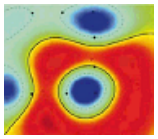
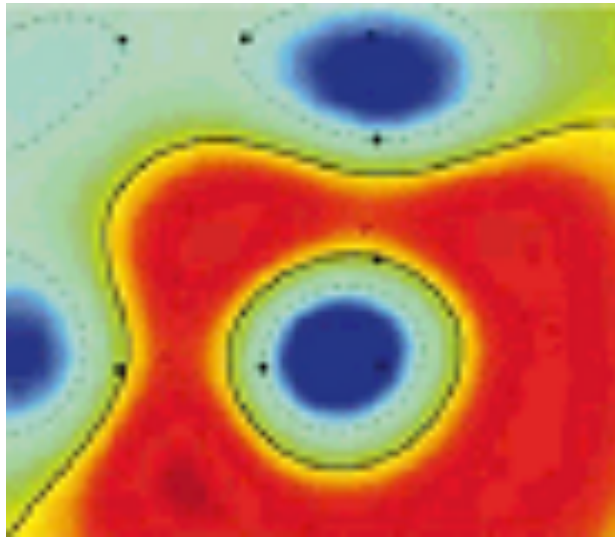


SoSe 2024

**Deep Learning 2**

Technische Universität Berlin

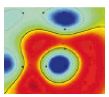


Lecture 5

# Advanced Explainable AI

