN-LRP explanations faithfully explain the model, we have used both synthetic and real-world datasets in our activation and pruning task analysis. Our results suggest that they do provide additional relevant information over simple attribution to input features. In order to also evaluate if walk-based explanations accurately match ground truth data in the context of more complex real-world scenarios, suitable data and ground truth annotations are needed.

The ability of GNNs to learn relations between features makes them both a flexible and powerful method for many complex problems. So far, we have considered walk-based explanations for single input instances. While this has offered interesting and detailed insights into specific decision strategies, it remains open how to extract dataset-wide prediction strategies from higher-order explanations.

Overall, a careful analysis of the structure of typical GNN architectures has motivated the explanation of their predictions using higher-order feature interactions. Compared to common approaches that attribute relevance to nodes, edges, or subgraphs at the input level, this results in novel walk-based explanations. This has broad implications for studying the prediction strategies of GNNs and can inform many important tasks, including model refinement and insight discovery across disciplines.

5

Transformer Explanations

Transformers have become an important architecture choice of machine learning and essentially present the default model of choice for any natural language processing task. This necessitates the development of reliable methods for increasing their transparency. Various explainability approaches have been proposed to understand inner processing beyond the inspection of attention weights that have been found to be of limited faithfulness with regard to the model prediction. More recent approaches have considered gradient information to represent the model prediction more faithfully. We demonstrate that the gradient in a Transformer reflects the function only locally. Our analysis identifies the attention head and the layer normalization layers as main sources of unreliable explanations and we develop a more stable way for propagation through these layers. This approach can be seen as an extension of the well-established LRP method to Transformer models. We observe both theoretically and empirically that it overcomes the deficiency of a naive gradient-based approach and achieves state-of-the-art explanation performance. In two Transformer use cases, we further study the usefulness of these explanations. We demonstrate how Transformer explanations can be used to investigate biased model behavior. In addition, we investigate the alignment between human reading patterns and attributions extracted from a variety of models and attribution approaches. We give complimentary insights on an observed trade-off between faithfulness, entropy and human correlation scores, and our analysis demonstrates the influence the choice of explanation method can have on the observed alignment to human attention.

This chapter is based on the following works and partly includes already published material from:

> [260] O. Eberle*, S. Brandl*, J. Pilot, A. Søgaard. Do Transformer Models Show Similar Attention Patterns to Task-Specific Human Gaze? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022. (*equal contribution)
>
> [261] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller and L. Wolf. XAI for Transformers: Better Explanations through Conservative Propagation. *(accepted at ICML)*, 2022.

5.1 Introduction

Transformer models [14] have attracted increasing interest and have shown strong performance in domains such as natural language processing (NLP) [14, 4, 262], vision [263], or graph tasks [264, 265]. Yet, their typically very high complexity (up to billions of parameters [266]) makes these models notoriously intransparent and their predictions inaccessible to the user. Transformer models are also applied in potentially sensitive domains, e.g., as support in cancer detection [267] or recruiting processes [268]. This further motivates the development of methods that explain their decisions since they allow to verify whether the model makes fair decisions and does not systematically discriminate against specific classes or user groups [269, 270]. This is especially urgent since many instances of biased Transformer predictions have already been identified [271, 272, 273, 274, 275], and a unified and robust way to evaluate fairness is still missing [276, 277].

To bring explainable AI to Transformer models, we focus on the axiom of conservation that underlies several popular explanation techniques, i.e., [23, 44, 22]. We embed our analysis in the layer-wise relevance propagation (LRP) framework [23], which allows us to analyze conservation at the level of individual modules and layers of the Transformer model. Our analysis reveals that the conservation properties of existing explanation techniques can severely break when extended to Transformer models. We introduce in the following how theoretically well-grounded Transformer explanations can be obtained. The resulting rules can be implemented in straightforward ways by 'detaching' parts of the forward computation and extracting explanations using specific backpropagation schemes.

We compare our approach to state-of-the-art baseline explanation methods for Transformers, including attention-based and gradient-based approaches. For the quantitative analysis, we perform different input perturbation schemes that track the model's behavior when least or most relevant features are added to or removed from the input sequence. We find that carefully handling the gradient results in clearly improved conservation and quantitatively better explanations that outperform the here considered baselines for most tasks. In two use cases, we investigate the usefulness of Transformer explanations to detect model bias and study their alignment to human reading.

5.2 Related Work

This section summarizes the different approaches proposed to explain Transformer models. These can be divided into methods that extract the attention scores from each Transformer block, treating them as directly interpretable attribution, gradient-based methods, and perturbation-based methods.

Attention-based Explanations The extraction of attention vectors from an attention module is a widely-used and intuitive way to understand or visualize the inner workings of a model [278]. It is a central design principle for the Transformer architecture, but it has remained difficult to assess whether attention scores are directly interpretable and faithful representations of the model prediction process. Growing evidence supports the hypothesis that attention may not be a robust explanation [279, 280]. Besides the simple extraction of the raw attention weights, there have been attempts to use the attention heads for defining more elaborate explanation mechanisms, such as 'Attention Rollout' and 'Attention Flow' [281]. More recently, other ways to aggregate attention information have been developed [282, 283]. These have provided empirical evidence that Transformers can be made explainable to a significant extent, especially in combination with gradient information.

Gradient-based Explanations Other authors incorporate gradient methods to explain Transformer models, such as integrated gradients [284] or input gradients [76]. The gradient methods Saliency, Gradient $\times$ Input or Guided Backpropagation have already been applied in numerous models and domains and were also applied to Transformer models [76]. In particular, there have been multiple attempts to implement the LRP method [23] in Transformers [285] and other attention-based models [105]. In addition, LRP has been applied to explain predictions of other models on NLP tasks [286], such as the popular BERT model [4]. Other approaches to gradient propagation in Transformer blocks were proposed in [282, 283], where the relevancy scores are obtained by combining attention scores with LRP or attention gradients.

Perturbation-based Explanations Additionally, different perturbation-based analyses have been used to gain insights into the processing of Transformers. Input reduction has been used to determine the most relevant parts of the input sequence by observing change in model confidence [287, 272] or computing Shapley values [44, 76]. Other approaches have focused on sequence probing, e.g., by permuting or swapping tokens and observing resulting changes in attention weights [288] or by measuring how strongly internal sequence representations are distorted by such permutations [289].

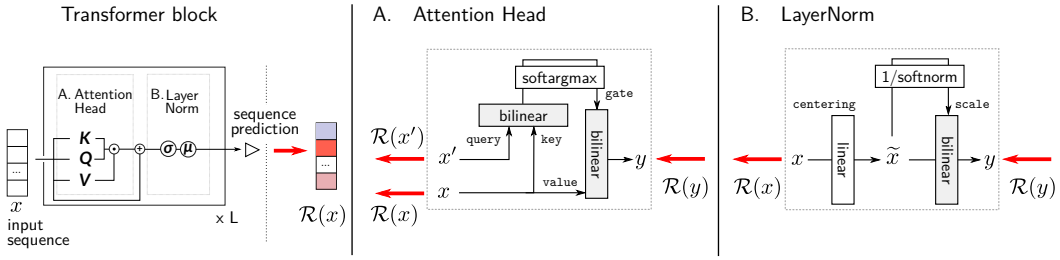


Figure 5.1: Extraction of explanations in Transformers models. Typical models are built from several blocks of Transformer blocks that extract sequence representations using nonlinear computations in attention head and layer norm modules. These nonlinearities include bilinear, softmax and division by the softnorm. The relevant variables used to compute layer-wise relevance scores $\mathcal{R}(x)$ are annotated. *A*: The attention head computes a matching between a set of key-value pairs that is used as a gating to decide what elements of the query are maintained. *B*: The layer norm layer computes elementwise statistics over the mean and standard deviation to normalize the block output.

5.3 A Theoretical View on Explaining Transformers

To gain a better theoretical understanding of the problem of explaining Transformers, we take the same 'axiomatic approach' used for analyzing and developing explainable AI in the context of standard deep neural networks [290, 291, 22, 39, 292]. A central axiom used in explainable AI, especially for the task of attribution, is *conservation*. The conservation axiom states that attribution scores assigned to input variables must sum to the observed model prediction score at the output. Central explainable AI methods that include LRP [23], Gradient $\times$ Input [21, 293, 292]), Integrated Gradients [22], or Shapley Values [291, 44], are either designed to satisfy conservation, have been shown to satisfy it, or can be derived from it directly. Overall, this highlights the importance of conservation as a desired principle for explanations.

The LRP framework [23] considers a particularly strong form of conservation, where each layer, component, or even neuron in the network is subject to the conservation axiom. In particular, the 'relevance' received by a given component (e.g., layer or neuron) from the layer above must be fully redistributed to the layer below. Many relevance propagation rules have been developed within the LRP framework to address the specificities of different data and architectures. Notably, the popular Gradient $\times$ Input (GI) method, commonly viewed as a gradient-based method, can be embedded in the LRP framework [21, 294, 39]. This can serve as a starting point for developing improved propagation rules. With this embedding into LRP, a detailed analysis of the GI explanation procedure can be performed to identify layers or components where the conservation breaks and derive better propagation rules for these layers in a second step. We provide a diagram of the standard Transformer attention block in Fig. 5.1 and illustrate relevance propagation

through the attention head and LayerNorm. Next, we turn towards an analysis of the gradient computation in these layers.

Denote by $(x_i)_i$ and $(y_j)_j$ the vectors of neurons representing the input and output of some layer or component of interest in a neural network, and by f the output of the neural network. To further analyse the attributions on these two vector representations, we compute Gradient $\times$ Input attributions, which can be seen as a special case of LRP [23, 39]:

$$\mathcal{R}(x_i) = x_i \cdot (\partial f / \partial x_i) \tag{5.1}$$
$$\mathcal{R}(y_j) = y_j \cdot (\partial f / \partial y_j). \tag{5.2}$$

Recall that the gradients at different layers of a neural network are related via the chain rule as follows:

$$\frac{\partial f}{\partial x_i} = \sum_j \frac{\partial y_j}{\partial x_i} \frac{\partial f}{\partial y_j}. \tag{5.3}$$

Injecting Eq. (5.1) and Eq. (5.2) into Eq. (5.3), the gradient propagation rule can be converted into an equivalent relevance propagation rule:

$$\mathcal{R}(x_i) = \sum_j \frac{\partial y_j}{\partial x_i} \frac{x_i}{y_j} \mathcal{R}(y_j), \tag{5.4}$$

where we use the convention $0/0 = 0$. With the embedding of GI into the LRP framework, we next study whether GI is conservative layer-wise or at the level of the given component, by testing if $\sum_i \mathcal{R}(x_i) = \sum_j \mathcal{R}(y_j)$ is satisfied. If this holds for every component of the neural network, then conservation also holds globally. We will now show that two components of the Transformer, namely attention heads and LayerNorm, cause a significant break of conservation and therefore require improved propagation rules.

5.3.1 Propagation in Attention Heads

Let us consider the attention head, which uses a multi-head attention mechanism that uses query, key and value embeddings and, which are the core component of Transformers [14]. Standard attention heads have the structure

$$Y = \mathrm{softmax}\Big(\frac{1}{\sqrt{d_K}}(X'W_Q)(XW_K)^\top\Big)X, \tag{5.5}$$

where $X = (x_i)_i$ and $X' = (x'_j)_j$ are the input sequences of token embeddings, $Y = (y_j)_j$ is the sequence of output embeddings, d_K denotes the dimensionality of the key-vector, and $W_{\{Q,K,V\}}$ are learned matrix projections. In the equation above, we omitted the multiplication by the embedding W_V, since the latter can be viewed

as a subsequent linear layer and can therefore be treated separately. For the purpose of our analysis, we rewrite Eq. (5.5) as:

$$y_j = \sum_i x_i p_{ij}, \tag{5.6}$$

where

$$p_{ij} = \frac{\exp(q_{ij})}{\sum_{i'} \exp(q_{i'j})}$$

is the softmax computation and $q_{ij} = \frac{1}{\sqrt{d_K}} x_i^\top W_K W_Q^\top x'_j$ is the matching function between two input sequences. Note that the output of Eq. (5.6) depends on the input tokens both explicitly, via the term x_i, and via the gating term p_{ij}, which itself depends on x and x'. Also, we observe that for each index j, we have an associated distribution $p_j = (p_{ij})_i$, and we denote by $\mathbb{E}_j[\cdot]$ and $\mathrm{Cov}_j(\cdot,\cdot)$ the expectation and covariance over this distribution, e.g. $\mathbb{E}_j[x] = \sum_i x_i p_{ij}$.

We analyze the relevance propagation associated with applying Gradient × Input to the Transformer model. We define input token relevance as $\mathcal{R}(x_i) = x_i^\top (\partial f / \partial x_i)$ and $\mathcal{R}(x'_j) = x'^\top_j (\partial f / \partial x'_j)$, and output token relevance as $\mathcal{R}(y_j) = y_j^\top (\partial f / \partial y_j)$.

In order to analyze gradient behavior in the attention head module, we first recall that output y_j is computed by a weighting of input x_i using attention probabilities p_{ij}:

$$y_j = \sum_i x_i p_{ij} \quad \text{with} \quad p_{ij} = \frac{\exp(q_{ij})}{\alpha_j} \quad \text{and} \quad \alpha_j = \sum_i \exp(q_{ij}),$$

with q_{ij} denoting $q(x_i, x'_j)$. Thus, we require partial derivatives of p_{ij} with respect to token sequences x_i and x'_j in order to compute the full module gradients.

As shown in [261], the following terms define how relevance is redistributed to the two input sequences x_i and x'_j in the attention head module:

$$\sum_i \mathcal{R}(x_i) = \sum_j \mathcal{R}(y_j) + \sum_j \mathbb{E}_j\big[q_{:j} \cdot (x - \mathbb{E}_j[x])^\top\big] \frac{\partial f}{\partial y_j}$$

and

$$\sum_i \mathcal{R}(x'_j) = \sum_j \mathbb{E}_j\big[(q_{:j} - \mathbb{E}_j[q_{:j}])x^\top\big] \frac{\partial f}{\partial y_j}.$$

After summing both relevance terms we arrive at

$$\sum_i \mathcal{R}(x_i) + \sum_j \mathcal{R}(x'_j) = \sum_j \mathcal{R}(y_j) + \sum_j \Big(\mathbb{E}_j\big[(q_{:j} - \mathbb{E}_j[q_{:j}])x^\top\big] + \mathbb{E}_j\big[q_{:j} \cdot (x - \mathbb{E}_j[x])^\top\big]\Big) \frac{\partial f}{\partial y_j}.$$

Making the further assumption that the first sequence of tokens and the softmax input both have expected value zero, we obtain the simplified form:

$$\sum_i \mathcal{R}(x_i) + \sum_j \mathcal{R}(x'_j) = \sum_j \mathcal{R}(y_j) + \sum_j 2\,\mathrm{Cov}_j(q_{:j}, x)^\top \frac{\partial f}{\partial y_j}.$$

This implies that conservation between layers may not hold in the presence of covariates between $q_{:j}$ and x. Since $q_{:j}$ is a function of x, such dependencies are likely to occur and we will introduce in Section 5.4 an alternative propagation rule that retains the conservation property and also results in more faithful explanations.

5.3.2 Propagation in LayerNorm

We now turn to an analysis of the commonly used 'LayerNorm' in Transformers, which computes normalization statistics over the hidden units in each layer. For our analysis, we focus on the core part of LayerNorm, consisting of centering and standardization (cf. [190]):

$$y_i = \frac{x_i - \mathbb{E}[x]}{\sqrt{\epsilon + \mathrm{Var}[x]}} \tag{5.7}$$

Here, $\mathbb{E}[\cdot]$ and $\mathrm{Var}[\cdot]$ denote the mean and variance over all activations of the corresponding channel (and potentially minibatch). The subsequent affine transformation can be handled using standard propagation rules for linear layers. Then, the core part of LayerNorm can be decomposed into two parts consisting of

$$\text{centering} \quad \widetilde{x}_i = x_i - \mathbb{E}[x], \quad \text{and rescaling} \quad y_i = \frac{\widetilde{x}_i}{\sqrt{\epsilon + \mathbb{E}[\widetilde{x}^2]}},$$

where $\mathbb{E}[\cdot]$ is computed over a uniform distribution, i.e. $\mathbb{E}[x] = \frac{1}{N}\sum_i x_i$ and value ϵ added to the denominator for numerical stability. An analysis of the relevance for both computation steps results in the following terms [261]: for the centering step this results in

$$\sum_i \mathcal{R}(x_i) = \mathcal{R}(\widetilde{x}_j),$$

and for the rescaling step in

$$\sum_i \mathcal{R}(\widetilde{x}_i) = \Big(1 - \frac{\mathbb{E}[\widetilde{x}^2]}{\epsilon + \mathbb{E}[\widetilde{x}^2]}\Big) \sum_j \mathcal{R}(y_j).$$

After combination with the centering step, we obtain the conservation equation:

$$\sum_i \mathcal{R}(x_i) = \Big(1 - \frac{\mathrm{Var}[x]}{\epsilon + \mathrm{Var}[x]}\Big) \sum_i \mathcal{R}(y_i),$$

where conservation holds only approximately for large values of ϵ. Thus, conservation is not satisfied and breaks especially strong when ϵ is small compared to $\text{Var}[x]$, which is usually the case.

5.4 Better LRP Rules for Transformers

This layer-wise analysis revealed deficiencies of Gradient $\times$ Input that we further address in the following. The integration of GI as as a special case of LRP lets us replace the implicit propagation rules in attention heads and LayerNorm, which we identified as breaking conservation, by specific propagation rules that conserve relevance by design.

Specifically, we make a locally linear approximation of the attention head during computation of explanations by viewing the gating terms p_{ij} as constants. Consequently, these terms can be interpreted as a linear weighting that locally maps the input sequence x to the output sequence y. We can then use the canonical LRP rule for linear layers defined as

$$\mathcal{R}(x_i) = \sum_j \frac{x_i p_{ij}}{\sum_{i'} x_{i'} p_{i'j}} \mathcal{R}(y_j) \qquad \text{(AH-rule)}$$

to propagate the relevance scores from the layer output to the layer input. With such a reformulation, we note that the query sequence x' appears disconnected, and consequently, we have implicitly $\mathcal{R}(x'_j) = 0$. Phrased differently, the relevance signal is not propagated through the attention weights and only considers the value features. This strategy has also been used and justified theoretically for LSTM blocks, [295] where it was shown empirically to yield superior performance compared to gradient-based methods, in particular Gradient $\times$ Input, and it was recently applied in image captioning models [296].

Furthermore, to address the particularly severe break of conservation in LayerNorm, we use again a locally linear approximation at explanation time, by viewing the multiplicative factor $\alpha = (\sqrt{\epsilon + \text{Var}[x]})^{-1}$ as constant. The LayerNorm operation can then be expressed by the linear transformation $\alpha C x$ with centering matrix C and entries $C_{ij} = \delta_{ij} - \frac{1}{N}$ and N denoting the length of the input sequence. Using again the same standard LRP rule for linear layers with weights αC, we obtain:

$$\mathcal{R}(x_i) = \sum_j \frac{x_i \cdot (\delta_{ij} - \frac{1}{N})}{\sum_{i'} x_{i'} \cdot (\delta_{i'j} - \frac{1}{N})} \mathcal{R}(y_j) \qquad \text{(LN-rule)}$$

where the factor α present in the numerator and denominator cancels out. We will next introduce efficient ways to implement these obtained rules for the attention head and LayerNorm computations.

Implementation of propagation rules

In practice, these rules do not need to be implemented explicitly and we observe that they are effectively the same rules as those induced by Gradient × Input, with the gating and rescaling terms in their respective layers treated as constant. By detaching the respective terms in the forward pass, we can prevent the gradient from propagating through them. In standard machine learning software, this can be implemented by adding a `detach()` call to a variable. This results in an updated computation graph that declares this variable as not requiring a gradient. We use the following implementation trick for the rules introduced above. To compute the improved LRP explanation, rewrite Eq. (5.6) as

$$y_j = \sum_i x_i [p_{ij}]_{\texttt{.detach()}}$$

in every attention head, and rewrite Eq. (5.7) as

$$y_i = \frac{x_i - \mathbb{E}[x]}{[\sqrt{\epsilon + \mathrm{Var}[x]}]_{\texttt{.detach()}}}$$

in every LayerNorm. Then we can extract the LRP explanation by simply calling Gradient × Input on the resulting function f. This implementation makes the method straight-forward to use, as it simply consists of adding `detach()` calls at the appropriate locations in the neural network code and then running standard Gradient × Input. Furthermore, the computation time is at least as good as Gradient × Input, or even better due to the simplified gradient computation.

5.5 Evaluating Transformer Explanations

The evaluation approach is tested on Transformer models trained on various datasets. We benchmark the performance of our method against other commonly used approaches for explaining Transformer architectures. As a first step, we analyse the desired principle of conservation and then continue with quantitative perturbation experiments.[1]

Datasets We use the following NLP datasets from natural language processing to evaluate the different explanation approaches. We consider sentiment classification on the SST-2 [256] and IMDB datasets [297] for binary classification into negative or positive sentiment. In addition, we use the TweetEval Dataset [298] for tweet sentiment classification, hate detection and emotion recognition. From the SILICONE Dataset [299] we use the tasks of emotion detection (Semaine) and utterance sentiment analysis (Meld-S).

[1] Our code for the implementation and analysis is available at: `https://github.com/AmeenAli/XAI_Transformers`

Benchmark Methods First, we compare to a 'Gradient×Input' [300, 21, 76] baseline without considering any modifications to the individual gradient computations as described in Section 5.3. In addition, we compute averages over last-layer attention head vectors ('Attention-last') [301], as well as 'Rollout' and attention flow ('A-flow') [281], which capture the layer-wise structure of deep Transformer models in comparison to raw attention head analysis. Attention flow views the attention network as a flow graph with nodes describing tokens and edges that define the maximal flow possible between them. The 'Generic Attention Explainability' (GAE) method combines attention gradients with attention relevance scores, resulting in state-of-the-art performance in explaining Transformer architectures [282].

We consider two variants to improve the LRP-based explanation technique. First 'LRP (AH)' where propagation through attention heads is handled via the AH-rule described in Section 5.4. For any other layer, we use the GI-equivalent propagation rule by simply propagating gradient without detaching terms. The second variant, which we call 'LRP (AH+LN)', additionally propagates through the LayerNorm using the LN-Rule.

5.5.1 Relevance Conservation

Setup To analyze the desired conservation properties and the insights about propagation rules from Section 5.3, we consider two Transformer models trained on the SST-2 and IMDB datasets. We compute both GAE and GI as well as LRP (AH), LRP (LN) and LRP (AH+LN) explanations. Since attention scores used in attention flow and rollout are normalized to be probability distributions they are not designed to be conservative and are not considered here. To inspect conservation properties, we compare the score produced at the output of the Transformer network against the sum of explanation scores over input features of the network. For this purpose, the input features are the positionally encoded embedding vectors present in the first layer. A fully conserving method results in points that lie on the identity diagonal line since no additional relevance should be produced or disappear.

Results The results are shown in Fig. 5.2. Our LRP (AH + LN) approach produces explanations that reflect the output score much more closely than GAE, GI and the partial application of the LN or AH rule, although mild breaks of conservation still occur. In addition, we observe that for GAE and GI the sum of explanation scores is not or very weakly correlated to the model output, which highlights the need for better Transformer explanations. Overall, this indicates that the proposed propagation rules work as intended and mitigate the lack of conservation in Transformer self-attention blocks. Next, we would like to verify if this is also reflected in an improved ability to explain the underlying model predictions.

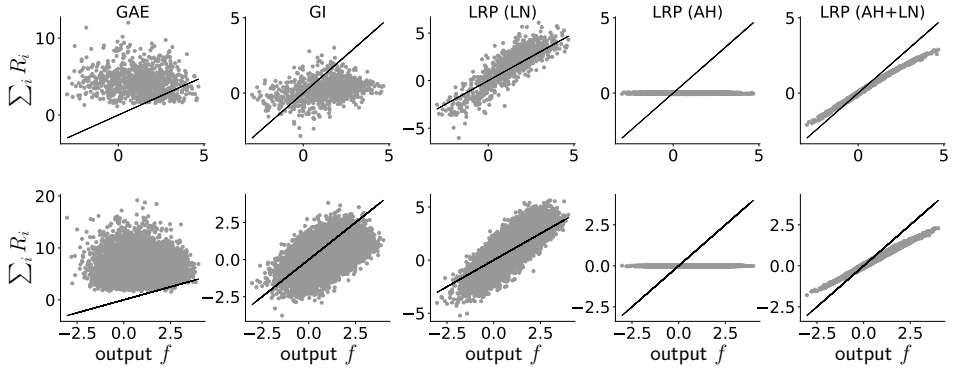


Figure 5.2: Conservation, or lack thereof, for different attributions in a Transformer model for sentiment classification on SST-2 (top row) and IMDB (bottom row). The x-axis represents the output score against the y-axis showing the sum of explanation scores over the input sequence. Each point in the scatter plot represents one dataset sample. The closer the points to the diagonal, the more conservative the explanation technique.

5.5.2 Quantitative Faithfulness Evaluation

We test the performance of different explanation methods using an input perturbation scheme in which the most or least relevant input nodes are considered.

Setup For the activation task, a good explanation gives an ordering from most to least relevant nodes that, when added to an empty sequence, activates the network output maximally and as quickly as possible. Thus, we observe the output probability $p_c(x)$ of the correct class c and report the area under the activation curve (AUAC) with higher AUAC indicating a more faithful explanation with respect to the correct prediction.

In the pruning task, we start with the original sequence and remove nodes in the order from smallest to largest absolute values. We measure AU-MSE, which is the area under the mean squared error $(y_0 - y_{m_t})^2$, with the model output logits of the unpruned model y_0 and y_{m_t} representing the sequence after applying the masking m_t at step t to the input graph. A lower AU-MSE is desired and indicates that removing less relevant nodes has little effect on the model prediction.

The activation task starts with an empty sentence of 'UNK' tokens, which are then gradually replaced with the original tokens in the order of highest to lowest relevancy. In the pruning task, we remove tokens from lowest to highest absolute relevance by replacing them with 'UNK' tokens [281].

Results In Table 5.1 we report results for the activation and pruning tasks and observe that the handling of the gradient in the attention head (AH) and layer normalization (LN) during backpropagation indeed results in consistently better performance across all datasets. We see the best performance when applying both

the detaching of gating and rescaling terms (AH+LN). In addition, explanations based on gradient information are superior to raw attention-based methods (A-Last, Rollout, A-Flow). Figure 5.3 shows activation and pruning curves for the SST-2 and IMDB dataset in a Transformer model. The application of the specific gradient rules leads to a gradual improvement over naive gradient implementations of GI, especially during the transition from very relevant to less relevant inputs for the activation task. This suggests that the improved LRP explanations are systematically more effective at determining the most relevant input nodes while attributing low relevance to task-irrelevant input nodes.

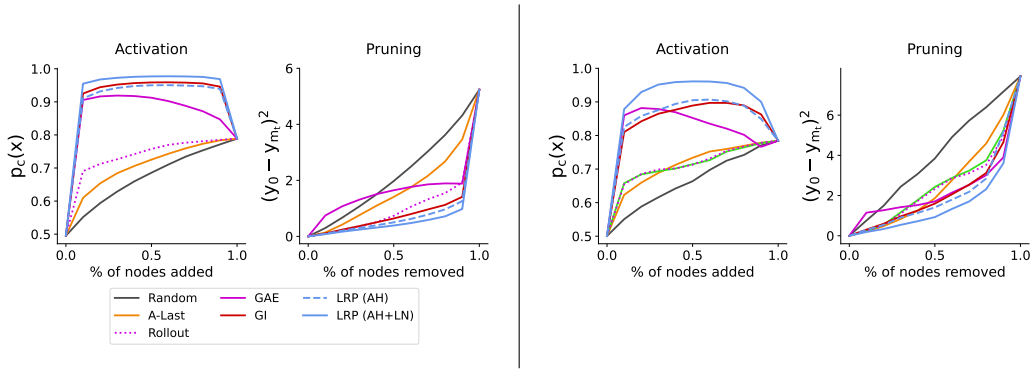


Figure 5.3: Evaluation of explanations using input perturbations. Results are shown for the activation task, where the most relevant nodes are added first, and for pruning, where nodes of least absolute relevance are removed first. *Left*: Results for SST-2 sentiment classification. *Right*: Results for the IMDB dataset.

Table 5.1: Evaluation of Transformer explanation methods across various datasets using activation and pruning tasks. Best performing methods are shown in bold. A-Flow is a computationally very expensive method and was therefore omitted for larger datasets. *Left*: Activation task scores as measured using the area under the activation curve (AUAC). Higher AUAC is better. *Right*: Pruning task scores as measured using the area under the pruning curve (AUPC). Lower AUPC is better.

Activation							
Method	IMDB	SST-2	T-Emotions	T-Hate	T-Sentiment	Meld-S	Semaine
Random	.673	.664	.516	.640	.484	.460	.432
A-Last	.708	.712	.542	.663	.515	.483	.451
A-Flow	-	.711	-	-	-	-	-
Rollout	.738	.713	.554	.659	.520	.489	.441
GAE	.872	.821	.675	.762	.611	.548	.532
GI	.920	.847	.652	.772	.651	.591	.529
LRP(AH)	.911	.855	.675	.797	.668	.594	.544
LRP (LN)	.935	.907	.735	.829	.710	.632	.593
LRP(AH+LN)	**.939**	**.908**	**.750**	**.838**	**.713**	**.635**	**.606**

Pruning							
Method	IMDB	SST-2	T-Emotions	T-Hate	T-Sentiment	Meld-S	Semaine
Random	2.16	3.97	4.25	9.12	2.87	2.54	1.92
A-Last	1.65	2.56	3.73	7.77	1.90	1.74	1.42
A-Flow	-	2.52	-	-	-	-	-
Rollout	1.04	2.43	2.85	6.55	1.71	1.53	1.40
GAE	1.63	2.26	2.21	7.40	1.61	1.56	1.37
GI	0.87	2.10	2.09	6.69	1.41	1.57	1.43
LRP(AH)	0.77	2.02	1.83	6.43	1.43	1.69	1.38
LRP (LN)	0.69	1.78	1.55	5.02	1.25	1.50	1.13
LRP(AH+LN)	**0.65**	**1.56**	**1.47**	**4.88**	**1.23**	**1.48**	**1.08**

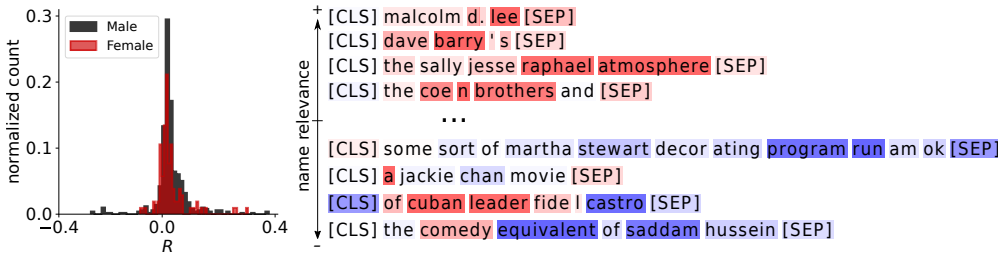


Figure 5.4: Investigating gender bias in a pre-trained DistilBERT model for sentiment classification on SST-2 movie reviews. Distribution of normalized name occurrences over relevance scores is shown along with sentence samples that contain names that most (top rows) or least (bottom rows) influence a classification towards positive or negative sentiment.

5.6 Use Case A: Analyzing Bias in Transformers

We now use our method on a popular Transformer architecture, DistilBERT [302], to study the detection of systematic bias in machine learning systems through explainable AI.

Setup We download the publicly available checkpoint for sentiment classification on SST-2 from HuggingFace[2] and apply the implementation trick introduced in Section 5.4. In order to detect such bias, template-based approaches, i.e. '<name> is a successful <job_title>', have been used to test the behavior of the model regarding different systematic relations between, for example, demographics and most likely model predictions [271, 272, 273, 274, 275]. While this is a flexible approach, it involves the risk of producing model inputs out of the training distribution and thus can cause unstable predictions.

Instead, we study relevance attribution to bias-sensitive groups of tokens that are of interest. This example explores the possible gender bias in sentiment analysis using the DistilBERT model. For this, we explain the *difference* between positive and negative model outputs for sentiment classification in order to observe which entities and related gender may exhibit a tendency to be more/less relevant to change the classification toward a positive or negative sentiment.

Results In Figure 5.4 (left), we observe that there is no consistent bias for female or male names in the DistilBERT model. Overall, there are more male than female names in the dataset, but the distributions of positive/negative sentiment attributed to them are similar. However, we do observe biased model responses towards certain entity categories: After ranking entities based on their assigned relevance from most to least relevant in Figure 5.4 (right) we observe that common Western male names such as 'lee', 'barry' or 'coen' can modulate sentiment the strongest towards positive. Interestingly, the first female entity, 'sally jesse raphael', is ranked high because of

[2] `https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english`

her typically male family name 'raphael'. At the other extreme, among the names with the strongest negative impact on sentiment, we find a non-Western family name ('chan') and male political figures ('saddam hussein', 'castro').

We further note that because our explainable AI-based approach disentangles the contribution of individual words from that of other words in the sentence, our approach is more immune to confounders than a naive approach that would simply look at the correlation between name occurrence and predicted sentiment.

5.7 Use Case B: Task-Solving in Humans and Transformers

The usefulness of learned self-attention functions often correlates with how well it aligns with human attention [113, 114, 115, 116, 303]. In the following, we evaluate how well attention flow [281] in large language models, namely BERT [4], RoBERTa [304] and T5 [305], aligns with human eye fixations during task-specific reading, compared to other shallow sequence labeling models [306, 14] and a classic, heuristic model of human reading [307]. To compare the different model attributions and the heuristic model across task-specific English reading tasks, we use a publicly available dataset with eye-tracking recordings of twelve English native speakers [308]. It contains data from two different task scenarios, one for sentiment analysis on SST movie reviews and the other for relation extraction on Wikipedia. [3]

5.7.1 Methods

Next, we briefly describe our used methods and refer to Appendix A.4.2 for additional details on model architectures, optimization and hyperparameter choices.

Human reading To compute human attention vectors, we extract and average word-based total fixation times across participants and focus on relation extraction and sentiment reading samples from the task-specific reading (TSR) subset of the ZuCo corpus. We refer to this as the 'TSR (ZuCo)' case in the following.

Models The superior performance of Transformer architectures across broad sets of NLP tasks raises the question of how task-related attention patterns really are. We use both pre-trained uncased BERT-base [4] as well as fine-tuned BERT models on the respective tasks. BERT was originally pre-trained on the English Wikipedia and the BookCorpus. Additionally, we use the RoBERTa model, which has the same architecture as BERT and demonstrates better performance on downstream tasks using an improved pre-training scheme and the use of additional news article data [304]. The Text-to-Text Transfer Transformer (T5) uses parallel task-training and

[3] Our code for the analysis and to reproduce the results is available at: `https://github.com/oeberle/task_gaze_transformers`

has demonstrated state-of-the-art performance on several transfer tasks including sentiment analysis and natural language inference [305].

In addition, we evaluate different ways of extracting token-level importance scores based on attention. We collect attention representations and compute the mean attention vector over the last layer heads to capture the mixing of information in Transformer self-attention modules [301], which we denote as *last* for the Transformer models considered here. In addition, we consider attention flow, which captures the layer-wise attention structure of Transformer models [281] .

In order to apply LRP to large-scale Transformer architectures, we have to consider any architecture choices that affect the gradient computation as discussed previously. The original BERT model differs from our previous scenario in the following ways: (i) it uses an overall deeper architecture that consists of 12 self-attention layers, (ii) the GeLU non-linearity is used as an activation function between two linear readout layers after each self-attention computation, and, (iii) a hyperbolic tangent is applied as an activation function after the pooling layer. We can treat the non-linear activation functions $\sigma(x) = \tanh(x)$ and $\sigma(x) = \text{gelu}(x)$ via application of the following trick that has been used in various forms for the efficient implementation of propagation rules [12, 226, 261]. It does not affect the forward prediction of the model but considers the gradient of the identity $id(x)$ during the backward pass and is by design relevance conserving:

$$y_i = id(x_i)\left[\frac{\sigma(x_i)}{id(x_i)}\right]_{\texttt{.detach()}}$$

Combined with the layer-wise application of the AH and LN rules, we then extract explanations from deeper Transformer models and, in our study, focus on task-tuned BERT models, which we refer to as the 'BERT (LRP)'.

We ground our analysis on Transformers by comparing them to relatively shallow models that were trained from scratch. We train a standard CNN [309] network with multiple filter sizes on pre-trained GloVe embeddings [310]. Another widely-used model that paved the way for Transformer-type models is the shallow multi-head self-attention network [52]. We use a version of this shallow self-attention model using GloVe vectors as embedding initialization in combination with a linear read-out layer. For both models, we extract relevance scores over tokens using LRP, for which we pool relevance over the embedding dimensions [311, 312].

As a cognitive model for human reading, we compute task-neutral fixation times using the E-Z Reader model [313]. The E-Z Reader is a multi-stage, hybrid model, which relies on an n-gram model and several heuristics that are based on theoretical assumptions and include the role of word predictability and average saccade length. Additionally, we compare to a frequency baseline using word statistics of the BNC (British National Corpus, Kilgarriff [314])[4] as proposed by Barrett et al. [115].

[4]We compute the negative log-transformed probability of each lower-cased token corresponding to an inverse relation between word-frequency and human gaze duration [315]

Metric To compare models with human attention, we compute the Spearman correlation between human and model-based importance vectors both on a token-level by concatenating individual sentences, and and on a sentence-level [301]. This enables us to distinguish unrelated effects caused by varying sentence length from token-level importance. As described before, we extract human attention from gaze (ZuCo), simulated gaze (E-Z Reader), averaged last layer attentions (BERT, RoBERTa, T5), relevance scores (CNN, self-attention) and inverse token probability scores (BNC).[5] We use tokenization as present in the ZuCo dataset to align sentences across tokenizers and apply max-pooling of scores when bins are merged.

5.7.2 Results

In the following, we present our results on how well different attributions patterns align to human attention during task-solving. We further investigate in what ways they differ by analyzing sparsity levels and assess faithfulness.

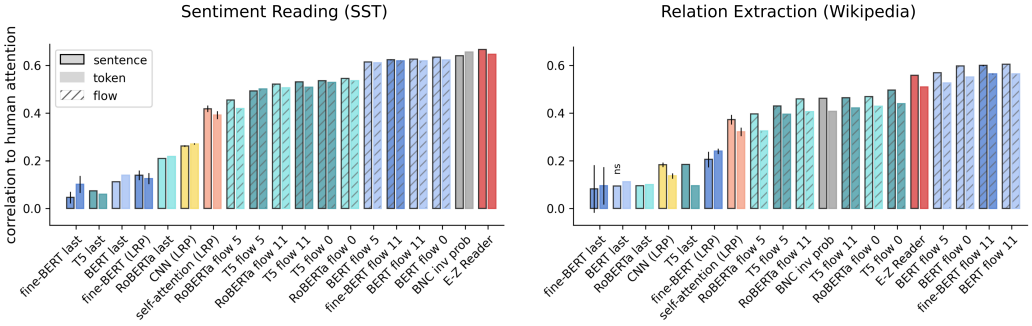


Figure 5.5: Spearman correlation analysis between human attention and different model attributions in two task settings. Solid bar edges indicate sentence-level correlations in contrast to a token-level analysis. *Left:* Sentiment Reading on the SST dataset. *Right:* Relation Extraction on Wikipedia. Correlations are statistically significant with $p < 0.05$ unless stated otherwise (ns: not significant).

Human correlation To measure how well model-based token attributions and human attention patterns for sentiment reading and relation extraction align, we compute pair-wise correlation scores as displayed in Figure 5.5. Reported correlations are statistically significant with $p < 0.05$ if not indicated otherwise (ns: not significant). After ranking based on the correlations on sentence-level, we observe differences between sentiment reading on SST and relation extraction on Wikipedia for the different models. For sentiment reading, the E-Z Reader and BNC show the highest correlations followed by the Transformer attention flow values. For relation extraction, we see the highest correlation for BERT attention flows (with and without

[5]First and last token bins from each sentence are ignored to avoid the influence of sentence border effects in Transformers [53] and for which the E-Z Reader does not compute fixations.

fine-tuning) followed by the E-Z Reader. On the lower end, computing means over Transformer attentions across the last layer shows weak to no correlations for both tasks. The shallow architectures result in low to moderate correlations with a gap to the best attention flow correlation scores. BERT (LRP) shows relatively weak correlation, more similar to levels of Transformer last, and lower than the shallow self-attention model.

Sparsity analysis Averaged sentence-level entropy is used to measure sparsity levels of the different attributions. We compensate for the different sentence lengths by performing a stratified analysis such that every length occurs equally often. As summarized in Table 5.2, we find that BERT, RoBERTa and T5 attention flow, the E-Z Reader and BNC baseline obtain similar levels of sparsity as human attention (at around 3.4-3.6 bits). Entropies are lower for the shallow networks with self-attention (LRP) at 1.8-2.2 bits and CNN (LRP) at around 2.9 bits. The BERT explanations extracted using the LRP (AH+LN) rule result in moderate entropy levels between the very sparse shallow models and the least sparse cases (TSR, E-Z Reader, Transformer flow) at levels of 2.9-3.1 bits. The shallow models were trained from scratch for the respective tasks whereas all other models (including human attention) are heavily influenced by a more general modeling of natural language, which might explain why Transformer attributions are more broadly distributed over all tokens. This analysis thus highlights that attributions show varying levels of sparsity across models and attribution methods.

Table 5.2: Mean entropy over all sentences for each task setting. Lower entropy means sparser token importance. The maximal entropy of a uniform model is at 4.09 bits.

	TSR (ZuCo)	E-Z Reader	BNC inv prob	CNN (LRP)	self-attention (LRP)	BERT (LRP)	BERT flow 11	RoBERTa flow 11	T5 flow 11	BERT last	RoBERTa last	T5 last
SR	3.44	3.44	3.40	2.93	2.16	3.11	3.57	3.61	3.61	2.37	2.65	2.45
TSR	3.38	3.46	3.39	2.98	1.81	2.92	3.54	3.60	3.63	2.48	2.56	2.29

Faithfulness We further test the ability of both human and model-based attributions to activate the correct output neuron using a gradual flipping of features [287]. By this, we aim to test how effective the exact token ranking based on attribution scores is at producing the true output probability in a task-tuned BERT model. As presented in previous chapters, such measures of faithfulness are typically used to test how sensitive a given model responds to a specific flipping order as provided by an explanation of this same model. Here, we perform the analysis according to a ranking order that we extract from a set of models or human-based rankings. Faithfulness can now be understood as the combination of this model and its respective attribution approach. Hence, a high score may not only be achieved by

a faithful method but also by a model that processes information in favorable ways, e.g., models that by design produce more sparse explanations.

In the following, we perform the flipping of features using fine-tuned BERT models for both sentiment classification and relation extraction as the reference model and present results in Figure 5.6. In our analysis, we observe that adding tokens according to absolute token probability (BNC prob) results in worse performance than randomly adding tokens as to be expected. From-scratch trained models (CNN and self-attention) are more effective in selecting task-relevant tokens than any Transformer attention flow, the E-Z Reader and human reading. The BERT explanations using the improved LRP rule (AH+LN) result in the most effective Transformer explanation, with only the CNN (LRP) achieving a more faithful explanation. In parallel, the most sparse shallow self-attention explanations are less faithful. Adding tokens based on human attention is as effective for the sentiment task as the E-Z Reader. For the relation extraction task, human attention vectors provide the most effective flipping order after the relevance-based methods and we observe that all Transformer flows perform comparably in both tasks.

The observed difference in sparsity levels might explain the advantage of CNNs and shallow self-attention models in this analysis. The early addition of few but very relevant words has a strong effect on the model's decision when compared to less sparse attributions as observed in Transformer flow or human reading.

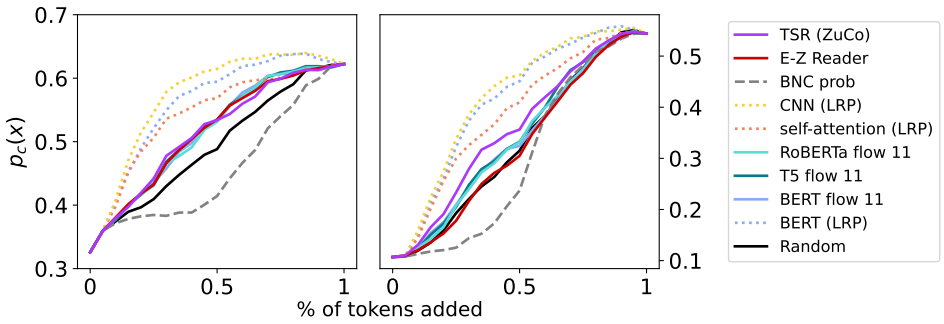


Figure 5.6: Feature flipping experiment. Tokens are ordered according to different attribution methods and added from most to least relevant to an empty input sentence. Observed probability for the true class is shown. *Left*: Activation curves for sentiment reading on SST-2. *Right*: Results for relation extraction on Wikipedia.

To summarize, this case study illustrates the trade-off between human alignment, sparsity levels and model faithfulness. In our experiments, we observe that the more sparse and faithful explanations with regard to a task-tuned BERT model show moderate correlation to human fixations, and and vice versa, the strongest correlation to human reading patterns is observed for less faithful explanations. This is in line with previous findings that gradient-based methods are often more faithful [279, 280, 283, 282], but more faithful explanations do not always agree more with human rationales [75, 316]. Our findings highlight that the selected explanation method can strongly affect the observed alignment between machine learning model

and human rationales. Therefore the selected approach should be carefully evaluated using objective explainable AI measures to ensure that the explanation reflects the task model's prediction process.

5.8 Summary and Discussion

In this chapter, we have proposed better explanations for Transformer models. The central role of Transformers in machine learning and their strong uptake in practical applications have highlighted the importance of bringing transparency to their decisions. Within the LRP framework, we have shown that Gradient $\times$ Input fails to implement conservation, a common property of attribution techniques. Our analysis has highlighted that specific architecture choices such as attention heads and layer normalization need to be addressed specifically in order to ensure conservation and faithful explanations. Our experiments have demonstrated that our method systematically achieves state-of-the-art faithfulness scores. In addition, we have showcased our explanation technique on the problem of detecting biases in a widely used sentiment classification model. Our explanation technique was able to characterize model bias in a detailed manner without having to generate counterfactual examples and the risk of stepping out of the data manifold. In another use case, we have investigated the alignment of different attribution approaches to task-specific human reading in English native speakers. We found that high faithfulness does not equal high correlation with human rationales. This suggests that both pre-trained language models *and* humans are regularized by natural language contexts, which can result in suboptimal task-solving strategies that do not necessarily present the most effective solution to a task.

Limitations Our experiments have focused on sequence classification scenarios consisting of a sequence encoder combined with a readout module that computes the model prediction. In practice, Transformers have been used in many task settings, which have introduced a great variety of architectures built from basic Transformer sub-modules. This includes additional encoder-decoder attention modules in sequence-to-sequence tasks [14, 305], or cross-modality encoders and co-attention modules [317] for vision-and-language reasoning [318]. Generally, we observe that the great variety of Transformer architectures requires careful implementation of propagation rules and specific architecture choices may need additional treatment, as we have observed in Section 5.7.1 for specific non-linear activation functions.

In our evaluation of Transformer explanation methods, we have focused on faithfulness and the desired principle of conservation. Faithfulness has arguably become the standard way of assessing how well an explanation method is able to explain the model predictions. Our choice of including conservation as a desired principle was informed by observations that explanations on basic Transformers were not able to produce plausible attributions using naive Gradient $\times$ Input. As we have demonstrated, the observed relevance did not conserve the model prediction

score throughout the self-attention modules. Further theoretical and experimental analysis revealed that the attention head and layer normalization modules are the main factors that lead to a break in conservation. We have thus used conservation as a guiding principle to inform the development of better propagation rules and perform objective evaluation using activation and pruning tasks.

In our use case A, we have demonstrated that explanations are a promising candidate to detect model bias and increase the fairness of machine learning systems. During the design of our study on gender bias in movie reviews, we have found that this approach can be difficult to implement since suitable datasets are sparse and template-based approaches lack robustness. We note that the here investigated case of representational gender bias is only one example of many possible ways models can be biased in undesired ways [319, 320].

In use case B, we have investigated the relation between human reading patterns and different model attribution methods during task-solving. We observe that the choice of the method can strongly affect the respective alignment to human attention, as shown by the differences in correlation scores, e.g., between BERT attention flow and BERT (LRP). This emphasizes that in order to make robust inferences from explanations, the selected explanation method should be assessed thoroughly. Explanations that best explain the model prediction, may not coincide with human reading and task-solving strategies since models do not necessarily learn human-like concepts, and humans might over-look corpus correlations between labels and simple features [321]. Thus, perfectly aligning humans and machines may not be feasible or even a desired goal, yet human gaze information was shown to be a useful learning signal to improve model performance: Human attention patterns were used to regularize learning during model training resulting in comparable or improved task performance for part-of-speech tagging [322, 118, 115], sentence compression [114], detecting sentiment [323, 119] or reading comprehension [120]. In these works, gaze data is used without consideration of the specific task, questioning to what extent human reading is task-modulated and if models would even be able to further benefit from task-specific human signals. To better understand such internal model mechanisms, faithful explanations can help to gain more nuanced insights into task-related strategies that go beyond observing parameter changes in response to task-tuning [324, 325].

Overall, our analysis of Transformer explanations has highlighted that common explainable AI methods will not necessarily continue to work well on Transformer models. We have seen that desired principles for explanations such as conservation can guide the way towards improved propagation rules. The resulting explanations offer to analyze the steadily growing number of Transformer variants and allow insights for their understanding and improvement.

6

Conclusion and Outlook

We conclude this thesis with a summary of our key findings and discuss their wider implications and contributions thereof. This includes an outlook on promising future research and application areas.

6.1 Summary and Discussion

This thesis has aimed to bring interpretability to highly predictive models that have gained wide popularity in the machine learning community but remained challenging to make explainable. The specific structure of such models motivated the use of carefully designed explanation techniques. We have considered three such model architectures, deep similarity models, graph neural networks and Transformers, and demonstrated how propagation-based methods could be developed to compute robust explanations for each scenario.

6.1.1 Methods

In Chapter 3, we have introduced 'BiLRP', which is an explanation method for deep similarity models. It decomposes the dot product similarity score on pairs of input features. BiLRP is embedded into the framework of the deep Taylor decomposition method, which resulted in second-order terms that represent the interaction between features. These accurately describe which pairs of features are most or least relevant to produce a particular similarity score. In experiments, we have confirmed that BiLRP gives more robust explanations compared to approaches such as Saliency or Hessian $\times$ Product that we have proposed here. Our analysis of similarity models has thus enabled detailed insight into model mechanisms and data structure.

Next, we have introduced 'GNN-LRP', a method that computes explanations in graph neural networks in Chapter 4. Their layered aggregation-and-combine structure on the input graph has resulted in higher-order terms during the derivation using a deep Taylor expansion. We have described these higher-order interactions between nodes via walks through the network. In experiments on synthetic and real-world data, we have found that GNN-LRP outperforms other graph explanation methods that rely solely on explanations in terms of input nodes or edges, which suggests that the model is explained more faithfully using higher-order information. Both BiLRP and GNN-LRP have illustrated how higher-order information naturally emerges in a layer-wise, Taylor-based decomposition of the model prediction and presents an appropriate explanation complexity for these models. These explanations additionally offer the flexibility to explain at different levels of granularity by pooling relevance over specified dimensions such as image patches or subgraphs. The consideration of high-order information should thus be informed by the structure of the model and input, and generally, the lowest available explanation complexity should be chosen to avoid an unnecessarily complex attribution.

For Transformer models, we have observed that the break of conservation caused by specific structures in the self-attention blocks has resulted in unsatisfactory explanation quality. In Chapter 5, we have developed better explanations for Transformer architectures and introduced LRP propagation rules that handle the non-conserving layer normalization and attention head modules. We have demonstrated that these improved LRP rules result in more faithful explanations that can explain the Transformer prediction process better than naive gradient computation and previously proposed Transformation explanation approaches.

These methods contribute to a novel direction of explainable AI methods that go beyond heatmaps over input features if needed and demonstrate how robust explanations can be designed and evaluated by considering their distinct structure.

6.1.2 Evaluation

For the objective evaluation of explanations, we have used activation task procedures that iteratively add the most relevant features to observe how strongly they activate the correct output neuron. A similar procedure was introduced to remove the least relevant features to check if the model is only minimally affected. We have contributed an extension of pixel-flipping to higher-order explanations, which considers the additional information available when deciding which feature to select next. The resulting area under the curve serves as a measure of faithfulness, which is arguably the most commonly used approach to quantify a method's ability to explain the model. In addition to the evaluation using faithfulness, we have observed that the principle of conservation is not met by common Transformer explanations, including Gradient $\times$ Input. We have further described under which conditions conservation is fulfilled for BiLRP, GNN-LRP and the Transformer LRP propagation rules.

Additionally, we explored a case study in which we compared language models to human reading patterns. We investigated the agreement between task-dependent

human attention with different model-based attribution vectors and found that the quality of the used explanation method can strongly affect the correlation strength between humans and machines. We have observed that faithful explanations tend to be sparser than human attention vectors, and conversely, explanations that match human sparsity levels, i.e., Transformer attention and attention flow, have resulted in lower faithfulness. Our results suggest that the alignment of humans to language models depends strongly on the selected model architecture and the selected explanation method.

6.1.3 Robustness of Models

We have identified numerous instances of 'Clever Hans' behavior in popular machine learning models. By viewing VGG-16 as a graph, our GNN-LRP method enabled us to trace how objects are assembled from distinct parts throughout the convolutional encoder blocks. This allowed us to observe how co-occurring objects, e.g. a 'dumbell' that often appears together with an 'arm', are merged into a single object 'arm-dumbell'–a model strategy that will not generalize well. This previously observed effect (cf. [258]) could now be analyzed using the detailed information provided by GNN-LRP explanations, which provide information for future mitigation strategies.

We further demonstrated the usefulness of explanations to identify flawed strategies and improve machine learning models. The lack of rotation and translation invariance during the processing of historic illustrations could be mitigated using the insights offered by BilRP. A similar lack of invariance emerged in the frame-to-frame analysis of sports videos. While invariance score across VGG-16 layers were highest for the last encoding layer 31, the explanations produced by BiLRP revealed undesired model behavior. This includes 'Clever Hans' cases, for which similarity between frames was attributed surprisingly to interactions of background features instead of subjects in the foreground. Thus, high nominal accuracy or invariance scores can be misleading and explainable AI offers a way to verify these model properties to guide model development towards high robustness and performance.

6.1.4 Application and Insights

In a series of use cases, we have used our explanation methods to extract insights and explored their role in enabling new applications and research directions. In close collaboration with historians of science, we have developed a machine learning-assisted approach that allowed us to infer the similarity structure in a highly heterogeneous corpus of early modern computational tables. This enabled us to study the emergence and evolution of science in the 15th to 17th century at unprecedented scale and granularity. Using BiLRP, we have verified that our model that relies on representing a dense numerical table via a bag-of-bigrams, e.g. '01' or '98', indeed grounds similarity on the desired numerical features instead of attributing relevance to task-irrelevant page elements (such as drawings or surrounding text). Our automated corpus-wide analysis has enabled us to study the spatio-temporal evolution of knowledge

and innovation, which highlighted singularities that could further be supported by historical evidence.

In NLP, modern corpora have demonstrated to bias large-scale language models towards undesired predictions that are based on characteristics such as gender, ethnicity or age. Instead of probing bias-sensitive tokens in a template-based approach, our developed Transformer LRP-based explanations have allowed us to directly quantify how relevant a given entity is to change the sentiment prediction towards a positive or negative classification. In our experiment using the widely-used DistilBERT Transformer model, we did not observe a corpus-level gender bias. However, we have identified biases that are specific to certain categories, e.g., Western male names are most likely to produce a positive sentiment, whereas non-Western surnames have the opposite effect. Hence, this relevance-based approach has offered detailed insight into biases by comparing the relevance of an entity to influence the model prediction.

The development of robust explainable AI methods is thus a crucial step to investigate and resolve pressing issues of modern machine learning models.

6.2 Outlook

After the analysis of our results, we now provide an overview of promising future directions for the development of explainable AI methods. We focus in particular on directions that demand considering the model structure and ways to evaluate and gain insights from resulting explanations.

6.2.1 Transfer to Similar Structures and Models

The increasingly complex design of modern architectures often requires that methods such as LRP have to be continuously extended to robustly explain these models. Directly applicable gradient-based methods may produce satisfactory results for simple models, but gradient behavior becomes unstable for deep models [41]. This necessitates to develop more robust explanations. In our experiments, we evaluated and used different explanation methods for a selection of different architectures and domains. These insights and analyses can be further extended and applied to obtain better explanations for complex models in the future.

Many practical challenges in machine learning require inputs from different sources, e.g., in hospital applications this can include medical scans and text data about a patient's condition, which motivates multi-model learning [326, 327, 328]. Modeling multi-modal data requires considering interactions between the respective feature representations and using an intermediate fusion step to combine different modalities. Highly non-linear deep architectures are commonly used to learn suitable representations, and dot products are a natural choice to detect joint patterns across modalities [329, 317, 328]. This structure makes computing explanations technically challenging, and our BiLRP approach of explaining similarity predictions is well-suited to explain such multi-modal feature interactions. Additionally, the analysis of

not directly interpretable latent variables, e.g., in the context of generative adversarial networks and learning shared latent spaces using dot products, have so far required specifically designed explainable models to become interpretable [330, 331, 332, 333]. The BiLRP approach offers to explain such latent variables to understand respective feature interactions better.

The popularity of GNNs has resulted in many use cases, including temporal sequences [334, 335] and multi-modal data [336, 337, 338], that we have not considered here. These approaches may require specific treatment to arrive at walk-based explanations through time. We expect that GNN-LRP extends naturally to variants of standard GNNs that can be formulated via repeated aggregate-and-combine computations on an adjacency matrix.

Relation networks that represent relational reasoning processes gained increasing interest and learn complex relations between features [339, 340]. Conceptually, these approaches use specifically structured computations such as attentional similarity and graph networks to represent interactions between entities [340, 341, 342, 343, 344]. Our introduced explanation methods can be used to highlight the relevant higher-order feature interactions, which enables detailed insights into complex reasoning strategies.

Related graph approaches have used Transformer-based architectures [264], which motivate to capture higher-order interactions with a focus on the joint contribution of data features. This again highlights that considering structure is important during selection or development of appropriate explanation methods.

The unwavering influence of Transformer models and modules on machine learning has resulted in numerous extensions, modifications and applications of the model considered here. This large variety motivates to develop explainable AI that robustly explains these variants. For example, a seemingly small change of an activation function can severely break conservation, and hence modifications such as cross-modality encoders and co-attention modules [317, 318] must be carefully analyzed before extracting propagation-based explanations. More work and community efforts are needed to implement robust explanation techniques for the most commonly used architectures.

6.2.2 Evaluation Datasets and Ground Truth Explanations

To evaluate progress in explainable AI in consistent ways, it is important to distinguish between approaches that test the ability of an explanation method to faithfully reflect the model prediction, and its ability to match a given ground truth.

Common procedures that compute faithfulness of an explanation method rely on masking certain parts of the input to observe change in the model prediction. This served as a starting point to develop a masking-based approach that is applicable to higher-order explanations as introduced in Section 4.4.2. Moreover, it can serve as a general strategy for future evaluations of higher-order feature interactions, and is also applicable to second-order interactions, e.g., in similarity models.

This extension can require to extract subgraphs (cf. Section 4.4.2). Since evaluating all possible subgraphs is computationally not feasible for complex models, we have used approximation schemes, i.e., local approximation. Further development of approaches to extract relevant subgraphs in efficient and accurate ways poses an important future direction.

To address how well an explanation method is able to match a given ground truth, requires to develop and make available standardized datasets. In order to challenge explanation methods that consider higher-order information, these should be sufficiently complex and go beyond binary or few label classification settings. Since currently no appropriate ground truth data that considers higher-order interactions exists, evaluation procedures so far have to be designed to match this ground truth of lower complexity. For graph explanations, the BA-2motif dataset [241] has provided ground truth edge annotations for two types of classes that differ in one particular edge and has been used to confirm the effectiveness of GNN-LRP in addition to perturbation-based evaluation [226]. While the higher-order information did provide an advantage when compared to standard input explanations, it would be interesting to observe what data and tasks are structured in ways that can not be explained accurately using standard explanations alone.

Thus, the development of more sophisticated synthetic datasets is an important future direction. This allows to control the underlying features and the level of ground truth complexity better. Tasks that are best described using such higher-order interactions are especially suited. Ground truth that evolves along additional, e.g., spatial or temporal, axes could provide appropriate data which may be used in causal reasoning graphs [345, 346], or task-solving in temporal networks [334, 335]. In addition, pairwise interactions have recently been used to test directly feature importance for relational reasoning between objects [339, 175], which could provide a future benchmark for second-order methods on synthetic data.

Collecting ground truth annotations for real-world data that considers higher-order information is generally challenging. These require annotations on a fine-grained level that match the level of granularity of features that the prediction is grounded in, e.g., the matching of cat ear and eye features for similarity between cats, while focusing less on overall body outline or other class-typical features. For example, in the case of similarity, the lack of a clear definition of what features make two data points similar further impedes the collection of ground truth. It requires to decide on what level annotations are to be collected: high-level features such as individual object parts or more low-level color or texture features.

Human similarity judgment tasks developed to evaluate the agreement between humans and machines [347, 348] may be useful to provide pairs of input data that are considered similar by annotators, thereby using an implicit definition of similarity. In order to directly measure if the appropriate features are used by the model to build similarity, appropriate segmentation and their pair-wise interactions will be necessary to quantify the accuracy of explanations.

While the human evaluation of explanations is often acknowledged as an important direction, it remains difficult to define robust protocols to collect ground truth

explanations and measure their alignment, especially for scenarios that consider interactions between features. The resulting limited availability of data and the large variety of study designs currently impede the development of standardized community benchmarks. Future work is needed to define and unify the gold standards to measure agreement between models and human data, e.g., using information-theoretic approaches [349]. As we have observed, the eye fixation-based attention scores in human reading are strongly regularized by natural reading patterns, which illustrates the need to separate such task-independent priors from task-specific contributions.

6.2.3 Better Models using Explainable AI

We have seen that developing reliable explanations requires a careful analysis of models and methods. Assuming we have access to such explanations, another important future challenge is to use them to improve machine learning towards more efficient, robust and fair models.

Our analysis that focused on particular model structures, e.g., graph networks or self-attention modules, can inform the design of future architectures. Using well-studied model components over very specific network functions that do not significantly boost performance can thus facilitate to make machine learning more transparent and connect them to existing theoretical frameworks. This approach can provide a trade-off between selecting less flexible self-explainable models, and using inexplicable high-performing models.

The identification of undesired prediction strategies is an important first step to improve performance of machine learning models. As we have observed throughout this thesis, models can be flawed in many different ways, e.g., caused by a lack of invariance or spurious correlations between features, which motivates to alleviate or remove them in strategic and reliable ways. The automated detection and mitigation of model artifacts has been explored recently [128], and future work is needed to investigate how the additional information provided by higher-order explanations can be used to correct flawed model behavior.

The widening reach of more complex models, with application to sensitive and high stake areas such as medical diagnosis, automated decisions-making, or finance, emphasizes the need for transparent and robust predictions. Explanations are thus crucial to detect and correct unwanted model biases, for example, when using similarity models for information retrieval tasks across diverse demographics or Transformer models for robust language processing in fair and transparent ways. Propagation-based methods hereby offer direct ways to decompose model predictions at different processing steps and enable pinpointing the modules or processing layers that are most sensitive to bias models in undesired ways. In addition, approaches that investigate diverse types of biases and systematically detect them at scale are much needed in future work.

Explainable AI thus plays a central role in guiding this search for better architectures, data and optimization schemes, and our methods provide reliable and detailed explanations to inform this process.

6.2.4 Scientific Insights

The increasing use of machine learning across various scientific fields has not only enabled the large-scale analysis and automatic organization of big data but has also started to be a valuable tool for the generation of novel domain insights, e.g., in quantum chemistry [350, 230, 6, 351], the climate and earth sciences [352, 353, 354], astronomy [355, 356], biomedicine [357, 358] or neuroscience [359, 360, 361, 362]. Besides the direct insights gained from these models, explainable AI offers to verify these model computations and brings an additional layer of information on the level of individual inputs, full datasets or corpora, and model computations. The potentially high complexity of problems encountered in the sciences motivates the use of specifically structured models. In order to extract domain insights and information about the structure of the problem itself, appropriate explanations are needed. For example, GNN-LRP can support researchers by providing detailed and scientifically valuable explanations in graph-based models for quantum chemistry [226].

Such machine learning-assisted insight discovery has most widely been used in the natural sciences [363], and consequently, other domains have been challenged to profit from machine learning techniques. In future work, under-represented languages or low-resource problems in the historical sciences can benefit from better Transformer explanations to verify robust model behavior in the absence of annotated material and ground truth explanations. Since the here addressed architectures are popular choices for many of the aforementioned machine learning applications, having access to faithful explanations opens vast possibilities to study models and data in the future.

In conclusion, explainable AI is a powerful and important direction for making machine learning usable. The complexity and structure of state-of-the-art architectures require careful theory, implementation and evaluation to extract meaningful explanations and build safe, robust and trustworthy models.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. Burges, L. Bottou, and K. Weinberger. Vol. 25. Curran Associates, Inc., 2012.

[2] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*. Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[3] V. Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 00280836.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*. 2019, pp. 4171–4186.

[5] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

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

[7] T. Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[8] G. Montavon, W. Samek, and K. Müller. "Methods for interpreting and understanding deep neural networks". In: *Digit. Signal Process.* 73 (2018), pp. 1–15.

[9] A. Holzinger. "From Machine Learning to Explainable AI". In: *IEEE 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. IEEE Computer Society, 2018, pp. 55–66.

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

[11] E. Tjoa and C. Guan. "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (11 Nov. 2021), pp. 4793–4813. ISSN: 21622388.

[12] 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". In: *Proceedings of the IEEE* 109.3 (2021), pp. 247–278.

[13] D. Alvarez-Melis and T. S. Jaakkola. “Towards Robust Interpretability with Self-Explaining Neural Networks”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Montréal, Canada: Curran Associates Inc., 2018, pp. 7786–7795.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[15] W. Brendel and M. Bethge. “Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet”. In: *International Conference on Learning Representations* (2019).

[16] M. D. Zeiler and R. Fergus. “Visualizing and Understanding Convolutional Networks”. In: *ECCV (1)*. Vol. 8689. Lecture Notes in Computer Science. Springer, 2014, pp. 818–833.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2016, pp. 1135–1144.

[18] 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.

[19] K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *ICLR (Workshop Poster)*. 2014.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015.

[21] A. Shrikumar, P. Greenside, and A. Kundaje. “Learning Important Features through Propagating Activation Differences”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia, 2017, pp. 3145–3153.

[22] M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 3319–3328.

[23] 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”. In: *PLoS ONE* 10.7 (July 2015), e0130140.

[24] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. “Striving for Simplicity: The All Convolutional Net”. In: *ICLR (workshop track)*. 2015.

[25] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller. “Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition”. In: *Pattern Recognition* 65 (2017), pp. 211–222.

[26] J. Kauffmann, M. Esders, L. Ruff, G. Montavon, W. Samek, and K.-R. Müller. *From Clustering to Cluster Explanations via Neural Networks*. 2021.

[27] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K. R. Müller. “Explainable Deep One-Class Classification”. In: *International Conference on Learning Representations*. 2021.

[28] G. Pang, C. Ding, C. Shen, and A. v. d. Hengel. “Explainable Deep Few-shot Anomaly Detection with Deviation Networks”. In: *arXiv preprint arXiv:2108.00462* (2021).

[29] A. Rosenfeld, R. S. Zemel, and J. K. Tsotsos. "The Elephant in the Room". In: *CoRR* abs/1808.03305 (2018).

[30] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn". In: *Nature Communications* 10 (2019), p. 1096.

[31] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel. "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition". In: *International conference on machine learning*. PMLR. 2019, pp. 5231–5240.

[32] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. "Adversarial Examples Are Not Bugs, They Are Features". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[33] Y. Belinkov and Y. Bisk. "Synthetic and Natural Noise Both Break Neural Machine Translation". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[34] T. McCoy, E. Pavlick, and T. Linzen. "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3428–3448.

[35] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans. "Trial without Error: Towards Safe Reinforcement Learning via Human Intervention". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 2067–2069.

[36] W. R. Swartout and J. D. Moore. "Explanation in Second Generation Expert Systems". In: *Second Generation Expert Systems*. Berlin, Heidelberg: Springer-Verlag, 1993, pp. 543–585. ISBN: 0387561927.

[37] F. Doshi-Velez and B. Kim. "A Roadmap for a Rigorous Science of Interpretability". In: *CoRR* abs/1702.08608 (2017).

[38] T. Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (Feb. 2019), pp. 1–38. ISSN: 0004-3702.

[39] G. Montavon. "Gradient-Based Vs. Propagation-Based Explanations: An Axiomatic Comparison". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, 2019, pp. 253–265.

[40] G. Vilone and L. Longo. "Notions of explainability and evaluation approaches for explainable artificial intelligence". In: *Information Fusion* 76 (Dec. 2021), pp. 89–106. ISSN: 1566-2535.

[41] 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". In: *Proceedings of the IEEE* 109.3 (2021), pp. 247–278.

[42] A. B. Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115.

[43] W. J. E. Potts. "Generalized Additive Neural Networks". In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '99. San Diego, California, USA: Association for Computing Machinery, 1999, pp. 194–200. ISBN: 1581131437.

[44] S. M. Lundberg and S. Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. NIPS'17. 2017, pp. 4765–4774.

[45] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton. "Neural additive models: Interpretable machine learning with neural nets". In: *arXiv preprint arXiv:2004.13912* (2020).

[46] J. Lampinen and A. Vehtari. "Bayesian approach for neural networks—review and case studies". In: *Neural Networks* 14.3 (2001), pp. 257–274. ISSN: 0893-6080.

[47] D. M. Titterington. "Bayesian Methods for Neural Networks and Related Models". In: *Statistical Science* 19.1 (2004), pp. 128–139.

[48] S. Saralajew, L. Holdijk, M. Rees, E. Asan, and T. Villmann. "Classification-by-Components: Probabilistic Modeling of Reasoning over a Set of Components". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[49] D. Nebel, M. Kaden, A. Villmann, and T. Villmann. "Types of (Dis-)Similarities and Adaptive Mixtures Thereof for Improved Classification Learning". In: *Neurocomput.* 268.C (Dec. 2017), pp. 42–54. ISSN: 0925-2312.

[50] J. Snell, K. Swersky, and R. Zemel. "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[51] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin. "This Looks like That: Deep Learning for Interpretable Image Recognition". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[52] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. "A Structured Self-attentive Sentence Embedding". In: *CoRR* abs/1703.03130 (2017).

[53] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. "What Does BERT Look at? An Analysis of BERT's Attention". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286.

[54] M. T. Ribeiro, S. Singh, and C. Guestrin. "Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance". In: *CoRR* abs/1611.05817 (2016).

[55] Z. Zhou, G. Hooker, and F. Wang. "S-LIME: Stabilized-LIME for Model Explanation". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 2429–2438. ISBN: 9781450383325.

[56] L. S. Shapley. "A Value for n-Person Games". In: *Contributions to the Theory of Games II*. Ed. by H. W. Kuhn and A. W. Tucker. Princeton: Princeton University Press, 1953, pp. 307–317.

[57] S. S. Fatima, M. Wooldridge, and N. R. Jennings. "A Linear Approximation Method for the Shapley Value". In: *Artif. Intell.* 172.14 (Sept. 2008), pp. 1673–1699. ISSN: 0004-3702.

[58] M. Ancona, C. Öztireli, and M. H. Gross. "Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 272–281.

[59] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. "How to Explain Individual Classification Decisions". In: *J. Mach. Learn. Res.* 11 (2010), pp. 1803–1831.

[60] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. "On the number of linear regions of deep neural networks". English (US). In: *Advances in Neural Information Processing Systems* 4.January (2014), pp. 2924–2932. ISSN: 1049-5258.

[61] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". English (US). In: 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014. Jan. 2014.

[62] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. "Top-down Neural Attention by Excitation Backprop". In: *European Conference on Computer Vision(ECCV)*. 2016.

[63] 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*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 342–350.

[64] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. "Layer-Wise Relevance Propagation: An Overview". In: *Explainable AI*. Vol. 11700. Lecture Notes in Computer Science. Springer, 2019, pp. 193–209.

[65] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. "Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Trans. Neural Netw. Learning Syst.* 28.11 (2017), pp. 2660–2673.

[66] H. Zhang, J. Chen, H. Xue, and Q. Zhang. "Towards a Unified Evaluation of Explanation Methods without Ground Truth". In: *CoRR* abs/1911.09017 (2019).

[67] A. Rosenfeld. "Better Metrics for Evaluating Explainable Artificial Intelligence". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. Virtual Event, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 45–50. ISBN: 9781450383073.

[68] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger. "Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics". In: *Electronics* 10.5 (2021). ISSN: 2079-9292.

[69] A. Hedström, L. Weber, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M.-C. Höhne. "Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations". In: (2022).

[70] M. Riveiro and S. Thill. ""That's (not) the output I expected!" On the role of end user expectations in creating explanations of AI systems". In: *Artificial Intelligence* 298 (2021), p. 103507. ISSN: 0004-3702.

[71] P. Croskerry. "A universal model of diagnostic reasoning." In: *Academic medicine : journal of the Association of American Medical Colleges* 84 8 (2009), pp. 1022–8.

[72] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim. "Designing Theory-Driven User-Centric Explainable AI". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–15. ISBN: 9781450359702.

[73] S. Zhang, X. Zhang, W. Zhang, and A. Søgaard. "Sociolectal Analysis of Pretrained Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4581–4588.

[74] C. Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. ISSN: 25225839.

[75] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace. "ERASER: A Benchmark to Evaluate Rationalized NLP Models". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4443–4458.

[76] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. "A Diagnostic Study of Explainability Techniques for Text Classification". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by B. Webber, T. Cohn, Y. He, and Y. Liu. Association for Computational Linguistics, 2020, pp. 3256–3274.

[77] A. F. Markus, J. A. Kors, and P. R. Rijnbeek. "The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies". In: *Journal of Biomedical Informatics* 113 (2021), p. 103655. ISSN: 1532-0464.

[78] S. Wachter, B. Mittelstadt, and C. Russell. "Counterfactual explanations without opening the black box: automated decisions and the GDPR". In: *Harvard Journal of Law and Technology* 31.2 (2018), pp. 841–887.

[79] B. Kim, R. Khanna, and O. O. Koyejo. "Examples are not enough, learn to criticize! Criticism for Interpretability". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016.

[80] A.-P. Nguyen and M. R. Mart'inez. "On quantitative aspects of model interpretability". In: *ArXiv* abs/2007.07584 (2020).

[81] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (2019), 93:1–93:42.

[82] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. "Interpretable & Explorable Approximations of Black Box Models". In: *CoRR* abs/1707.01154 (2017).

[83] B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy. *A Simple and Effective Model-Based Variable Importance Measure*. 2018.

[84] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *ICLR (Poster)*. OpenReview.net, 2018.

[85] C. Meister, S. Lazov, I. Augenstein, and R. Cotterell. "Is Sparse Attention more Interpretable?" In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 122–129.

[86] R. Henderson, D.-A. Clevert, and F. Montanari. "Improving Molecular Graph Neural Network Explainability with Orthonormalization and Induced Sparsity". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 4203–4213.

[87] M. T. Ribeiro, S. Singh, and C. Guestrin. "Anchors: High-Precision Model-Agnostic Explanations". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018).

[88] D. S. Watson, L. Gultchin, A. Taly, and L. Floridi. "Local Explanations Via Necessity and Sufficiency: Unifying Theory and Practice". In: *Minds and Machines* 32.1 (2022), pp. 185–218.

[89] D. Alvarez-Melis and T. S. Jaakkola. "On the Robustness of Interpretability Methods". In: (2018). cite arxiv:1806.08049Comment: presented at 2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden.

[90] C. Yeh, C. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar. "On the (In)fidelity and Sensitivity of Explanations". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett. 2019, pp. 10965–10976.

[91] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[92] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin. "Towards Best Practice in Explaining Neural Network Decisions with LRP". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–7.

[93] R. Hoffman, S. Mueller, G. Klein, and J. Litman. "Metrics for Explainable AI: Challenges and Prospects". In: (Dec. 2018).

[94] S. Mohseni, N. Zarei, and E. D. Ragan. "A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems". In: *ACM Trans. Interact. Intell. Syst.* 11.3–4 (Aug. 2021). ISSN: 2160-6455.

[95] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. "The role of trust in automation reliance". In: *International Journal of Human-Computer Studies* 58.6 (2003). Trust and Technology, pp. 697–718. ISSN: 1071-5819.

[96] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan. "Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–14. ISBN: 9781450367080.

[97] M. Harbers, J. Broekens, K. Bosch, and J.-J. Meyer. "Guidelines for Developing Explainable Cognitive Models". In: *Proceedings of ICCM 2010, Berlin, Germany*. Jan. 2010, pp. 85–90.

[98] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Wortman Vaughan, and H. Wallach. "Manipulating and Measuring Model Interpretability". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966.

[99] I. Lage, A. S. Ross, S. J. Gershman, B. Kim, and F. Doshi-Velez. "Human-in-the-Loop Interpretability Prior". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 10180–10189.

[100] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. "Principles of Explanatory Debugging to Personalize Interactive Machine Learning". In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. IUI '15. Atlanta, Georgia, USA: Association for Computing Machinery, 2015, pp. 126–137. ISBN: 9781450333061.

[101] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady. "explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 1064–1074.

[102] J. Adebayo, M. Muelly, I. Liccardi, and B. Kim. "Debugging Tests for Model Explanations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 700–712.

[103] J. Krause, A. Perer, and K. Ng. "Interacting with Predictions: Visual Inspection of Black-Box Machine Learning Models". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 5686–5697. ISBN: 9781450333627.

[104] A. Malhi, S. Knapic, and K. Främling. "Explainable Agents for Less Bias in Human-Agent Decision Making". In: *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Ed. by D. Calvaresi, A. Najjar, M. Winikoff, and K. Främling. Cham: Springer International Publishing, 2020, pp. 129–146. ISBN: 978-3-030-51924-7.

[105] Y. Ding, Y. Liu, H. Luan, and M. Sun. "Visualizing and Understanding Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1150–1159.

[106] W. Jin, X. Li, M. Fatehi Hassanabad, and G. Hamarneh. *Guidelines and evaluation for clinical explainable AI on medical image analysis*. Feb. 2022.

[107] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models". In: *Decision Support Systems* 51.1 (2011), pp. 141–154. ISSN: 0167-9236.

[108] P. Schmidt and F. Bießmann. "Quantifying Interpretability and Trust in Machine Learning Systems". In: *CoRR* abs/1901.08558 (2019).

[109] D. Nguyen. "Comparing Automatic and Human Evaluation of Local Explanations for Text Classification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1069–1078.

[110] H. Lakkaraju, S. H. Bach, and J. Leskovec. "Interpretable Decision Sets: A Joint Framework for Description and Prediction". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1675–1684. ISBN: 9781450342322.

[111] A. Holzinger, A. M. Carrington, and H. Müller. "Measuring the Quality of Explanations: The System Causability Scale (SCS)". In: *Kunstliche Intelligenz* 34 (2020), pp. 193–198.

[112] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. "e-SNLI: Natural Language Inference with Natural Language Explanations". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 9539–9549.

[113] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra. "Human Attention in Visual Question Answering: Do Humans and Deep Networks look at the same regions?" In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 932–937.

[114] S. Klerke, Y. Goldberg, and A. Søgaard. "Improving sentence compression by learning to predict gaze". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 1528–1533.

[115] M. Barrett, J. Bingel, N. Hollenstein, M. Rei, and A. Søgaard. "Sequence Classification with Human Attention". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 302–312.

[116] Y. Zhang and C. Zhang. "Using Human Attention to Extract Keyphrase from Microblog Post". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5867–5872.

[117] N. Hollenstein, F. Pirovano, C. Zhang, L. Jäger, and L. Beinborn. "Multilingual Language Models Predict Human Reading Behavior". In: *arXiv preprint arXiv:2104.05433* (2021).

[118] M. Barrett and A. Søgaard. “Using reading behavior to predict grammatical functions”. In: *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1–5.

[119] A. Mishra, K. Dey, and P. Bhattacharyya. “Learning Cognitive Features from Gaze Data for Sentiment and Sarcasm Classification using Convolutional Neural Network”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 377–387.

[120] J. Malmaud, R. Levy, and Y. Berzak. “Bridging Information-Seeking Human Gaze and Machine Reading Comprehension”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, Nov. 2020, pp. 142–152.

[121] T. Kulesza, S. Stumpf, M. M. Burnett, S. Yang, I. Kwan, and W.-K. Wong. “Too much, too little, or just right? Ways explanations impact end users’ mental models”. In: *2013 IEEE Symposium on Visual Languages and Human Centric Computing* (2013), pp. 3–10.

[122] A. Dhurandhar, V. S. Iyengar, R. Luss, and K. Shanmugam. “TIP: Typifying the Interpretability of Procedures”. In: *CoRR* abs/1706.02952 (2017).

[123] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilović, K. Natesan Ramamurthy, and K. R. Varshney. “TED: Teaching AI to Explain Its Decisions”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES ’19. Honolulu, HI, USA: Association for Computing Machinery, 2019, pp. 123–129. ISBN: 9781450363242.

[124] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. “The (un)reliability of saliency methods”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Lecture Notes in Computer Science. Springer, 2019, pp. 267–280.

[125] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck. “Evaluating Explanation Methods for Deep Learning in Security”. In: Sept. 2020, pp. 158–174.

[126] J. Aldrich. “Correlations Genuine and Spurious in Pearson and Yule”. In: *Statistical Science* 10.4 (1995), pp. 364–376.

[127] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322.

[128] 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”. In: *Information Fusion* 77 (2022), pp. 261–295. ISSN: 1566-2535.

[129] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. “On Causal and Anticausal Learning”. In: *Proceedings of the 29th International Conference on Machine Learning*. New York, NY, USA: Omnipress, 2012, pp. 1255–1262.

[130] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. “Shortcut Learning in Deep Neural Networks”. In: *Nature Machine Intelligence* 2 (Nov. 2020), pp. 665–673.

[131] J. Jo and Y. Bengio. “Measuring the tendency of CNNs to Learn Surface Statistical Regularities”. In: *CoRR* abs/1711.11561 (2017).

[132] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” In: *ICLR*. OpenReview.net, 2019.

[133] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim. "Learning Not to Learn: Training Deep Neural Networks With Biased Data". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[134] S. Teso and K. Kersting. "Explanatory Interactive Machine Learning". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '19. Honolulu, HI, USA: Association for Computing Machinery, 2019, pp. 239–245. ISBN: 9781450363242.

[135] M. Nauta, R. Walsh, A. Dubowski, and C. Seifert. "Uncovering and Correcting Shortcut Learning in Machine Learning Models for Skin Cancer Diagnosis". In: *Diagnostics* 12.1 (2022), p. 40.

[136] A. S. Ross, M. C. Hughes, and F. Doshi-Velez. "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 2662–2670.

[137] L. Rieger, C. Singh, W. Murdoch, and B. Yu. "Interpretations are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 8116–8126.

[138] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H. Luigs, A. Mahlein, and K. Kersting. "Making deep neural networks right for the right scientific reasons by interacting with their explanations". In: *Nat. Mach. Intell.* 2.8 (2020), pp. 476–486.

[139] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani, and G. Montavon. "Building and Interpreting Deep Similarity Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2022), pp. 1149–1161.

[140] H. El-Hajj et al. "An Ever-Expanding Humanities Knowledge Graph: The Sphaera Corpus at the Intersection of Humanities, Data Management, and Machine Learning". In: *Datenbank Spektrum* (May 2022).

[141] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. "Engineering support vector machine kernels that recognize translation initiation sites". In: *Bioinformatics* 16.9 (2000), pp. 799–807.

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

[143] A. Nierman and H. V. Jagadish. "Evaluating Structural Similarity in XML Documents". In: *WebDB*. 2002, pp. 61–66.

[144] E. Pampalk, A. Flexer, and G. Widmer. "Improvements of Audio-Based Music Similarity and Genre Classification". In: *ISMIR*. 2005, pp. 628–633.

[145] P. Willett, J. M. Barnard, and G. M. Downs. "Chemical Similarity Searching". In: *Journal of Chemical Information and Computer Sciences* 38.6 (1998), pp. 983–996.

[146] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

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

[148] 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.

[149] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data Clustering: A Review". In: *ACM Comput. Surv.* 31.3 (1999), pp. 264–323.

[150] J. Shi and J. Malik. “Normalized Cuts and Image Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8 (2000), pp. 888–905.

[151] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. “Estimating the Support of a High-Dimensional Distribution”. In: *Neural Computation* 13.7 (2001), pp. 1443–1471.

[152] B. Schölkopf, A. J. Smola, and K.-R. Müller. “Nonlinear Component Analysis as a Kernel Eigenvalue Problem”. In: *Neural Computation* 10.5 (1998), pp. 1299–1319.

[153] 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.

[154] L. van der Maaten and G. E. Hinton. “Visualizing non-metric similarities in multiple maps”. In: *Machine Learning* 87.1 (2012), pp. 33–55.

[155] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. “Multiple kernel learning, conic duality, and the SMO algorithm”. In: *ICML*. Vol. 69. ACM International Conference Proceeding Series. ACM, 2004.

[156] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. “Large Scale Multiple Kernel Learning”. In: *J. Mach. Learn. Res.* 7 (2006), pp. 1531–1565.

[157] K. Q. Weinberger and L. K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *J. Mach. Learn. Res.* 10 (2009), pp. 207–244.

[158] J. Bergstra and Y. Bengio. “Random Search for Hyper-Parameter Optimization”. In: *J. Mach. Learn. Res.* 13 (2012), pp. 281–305.

[159] 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.

[160] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *ICLR*. 2015.

[161] 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”. In: *Journal of Historical Network Research* 3 (2019), pp. 50–91.

[162] S. T. Roweis and L. K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326.

[163] R. R. Coifman and S. Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30.

[164] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. “SmoothGrad: removing noise by adding noise”. In: *CoRR* abs/1706.03825 (2017).

[165] A.-p. Nguyen and M. R. Martínez. “It’s FLAN time! Summing feature-wise latent representations for interpretability”. In: *ArXiv* abs/2106.10086 (2021).

[166] J. Kauffmann, K.-R. Müller, and G. Montavon. “Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models”. In: *Pattern Recognition* 101 (2020), p. 107198.

[167] B. Micenková, R. T. Ng, X. Dang, and I. Assent. “Explaining Outliers by Subspace Separability”. In: *ICDM*. IEEE Computer Society, 2013, pp. 518–527.

[168] 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*. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 2048–2057.

[169] M. Zheng, S. Karanam, T. Chen, R. J. Radke, and Z. Wu. “Learning Similarity Attention”. In: *CoRR* abs/1911.07381 (2019).

[170] M. Tsang, D. Cheng, and Y. Liu. “Detecting Statistical Interactions from Neural Network Weights”. In: *ICLR (Poster)*. OpenReview.net, 2018.

[171] T. Cui, P. Marttinen, and S. Kaski. “Recovering Pairwise Interactions Using Neural Networks”. In: *CoRR* abs/1901.08361 (2019).

[172] S. Leupold. “Second-Order Taylor decomposition for Explaining Spatial Transformation of Images”. MA thesis. Technische Universität Berlin, 2017.

[173] M. Simon, E. Rodner, T. Darrell, and J. Denzler. “The Whole Is More Than Its Parts? From Explicit to Implicit Pose Normalization”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 42.3 (2020), pp. 749–763.

[174] 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.

[175] S. Lerman, C. Venuto, H. Kautz, and C. Xu. “Explaining Local, Global, and Higher-Order Interactions in Deep Learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 1224–1233.

[176] J. D. Janizek, P. Sturmfels, and S.-I. Lee. “Explaining Explanations: Axiomatic Feature Interactions for Deep Networks”. In: *Journal of Machine Learning Research* 22.104 (2021), pp. 1–54.

[177] C. Watkins. “Dynamic Alignment Kernels”. In: *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 39–50.

[178] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller. “A New Discriminative Kernel from Probabilistic Models”. In: *Neural Computation* 14.10 (2002), pp. 2397–2414.

[179] T. Gärtner. “A survey of kernels for structured data”. In: *SIGKDD Explorations* 5.1 (2003), pp. 49–58.

[180] 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.

[181] 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.

[182] 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.

[183] E. Hoffer and N. Ailon. “Deep Metric Learning Using Triplet Network”. In: *SIMBAD*. Vol. 9370. Lecture Notes in Computer Science. Springer, 2015, pp. 84–92.

[184] B. Seguin, C. Striolo, I. diLenardo, and F. Kaplan. “Visual Link Retrieval in a Database of Paintings”. In: *ECCV Workshops (1)*. Vol. 9913. Lecture Notes in Computer Science. Springer, 2016, pp. 753–767.

[185] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. “Neural Collaborative Filtering”. In: *WWW*. ACM, 2017, pp. 173–182.

[186] R. Memisevic and G. E. Hinton. “Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines”. In: *Neural Computation* 22.6 (2010), pp. 1473–1492.

[187] 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.

[188] A.-K. Dombrowski, C. J. Anders, K.-R. Müller, and P. Kessel. “Towards robust explanations for deep neural networks”. In: *Pattern Recognition* 121 (2022), p. 108194. ISSN: 0031-3203.

[189] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks". In: *AISTATS*. Vol. 15. JMLR Proceedings. JMLR.org, 2011, pp. 315–323.

[190] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade (2nd ed.)* Vol. 7700. Lecture Notes in Computer Science. Springer, 2012, pp. 9–48.

[191] 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.

[192] 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.

[193] R. Caruana. "Multitask Learning". In: *Machine Learning* 28.1 (1997), pp. 41–75.

[194] 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.

[195] 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.

[196] 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". In: *Medical Image Analysis* 42 (2017), pp. 60–88.

[197] Y. Gao and K. M. Mosalam. "Deep Transfer Learning for Image-Based Structural Damage Recognition". In: *Comp.-Aided Civil and Infrastruct. Engineering* 33.9 (2018), pp. 748–768.

[198] L. Lenc and P. Král. "Unconstrained Facial Images: Database for Face Recognition Under Real-World Conditions". In: *MICAI (2)*. Vol. 9414. Lecture Notes in Computer Science. Springer, 2015, pp. 349–361.

[199] 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*. Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.

[200] 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.

[201] 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.

[202] F. Kräutli and M. Valleriani. "CorpusTracer: A CIDOC database for tracing knowledge networks". In: *DSH* 33.2 (2018), pp. 336–346.

[203] S. Lang and B. Ommer. "Attesting similarity: Supporting the organization and study of art image collections with computer vision". In: *Digital Scholarship in the Humanities* 33.4 (Apr. 2018), pp. 845–856. ISSN: 2055-7671.

[204] J. Bruna and S. Mallat. "Invariant Scattering Convolution Networks". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (2013), pp. 1872–1886.

[205] 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". In: *Science advances* 3.5 (2017), e1603015.

[206] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko. "Towards exact molecular dynamics simulations with machine-learned force fields". In: *Nature Communications* 9.1 (Sept. 2018).

[207] F. Anselmi, L. Rosasco, and T. Poggio. “On invariance and selectivity in representation learning”. In: *Information and Inference* 5.2 (May 2016), pp. 134–158.

[208] 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.

[209] M. D. Rodriguez, J. Ahmed, and M. Shah. “Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), pp. 1–8.

[210] K. Soomro and A. Zamir. “Action Recognition in Realistic Sports Videos”. In: *Advances in Computer Vision and Pattern Recognition* 71 (Jan. 2014), pp. 181–208.

[211] J. Qiu, Q.-H. Wu, G. Ding, Y. Xu, and S. Feng. “A survey of machine learning for big data processing”. In: *EURASIP Journal on Advances in Signal Processing* 2016 (2016), pp. 1–16.

[212] T. S. Kuhn. *The structure of scientific revolutions*. Vol. 111. Chicago University of Chicago Press, 1970.

[213] D. Wang, C. Song, and A.-L. Barabási. “Quantifying long-term scientific impact”. In: *Science* 342.6154 (2013), pp. 127–132.

[214] M. Zamani, A. Tejedor, M. Vogl, F. Kräutli, M. Valleriani, and H. Kantz. “Evolution and Transformation of Early Modern Cosmological Knowledge: A Network Study”. In: *Scientific Reports* 10 (2020), p. 19822.

[215] C. D. Jackson. “Educational Reforms of Wittenberg and Their Faithfulness to Martin Luther’s Thought”. In: *Christian Education Journal: Research on Educational Ministry* 10 (2013), pp. 71–87.

[216] I. C. Hennen. “Printers, Booksellers, and Bookbinders in Wittenberg in the Sixteenth Century: Real Estate, Vicinity, Political, and Cultural Activities”. In: *Publishing Sacrobosco’s De sphaera in Early Modern Europe. Modes of Material and Scientific Exchange*. Ed. by M. Valleriani and A. Ottone. Springer, 2022, pp. 99–154.

[217] I. Pantin. “Borrowers and Innovators in the Printing History of Sacrobosco: The Case of the “in-octavo” Tradition”. In: *De sphaera of Johannes de Sacrobosco in the Early Modern Period: The Authors of the Commentaries*. Ed. by M. Valleriani. Cham: Springer Nature, 2020, pp. 265–312.

[218] S. Limbach. “Scholars, Printers, and the Sphere: New Evidence for the Challenging Production of Academic Books in Wittenberg, 1531–1550”. In: *Publishing Sacrobosco’s «De sphaera» in Early Modern Europe. Modes of Material and Scientific Exchange*. Ed. by M. Valleriani and A. Ottone. Cham: Springer Nature, 2022, pp. 155–194.

[219] W. B. Johnson and J. Lindenstrauss. “Extensions of Lipschitz mappings into a Hilbert space 26”. In: *Contemporary mathematics* 26 (1984), p. 28.

[220] P. Frankl and H. Maehara. “The Johnson-Lindenstrauss lemma and the sphericity of some graphs”. In: *Journal of Combinatorial Theory, Series B* 44.3 (1988), pp. 355–362.

[221] E. Bingham and H. Mannila. “Random Projection in Dimensionality Reduction: Applications to Image and Text Data”. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’01. San Francisco, California: Association for Computing Machinery, 2001, pp. 245–250. ISBN: 158113391X.

[222] D. Fradkin and D. Madigan. “Experiments with Random Projections for Machine Learning”. In: KDD ’03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 517–522. ISBN: 1581137370.

[223] H. Caesar, J. Uijlings, and V. Ferrari. “COCO-Stuff: Thing and stuff classes in context”. In: *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE. 2018.

[224] M. Alber. “Software and Application Patterns for Explanation Methods”. In: Sept. 2019, pp. 399–433. ISBN: 978-3-030-28953-9.

[225] S. Beery, G. Van Horn, and P. Perona. "Recognition in Terra Incognita". In: *Proceedings of the European Conference on Computer Vision*. Munich, Germany, 2018.

[226] 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". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 10.1109/TPAMI.2021.3115452.

[227] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The Graph Neural Network Model". In: *IEEE Trans. Neural Networks* 20.1 (2009), pp. 61–80.

[228] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81.

[229] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020), pp. 1–21.

[230] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. "SchNet– A deep learning architecture for molecules and materials". In: *The Journal of Chemical Physics* 148.24 (2018), p. 241722.

[231] M. Zitnik, M. Agrawal, and J. Leskovec. "Modeling polypharmacy side effects with graph convolutional networks". In: *Bioinform.* 34.13 (2018), pp. i457–i466.

[232] D. Marcheggiani and I. Titov. "Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling". In: *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1506–1515.

[233] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an. "Graph Convolutional Encoders for Syntax-aware Neural Machine Translation". In: *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1957–1967.

[234] D. Beck, G. Haffari, and T. Cohn. "Graph-to-Sequence Learning using Gated Graph Neural Networks". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*. Association for Computational Linguistics, 2018, pp. 273–283.

[235] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. "Dynamic Graph CNN for Learning on Point Clouds". In: *Association for Computing Machinery Trans. Graph.* 38.5 (2019), 146:1–146:12.

[236] T. Cui, P. Marttinen, and S. Kaski. "Learning Global Pairwise Interactions with Bayesian Neural Networks". In: *24th European Conference on Artificial Intelligence*. Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 1087–1094.

[237] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann. "Explainability Methods for Graph Convolutional Neural Networks". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10772–10781.

[238] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji. "On Explainability of Graph Neural Networks via Subgraph Explorations". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 12241–12252.

[239] C. Ji, R. Wang, and H. Wu. "Perturb More, Trap More: Understanding Behaviors of Graph Neural Networks". In: *CoRR* abs/2004.09808 (2020).

[240] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang. "Graphlime: Local interpretable model explanations for graph neural networks". In: *arXiv preprint arXiv:2001.06216* (2020).

[241] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. "Parameterized Explainer for Graph Neural Network". In: *Advances in Neural Information Processing Systems* 33 (2020).

[242] Z. Ying, J. You, M. Zitnik, and J. Leskovec. "GNNExplainer: Generating Explanations for Graph Neural Networks". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 9240–9251.

[243] R. Schwarzenberg, M. Hübner, D. Harbecke, C. Alt, and L. Hennig. "Layerwise Relevance Visualization in Convolutional Text Graph Classifiers". In: *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing, TextGraphs@EMNLP*. Association for Computational Linguistics, 2019, pp. 58–62.

[244] H. Yuan, J. Tang, X. Hu, and S. Ji. "XGNN: Towards Model-Level Explanations of Graph Neural Networks". In: *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Ed. by R. Gupta, Y. Liu, J. Tang, and B. A. Prakash. ACM, 2020, pp. 430–438.

[245] W. Lin, H. Lan, and B. Li. "Generative Causal Explanations for Graph Neural Networks". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 6666–6679.

[246] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T.-S. Chua. "Reinforced Causal Explainer for Graph Neural Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–1.

[247] M. Vu and M. Thai. "PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks". In: *34th Conference on Neural Information Processing Systems* (2020).

[248] M. S. Schlichtkrull, N. D. Cao, and I. Titov. "Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking". In: *ICLR*. OpenReview.net, 2021.

[249] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Process. Mag.* 34.4 (2017), pp. 18–42.

[250] T. N. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations*. OpenReview.net, 2017.

[251] K. Schütt, P. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". In: *Neural Information Processing Systems*. 2017, pp. 991–1001.

[252] B. Yu, H. Yin, and Z. Zhu. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018, pp. 3634–3640.

[253] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. 2017, pp. 1263–1272.

[254] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. "How Powerful are Graph Neural Networks?" In: *7th International Conference on Learning Representations*. OpenReview.net, 2019.

[255] R. Albert and A.-L. Barabási. "Statistical mechanics of complex networks". In: *Reviews of Modern Physics* 74.1 (Jan. 2002), pp. 47–97.

[256] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistic, 2013, pp. 1631–1642.

[257] H. Yuan, H. Yu, S. Gui, and S. Ji. "Explainability in Graph Neural Networks: A Taxonomic Survey". In: (2020).

[258] A. Mordvintsev, C. Olah, and M. Tyka. *Inceptionism: Going Deeper into Neural Networks*. 2015.

[259] P. Xiong. "Efficient Higher-Order Subgraph Attribution via Message Passing". MA thesis. Technische Universität Berlin, 2022.

[260] O. Eberle, S. Brandl, J. Pilot, and A. Søgaard. "Do Transformer Models Show Similar Attention Patterns to Task-Specific Human Gaze?" In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4295–4309.

[261] A. Ali, T. Schnake, O. Eberle, G. Montavon, K. Müller, and L. Wolf. "XAI for Transformers: Better Explanations through Conservative Propagation". In: *Proceedings of the 39th International Conference on Machine Learning*. ICML'22. Baltimore, Maryland, USA: JMLR.org, 2022.

[262] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* (2019).

[263] A. Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021*. 2021.

[264] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. "Do Transformers Really Perform Badly for Graph Representation?" In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021.

[265] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim. "Graph Transformer Networks". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*. 2019, pp. 11960–11970.

[266] D. Narayanan et al. "Efficient large-scale language model training on GPU clusters using megatron-LM". In: *SC '21: The International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021.

[267] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi. "BEHRT: Transformer for Electronic Health Records". In: 10.1 (2020). ISSN: 2045-2322.

[268] M. A. Aleisa, N. Beloff, and M. White. "AIRM: a new AI recruiting model for the Saudi Arabia labor market". In: *Intelligent Systems Conference (IntelliSys) 2021*. Ed. by K. Arai. Vol. 3: 296. Lecture Notes in Networks and Systems. Cham: Springer, Sept. 2021, pp. 105–124.

[269] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 4356–4364.

[270] H. Gonen and Y. Goldberg. "Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Association for Computational Linguistics, 2019, pp. 609–614.

[271] S. Kiritchenko and S. Mohammad. "Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems". In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. 2018, pp. 43–53.

[272] V. Prabhakaran, B. Hutchinson, and M. Mitchell. "Perturbation Sensitivity Analysis to Detect Unintended Model Biases". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*. 2019, pp. 5739–5744.

[273] M. De-Arteaga, A. Romanov, H. Wallach, J. Chayes, C. Borgs, A. Chouldechova, S. Geyik, K. Kenthapadi, and A. T. Kalai. "Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 120–128.

[274] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020.

[275] N. Ousidhoum, X. Zhao, T. Fang, Y. Song, and D.-Y. Yeung. "Probing Toxic Content in Large Pre-Trained Language Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 4262–4274.

[276] Y. Zhou, M. Kaneko, and D. Bollegala. "Sense Embeddings are also Biased – Evaluating Social Biases in Static and Contextualised Sense Embeddings". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1924–1935.

[277] P. Czarnowska, Y. Vyas, and K. Shah. "Quantifying Social Biases in NLP: A Generalization and Empirical Comparison of Extrinsic Fairness Metrics". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1249–1267.

[278] D. Bahdanau, K. Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015.

[279] S. Jain and B. C. Wallace. "Attention is not Explanation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3543–3556.

[280] S. Serrano and N. A. Smith. "Is Attention Interpretable?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2931–2951.

[281] S. Abnar and W. H. Zuidema. "Quantifying Attention Flow in Transformers". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Ed. by D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault. 2020, pp. 4190–4197.

[282] H. Chefer, S. Gur, and L. Wolf. "Generic Attention-Model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 397–406.

[283] H. Chefer, S. Gur, and L. Wolf. "Transformer interpretability beyond attention visualization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 782–791.

[284] E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh. "AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. 2019, pp. 7–12.

[285] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 5797–5808.

[286] Z. Wu and D. C. Ong. "On Explaining Your Explanations of BERT: An Empirical Study with Sequence Classification". In: *CoRR* abs/2101.00196 (2021).

[287] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. "Pathologies of Neural Models Make Interpretations Difficult". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3719–3728.

[288] O. Pandit and Y. Hou. "Probing for Bridging Inference in Transformer Language Models". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4153–4163.

[289] M. Alleman, J. Mamou, M. A Del Rio, H. Tang, Y. Kim, and S. Chung. "Syntactic Perturbations Reveal Representational Correlates of Hierarchical Phrase Structure in Pretrained Language Models". In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 263–276.

[290] L. S. Shapley. "A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 1953, pp. 307–318.

[291] E. Strumbelj and I. Kononenko. "An Efficient Explanation of Individual Classifications using Game Theory". In: *J. Mach. Learn. Res.* 11 (2010), pp. 1–18.

[292] R. Hesse, S. Schaub-Meyer, and S. Roth. "Fast Axiomatic Attribution for Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021.

[293] S. Srinivas and F. Fleuret. "Full-Gradient Representation for Neural Network Visualization". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*. 2019, pp. 4126–4135.

[294] M. Ancona, E. Ceolini, C. Öztireli, and M. H. Gross. "Gradient-Based Attribution Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Lecture Notes in Computer Science. Springer, 2019, pp. 169–191.

[295] L. Arras, J. Arjona-Medina, M. Widrich, G. Montavon, M. Gillhofer, K.-R. Müller, S. Hochreiter, and W. Samek. "Explaining and Interpreting LSTMs". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller. Vol. 11700. Lecture Notes in Computer Science. Springer, Cham, 2019, pp. 211–238.

[296] J. Sun, S. Lapuschkin, W. Samek, and A. Binder. "Explain and improve: LRP-inference fine-tuning for image captioning models". In: *Information Fusion* 77 (2022), pp. 233–246. ISSN: 1566-2535.

[297] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 142–150.

[298] W. Xiong, J. Wu, H. Wang, V. Kulkarni, M. Yu, X. Guo, S. Chang, and W. Y. Wang. "TweetQA: A Social Media Focused Question Answering Dataset". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 5020–5031.

[299] E. Chapuis, P. Colombo, M. Manica, M. Labeau, and C. Clavel. "Hierarchical Pre-training for Sequence Labelling in Spoken Dialog". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 2636–2648.

[300] M. Denil, A. Demiraj, and N. de Freitas. "Extraction of Salient Sentences from Labelled Documents". In: *ArXiv* abs/1412.6815 (2014).

[301] N. Hollenstein and L. Beinborn. “Relative Importance in Sentence Processing”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 141–150.

[302] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *ArXiv* abs/1910.01108 (2019).

[303] S. Klerke and B. Plank. “At a Glance: The Impact of Gaze Aggregation Views on Syntactic Tagging”. In: *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 51–61.

[304] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[305] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67.

[306] Y. Lecun and Y. Bengio. “Convolutional Networks for Images, Speech and Time Series”. In: *The Handbook of Brain Theory and Neural Networks*. Ed. by M. A. Arbib. The MIT Press, 1995, pp. 255–258.

[307] E. Reichle, K. Rayner, and A. Pollatsek. “The E-Z reader model of eye-movement control in reading: comparisons to other models”. English. In: *The Behavioral and brain sciences* 26.4 (Aug. 2003), pp. 445–476. ISSN: 0140-525X.

[308] N. Hollenstein, J. Rotsztejn, M. Troendle, A. Pedroni, C. Zhang, and N. Langer. “ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading”. In: *Scientific data* 5.1 (2018), pp. 1–13.

[309] Y. Kim. “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[310] J. Pennington, R. Socher, and C. D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543.

[311] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. “Explaining Predictions of Non-Linear Classifiers in NLP”. In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1–7.

[312] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. “"What is relevant in a text document?": An interpretable machine learning approach”. In: *PLOS ONE* 12.8 (Aug. 2017), pp. 1–23.

[313] E. D. Reichle, A. Pollatsek, D. L. Fisher, and K. Rayner. “Toward a model of eye movement control in reading.” In: *Psychological review* 105.1 (1998), p. 125.

[314] A. Kilgarriff. “BNC database and word frequency lists”. In: (1995). *Accessed: 07/2020*.

[315] K. Rayner and S. A. Duffy. “Lexical complexity and fixation times in reading: effects of word frequency, verb complexity, and lexical ambiguity”. In: *Memory & cognition* 14.3 (May 1986), pp. 191–201. ISSN: 0090-502X.

[316] R. Zhong, S. Shao, and K. R. McKeown. “Fine-grained Sentiment Analysis with Faithful Attention”. In: *CoRR* abs/1908.06870 (2019).

[317] J. Lu, D. Batra, D. Parikh, and S. Lee. "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[318] H. Tan and M. Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5100–5111.

[319] P. Schramowski, C. Turan, N. Andersen, C. A. Rothkopf, and K. Kersting. "Large pre-trained language models contain human-like biases of what is right and wrong to do". In: *Nature Machine Intelligence* 4 (Mar. 2022), pp. 258–268.

[320] I. Garrido-Muñoz, A. Montejo-Ráez, F. Martínez-Santiago, and L. A. Ureña-López. "A Survey on Bias in Deep NLP". In: *Applied Sciences* 11.7 (2021). ISSN: 2076-3417.

[321] J. Borowski, C. M. Funke, K. Stosio, W. Brendel, T. Wallis, and M. Bethge. "The notorious difficulty of comparing human and machine perception". In: *2019 Conference on Cognitive Computational Neuroscience*. 2019, pp. 2019–1295.

[322] M. Barrett and A. Søgaard. "Reading behavior predicts syntactic categories". In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 345–349.

[323] A. Mishra, D. Kanojia, S. Nagar, K. Dey, and P. Bhattacharyya. "Leveraging Cognitive Features for Sentiment Analysis". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 156–166.

[324] Y. Zhao and S. Bethard. "How does BERT's attention change when you fine-tune? An analysis methodology and a case study in negation scope". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4729–4747.

[325] A. Merchant, E. Rahimtoroghi, E. Pavlick, and I. Tenney. "What Happens To BERT Embeddings During Fine-tuning?" In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, Nov. 2020, pp. 33–44.

[326] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. "Multimodal deep learning". In: *ICML*. 2011.

[327] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer. "Towards multi-modal causability with Graph Neural Networks enabling information fusion for explainable AI". In: *Information Fusion* 71 (2021), pp. 28–37. ISSN: 1566-2535.

[328] K. Bayoudh, R. Knani, F. Hamdaoui, and M. Abdellatif. "A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets". In: *The Visual Computer* (June 2021).

[329] A. Karpathy, A. Joulin, and L. F. Fei-Fei. "Deep Fragment Embeddings for Bidirectional Image Sentence Mapping". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger. Vol. 27. Curran Associates, Inc., 2014.

[330] V. Nagisetty, L. Graves, J. Scott, and V. Ganesh. *xAI-GAN: Enhancing Generative Adversarial Networks via Explainable AI Systems*. 2020.

[331] A. L. Smith, D. M. Asta, and C. A. Calder. "The Geometry of Continuous Latent Space Models for Network Data". In: *Statistical Science* 34.3 (2019), pp. 428–453.

[332] C. Wu, H. Zhang, J. Chen, Z. Gao, P. Zhang, K. Muhammad, and J. Del Ser. "Vessel-GAN: Angiographic reconstructions from myocardial CT perfusion with explainable generative adversarial networks". In: *Future Generation Computer Systems* 130 (2022), pp. 128–139. ISSN: 0167-739X.

[333] F. Ye and A. G. Bors. "Learning joint latent representations based on information maximization". In: *Information Sciences* 567 (2021), pp. 216–236. ISSN: 0020-0255.

[334] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton. "TeMP: Temporal Message Passing for Temporal Knowledge Graph Completion". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5730–5746.

[335] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein. "Temporal Graph Networks for Deep Learning on Dynamic Graphs". In: *ICML 2020 Workshop on Graph Representation Learning*. 2020.

[336] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua. "MMGCN: Multi-Modal Graph Convolution Network for Personalized Recommendation of Micro-Video". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: Association for Computing Machinery, 2019, pp. 1437–1445. ISBN: 9781450368896.

[337] S. Mai, H. Hu, and S. Xing. "Modality to Modality Translation: An Adversarial Representation Learning and Graph Fusion Network for Multimodal Fusion". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 164–172.

[338] R. Saqur and K. Narasimhan. "Multimodal Graph Networks for Compositional Generalization in Visual Question Answering". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 3070–3081.

[339] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. W. Battaglia, and T. Lillicrap. "A simple neural network module for relational reasoning". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett. 2017, pp. 4967–4976.

[340] P. Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv* (2018).

[341] V. F. Zambaldi et al. "Deep reinforcement learning with relational inductive biases". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[342] A. Santoro, R. Faulkner, D. Raposo, J. Rae, M. Chrzanowski, T. Weber, D. Wierstra, O. Vinyals, R. Pascanu, and T. Lillicrap. "Relational Recurrent Neural Networks". In: NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 7310–7321.

[343] M. Shanahan, K. Nikiforou, A. Creswell, C. Kaplanis, D. Barrett, and M. Garnelo. "An Explicitly Relational Neural Network Architecture". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 8593–8603.

[344] A. Creswell, M. Shanahan, and I. Higgins. *Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning*. 2022.

[345] L. Wang, A. Adiga, J. Chen, A. Sadilek, S. Venkatramanan, and M. Marathe. "CausalGNN: Causal-Based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting". In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. AAAI Press, 2022.

[346] N. R. Ke, S. Chiappa, J. Wang, J. Bornschein, T. Weber, A. Goyal, M. Botvinic, M. Mozer, and D. J. Rezende. *Learning to Induce Causal Structure*. 2022.

[347] C. Y. Zheng, C. I. Baker, F. Pereira, and M. N. Hebart. *Revealing interpretable object representations from human behavior*. 2019.

[348] B. D. Roads and B. C. Love. *Enriching ImageNet with Human Similarity Judgments and Psychological Embeddings*. 2021, pp. 3546–3556.

[349] M. Kümmerer, T. S. A. Wallis, and M. Bethge. "Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics". In: *Computer Vision – ECCV 2018*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 798–814.

[350] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. "Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning". In: *Physical review letters* 108 (Jan. 2012), p. 058301.

[351] 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". In: *Chemical Reviews* 121.16 (2021), pp. 10142–10186.

[352] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl. "Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding". In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium* (2019), pp. 5901–5904.

[353] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic. "Detecting and quantifying causal associations in large nonlinear time series datasets". In: *Science Advances* 5.11 (2019), eaau4996.

[354] B. A. Toms, E. A. Barnes, and I. Ebert-Uphoff. "Physically Interpretable Neural Networks for the Geosciences: Applications to Earth System Variability". In: *Journal of Advances in Modeling Earth Systems* 12.9 (2020). e2019MS002002 10.1029/2019MS002002, e2019MS002002.

[355] C. J. Shallue and A. M. Vanderburg. "Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90". In: *arXiv: Earth and Planetary Astrophysics* (2017).

[356] H. Valizadegan et al. "Exominer: A Highly Accurate And Explainable Deep Learning Classifier That Validates 200+ New Exoplanets". In: *Bulletin of the AAS* 53.6 (June 18, 2021).

[357] F. Klauschen et al. "Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning". In: *Seminars in Cancer Biology* 52 (July 2018).

[358] A. Binder et al. "Morphological and molecular breast cancer profiling through explainable machine learning". In: *Nature Machine Intelligence* 3 (Apr. 2021), pp. 1–12.

[359] U. Güçlü and M. A. J. van Gerven. "Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream". In: *Journal of Neuroscience* 35.27 (2015), pp. 10005–10014. ISSN: 0270-6474.

[360] S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. "Deep convolutional models improve predictions of macaque V1 responses to natural images". In: *PLoS Computational Biology* (2019).

[361] W. J. Neumann, R. S. Turner, B. Blankertz, T. Mitchell, A. A. Kühn, and R. M. Richardson. "Toward Electrophysiology-Based Intelligent Adaptive Deep Brain Stimulation for Movement Disorders". In: *Neurotherapeutics : the journal of the American Society for Experimental NeuroTherapeutics* 16 (1 Jan. 2019), pp. 105–118. ISSN: 1878-7479.

[362] M. W. Mathis and A. Mathis. "Deep learning tools for the measurement of animal behavior in neuroscience". In: *Current Opinion in Neurobiology* 60 (2020). Neurobiology of Behavior, pp. 1–11. ISSN: 0959-4388.

[363] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. "Explainable Machine Learning for Scientific Insights and Discoveries". In: *IEEE Access* 8 (2020), pp. 42200–42216.

[364] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[365] M. Weiler and G. Cesa. "General E(2)-Equivariant Steerable CNNs". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[366] E. Simoncelli and W. Freeman. "The steerable pyramid: a flexible architecture for multi-scale derivative computation". In: *Proceedings., International Conference on Image Processing*. Vol. 3. 1995, 444–447 vol.3.

[367] M. Honnibal and I. Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". 2017.

[368] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[369] M. Petrochuk. *PyTorch-NLP: Rapid Prototyping with PyTorch Natural Language Processing (NLP) Tools*. `https://github.com/PetrochukM/PyTorch-NLP`. 2018.

[370] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (Poster)*. 2015.

[371] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020.

[372] R. Kneser and H. Ney. "Improved backing-off for m-gram language modeling". In: *1995 international conference on acoustics, speech, and signal processing*. Vol. 1. IEEE. 1995, pp. 181–184.

[373] A. Stolcke. "SRILM – An extensible language modeling toolkit". In: *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*. 2002, pp. 901–904.

[374] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. *One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling*. Tech. rep. Google, 2013.

A

Supplementary Details

A.1 Details for Rendering BiLRP Explanations

Below we provide the parameters selected for visualizing BiLRP explanations as presented in Section 3.6–3.7.1 and Section 3.8. A description of the visualization procedure is given in Section 3.4.

Table A.1: Parameters used on each dataset for rendering BiLRP explanations.

Dataset	input size	pool	l	h	p
Pascal VOC 2007	128×128	8×8	0.25	13	2
Faces (UFI & LFW)	64×64	4×4	0.3	60	1
UCF Sport	128×128	8×8	0.25	20	1
Sphaera (illustrations)	96×96	6×6	0.25	15	2
Sphaera (tables)	140×140	20×20	0.01	4	2

A.2 Details for Use Case - Digital Humanities

The following provides additional details on methods and implementation for our use case in modeling historical numerical tables in Section 3.8.

A.2.1 Digit Recognition Model Architecture

The recognition architecture consists of two main encoder modules, namely, (i) the 'encoder' and (ii) the 'convolutional_encoder' that together form the 7-layer neural network. The digit recognition model was implemented in the `PyTorch`

```
NeuralOCR(
  (encoder): Sequential(
    (0): R2Conv([8-Rotations], kernel_size=3, stride=1, padding=1, bias=False)
    (1): ReLU(inplace=True)
    (2): R2Conv([8-Rotations], kernel_size=3, stride=1, padding=1, bias=False)
    (3): ReLU(inplace=True])
    (4): R2Conv([8-Rotations], kernel_size=5, stride=1, padding=2, bias=False)
    (5): ReLU(inplace=True)
    (6): R2Conv([8-Rotations], kernel_size=5, stride=1, padding=2, bias=False)
    (7): GroupPooling([8-Rotations])
  )
  (convolutional_encoder): Sequential(
    (0): Conv2d(64, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), bias=False)
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): ReLU(inplace=True)
    (4): Conv2d(32, 10, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
```

Figure A.1: Atom recognition model.

`1.8.1` [364] framework. We build a convolutional neural network that consists of multiple layers of convolutional layers that consist of either standard or equivariant [365] convolutional layers. An earlier version of this network consists of standard convolution layers instead of equivariant convolutional layers [139]. We use the network for the historical insights and analyses as presented in Figure A.1. It consists of an initial 4-layer equivariant convolutional block with filter sizes {3×3, 3×3, 5×5, 5×5} and 8-rotational groups, which ensures that low-level feature detectors are learned to be invariant to translations and rotations on the $\mathbb{R}^2$-plane. Thus, features required to recognize digits (such as lines, arches or circles) generalize over spatial input transformations resulting in increased data efficiency. We apply ReLU layers between all convolutional layers except at the final layer in a block. A final pooling layer selects the maximally activating map from the equivariant group. Subsequently, these features are learned to be combined into digits detectors using a stack of three standard convolution layers of kernel sizes {5×5, 1×1, 1×1} which output 10 activation maps $\{\boldsymbol{a}_j(\boldsymbol{x})\}_{j=1}^{10}$ for the digits 0–9. For each digit j the network produces a Gaussian blob positioned at the digit location in map $\{\boldsymbol{a}_j(\boldsymbol{x})\}$. We subtract a small bias term $b = 0.1$ before the ReLU layer to attenuate background activity in the final layer. To model variations in scan orientation and size, we identify the page scaling factor s and rotation θ, for which the single-digit activation maps are maximally activated (sum of activations).

A.2.1.1 Modeling Invariances

Local Scale and Rotation Invariance We further robustify the learned representations against style and scale heterogeneity by augmenting the training data patches using the following transformations: (i) We apply rotations of $\pm 10°$, (ii) translations of the patch by (0.025×img_width/height in x- and y-direction, (iii) proportional scaling of the full patch by a factor in the range $(0.8 - 1.2\times)$ using

bilinear interpolation and (iv) shearing transformation of ($\pm 5°$) along both spatial directions. A random value from the specified range is sampled and added to the training dataset for each possible augmentation. We sample as many augmented data points as there are annotated patches.

Global Scale Invariance Global scale differences in the corpus can be caused by either (i) different sizes of the used woodblock during printing, i.e. larger or smaller typesetting, but also (ii) from the resolution differences that can result in several orders of pixel height and width spans in the data. In order to model both sources in parallel, we chose to implement a multi-scale feature pyramid approach similar to the framework of steerable pyramids [366]. This has the advantage of parameter-efficacy since no additional trainable parameters are introduced and model transparency since the multi-scale approach is based on the linear decomposition of the image at different scales from which the most activating feature scale is chosen and thus, remains fully explainable.
Using bilinear interpolation, we re-scale the image to a reference height or width of 1200 pixels at reference scale $s = 1.0$ (depending on portrait or landscape orientation). Resulting input images are collected for every scale $s \in S = \{s_1, ..., 1.0, ..., s_K\}$ and fed through the atom-recognition network. The scale $s^* = \max_{s \in S} \sum_j \boldsymbol{a}_j(\boldsymbol{x}; s)$ maximizing the spatially pooled activity over all features j is then chosen for the further processing.

Global Rotation Invariance Similarly, there exist differences in page orientation that can be caused by either (i) the printing process itself when the printer considered a table or illustration to be better readable in landscape orientation, but also (ii) from the scan process. We model both of these as in the previous section concerning scale by including page input rotations $\theta \in \Theta = \{-90, 0, 90\}°$ and select the rotation that maximizes activity $\theta^* = \max_{\theta \in \Theta} \sum_j \boldsymbol{a}_j(\boldsymbol{x}; \theta)$.

A.2.1.2 Activity Peak Detection

The bigram network outputs can now be used to directly spatially pool activity. While this is a simple and viable approach that does produce meaningful similarity it is not always clear how the pooled activity corresponds to one bigram on a page. A pooled activity of 100 can correspond to two very prototypical bigrams that activate the network very strongly or four weakly activated less prototypical examples. Besides thresholding before pooling, we propose to use a standard peak detection process to convert the raw activation maps into bigram count maps. We start from a set of 100 bigram maps $\boldsymbol{a}_{jk}$ with $jk = \{00, ..., 99\}$ which are added to 10 maps for isolated digits $\hat{\boldsymbol{a}}_i$ with $i = \{\_0\_, ..., \_9\_\}$ resulting in $\bar{\mathbf{a}} = (\boldsymbol{a}_i, \boldsymbol{a}_{jk})$. Since the max-pooling used for the bigrams reduces the activity levels in comparison to the isolated digit maps, we introduce a scaling parameter α to the latter $\boldsymbol{a}_i = \hat{\boldsymbol{a}}_i / \alpha$. Next, we subtract a bias term $\beta \cdot \max_{(x,y)} \bar{\mathbf{a}}_{(x,y)}$ computed as the product of relative scaling parameter β and the maximum pixel value in all maps. The resulting maps are rectified, which,

similarly to the processing applied to the single-digit activation maps, reduces weak background activity. Then, for each of the 110 feature maps, we extract the feature regions that occur at all non-zero locations and compute all peaks using the center of activity mass. We determine the linkage matrix using the distances between centers and perform a hierarchical clustering to group close-by activated pixels into groups of pixels that belong to a bigram. To limit the size of clustered regions, we define a maximum distance parameter d. We select optimal parameters using histogram Pearson correlation scores on the training patches and set $\alpha = 3$, $\beta = 0.12$ and $d = 15$.

A.3 Details for Experiments on GNN models

We next give additional details on the design, training, and implementation of the GNN models used in Section 4.4, and applied to the synthetic BA-growth dataset for predicting graph types, the Stanford Sentiment Treebank dataset for sentiment classification, and the VGG-16 convolutional neural network for object classification.

A.3.1 GNNs trained on Synthetic Data

We use synthetic datasets to train different GNN architectures, which allows us to generate arbitrary large datasets to train well-performing and robust GNN models.

Data The synthetic dataset BA-growth consists of graphs of 20 nodes, generated from two different classes. The first class consists of Barabási-Albert graphs [255] with a growth parameter 1. For each sample, we start with a graph of two connected nodes, and at each step, we add an additional node and connect it to a node $\mathcal{N}$ from the current graph $\mathcal{G}$ randomly from the distribution

$$p(\mathcal{N}) = \frac{\text{degree}(\mathcal{N})}{\sum_{\mathcal{N}' \in \mathcal{G}} \text{degree}(\mathcal{N}')}.$$

The second class has a higher growth parameter than 1, where every fifth node gets connected to two nodes from the current graph instead of one. For the second class, we use the following inverse preferential attachment model that selects nodes without replacement and with probability

$$p(\mathcal{N}) = \frac{\text{degree}(\mathcal{N})^{-1}}{\sum_{\mathcal{N}' \in \mathcal{G}} \text{degree}(\mathcal{N}')^{-1}}.$$

GNN Models We train the BA-growth dataset on GCN, GIN and the spectral network models using binary cross-entropy loss. Each model is built from two interaction layers with dimensionality of hidden nodes of 128 neurons for the GCN, and 32 neurons for the GIN and the spectral network. We assume no additional information about the nodes at the input layer and set the initial state $\boldsymbol{H}_0$ to

a vector filled with ones. For the spectral network, we use the power expansion $\mathbf{\Lambda} = [\widetilde{\boldsymbol{A}}^0, \frac{1}{2}\widetilde{\boldsymbol{A}}^1, \frac{1}{4}\widetilde{\boldsymbol{A}}^2]$ as the input in each layer. We require biases to be non-positive, which avoids the the presence of factors that can not be attributed in meaningful ways. To achieve this, we reparametrize the biases using $b = -0.5\log(1+\exp(-2b_0))$ and optimize b_0 instead of b.

After creating train and test sets that each consist of 100,000 randomly selected text samples, we use SGD optimization and set the learning rate at each iteration to $\eta = 0.001/(1 + \text{iteration}/1000)$. We observe an average accuracy of 95 %, 96 % and 97 % for GCN, GIN and the spectral network on the test set.

A.3.2 Sentiment Analysis on SST

Data We use a GCN model to predict the sentiment of the Stanford Sentiment Treebank (SST) dataset [256]. We filter all samples of neutral sentiment and create a two-class dataset by merging 'positive' and 'very positive' as well as 'negative' and 'very negative' samples, respectively. This results in 6920 train and 1821 test samples. We extract the dependency tree for each samples using the spaCy package `en_core_web_sm` [367].

GNN Model The initial state $\boldsymbol{H}_0$ is built from text example $(\mathcal{G}, l)$, with graph $\mathcal{G} = (\boldsymbol{A}, N)$, where $\boldsymbol{A}$ is the adjacency matrix of the dependency tree, N are the words of the text, and l is the sentiment label of the graph. To find an initial representation of the sentence, given by N, we take vector representations of a pretrained FastText [368] word embedding h_w of dimension 300 provided by `torchnlp` [369], a randomly initialized word embedding h_v, an embedding for the part-of-speech h_p and an embedding for the stemmed words h_l. We set the network initialization to be $\boldsymbol{H}_0 = [h_v, h_w, h_p, h_l]$, where we keep h_w fixed during training and h_v, h_p and h_l learnable. In the forward propagation, we apply a feed-forward neural network (FFN) with ReLU activation simultaneously on each embedded word in $\boldsymbol{H}_0$, to obtain a hidden representation of dimension d_h of each word. Resulting embeddings are further processed using T interaction blocks (of same type as a GCNs [250]) with connectivity matrix $\mathbf{\Lambda} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$, $\tilde{\boldsymbol{D}} = \text{diag}((\sum_i \tilde{\boldsymbol{A}}_{ij})_j)$ and $\tilde{\boldsymbol{A}}$ is the adjacency matrix of the undirected input graph with added self-connections. Another FFN with ReLU activation is applied and followed by the final readout step for which we use a global average pooling in node direction to compute a vector of dimension d_h followed by a linear layer to map representations onto the target dimension. A softmax layer is used subsequently to output class probabilities.

We train the network for 50 epochs with the cross-entropy loss between the output and label l, and the Adam optimizer [370] with learning rate $\eta = 2e-4$. We use hidden dimension $d_h = 10$, the number of layers $T = 3$ and in $\boldsymbol{H}_0$ we used $h_v = 70$, $h_p = 30$ and $h_l = 50$ respectively. We obtain a test set accuracy of approximately 77%.

A.3.3 VGG-16

We use the pretrained VGG-16 neural network (without batch normalization) provided by the PyTorch library and do not change its structure or pretrained parameters.

A.4 Details for Experiments on Transformer models

Below we provide experimental details for our experiments on evaluating Transformer explanations in Section 5.5.

A.4.1 Sequence Classification

For the NLP experiments, we consider binary sentiment classification on the SST-2 and IMDB datasets which contain 11,844 and 50,000 movie reviews, respectively. In addition, we use the TweetEval Dataset for tweet classification on sentiment (59,899), hate detection (12,970) and emotion recognition (5,052). Furthermore, the SILICONE Dataset is used for emotion detection (Semaine 13,708) and utterance sentiment analysis (Meld-S 5,627). For SST-2 and IMDB sentiment classification, the embeddings module and the tokenizer are initialized from pre-trained BERT-Transformers (textattack/bert-base-uncased-{sst-2/imdb}). For training, we use batch sizes of $bs = 32$ and optimize the model parameters using the AdamW optimizer with a learning rate of $lr = 2e - 5$ for a maximal number of $T = 20$ epochs or until early stopping for decreasing validation performance is reached. We follow the same settings for the Twitter-X, Meld-S and Semaine datasets, except that we initialize the embedding model and the tokenizer from a commonly used HuggingFace pre-trained BERT-Transformer (bert-base-uncased)[1].

A.4.2 Details for Use Case B: Task-Solving in Humans and Transformers

In the following, we provide details on models, optimization and attribution methods used in our use case B in Section 5.7.

CNN The CNN models use 300-dimensional pre-trained GloVe_840B [310] embeddings. Input sentences are tokenized using the SpaCy tokenizer [371]. We use 150 convolutional filters of filter sizes $s = [3, 4, 5]$ with ReLU activation, followed by a max-pooling-layer and apply dropout of $p = 0.5$ of the linear classification layer during training. For training, we use a batch size of $bs = 50$ and train all model parameters using the Adam optimizer with a learning rate of $lr = 1e - 4$ for a maximum number of $T = 20$ epochs. For all model training, we apply early stopping to avoid overfitting during training and stop optimization as soon as the validation loss begins to increase. To compute LRP explanations, we use the LRP-γ propagation rule with $\gamma = 0.$ for the linear readout layers [64]. We take absolute

[1] `https://huggingface.co/bert-base-uncased`

values over resulting relevance scores since we find they correlate best with human attention in comparison to raw and rectified processing. For propagation through the max-pooling layer we apply the winner-take-all principle and for convolutional layers we use the LRP-γ redistribution rule and select $\gamma = 0.5$ after a search over $\gamma = [0., 0.25, 0.5, 0.75, 1.0]$ resulting in largest correlations to human attention.

Self-Attention Model The multi-head self-attention model again uses 300-dimensional pre-trained GloVe_840B embeddings and is tokenized via SpaCy. The architecture consists of $k = 3$ self-attention heads for the SR task and $k = 8$ for REL. The resulting sentence representation is then fed into a linear classification readout layer with $\gamma = 0.$, which we also use for the relevance propagation to input embeddings. During optimization we use $lr = 1e - 4$, $bs = 50$ and $T = 50$.

Transformer Models We use standard BERT-base-uncased architectures and tokenizers as provided by the huggingface library [7]. For BERT-base fine-tuning we use $lr = 1e - 5$ for REL and $lr = 1e - 6$ for SR, $bs = 32$ and $T = 50$ for both tasks. For RoBERTa and T5 we use the RoBERTa-base and T5-base checkpoints and respective tokenizers.

E-Z Reader We use version 10.2 of the E-Z Reader with default parameters and 1000 repetitions. Cloze scores, i.e. word predictability scores, were therefore computed using a 5-gram Kneser-Ney language model [372] as provided by the SRI Language Modeling Toolkit [373] and trained on the 1 billion token dataset [374]. The resulting perplexity on the held-out test set was $ppl = 81.9$. Then, word-based total fixation times are computed from the E-Z Readers trace files and averaged over all subjects.

After fine-tuned all neural network models, we report the following performance over five runs in Table A.2.

	Acc (SR)	F1 (SR)	Acc (REL)	F1 (REL)
self-attention	69.0 ± 0.2	64.5 ± 2.2	67.5 ± 1.3	55.5 ± 2.0
CNN	71.3 ± 0.2	69.8 ± 1.7	74.0 ± 1.9	68.7 ± 4.8
BERT-base	76.0 ± 0.1	67.0 ± 3.0	78.3 ± 1.5	72.7 ± 3.3

Table A.2: Accuracy and F1 scores after fine-tuning on the respective task dataset over five runs: sentiment reading on SST (SR) and relation extraction on Wikipedia (REL). Samples that overlap with the ZuCo dataset were filtered out.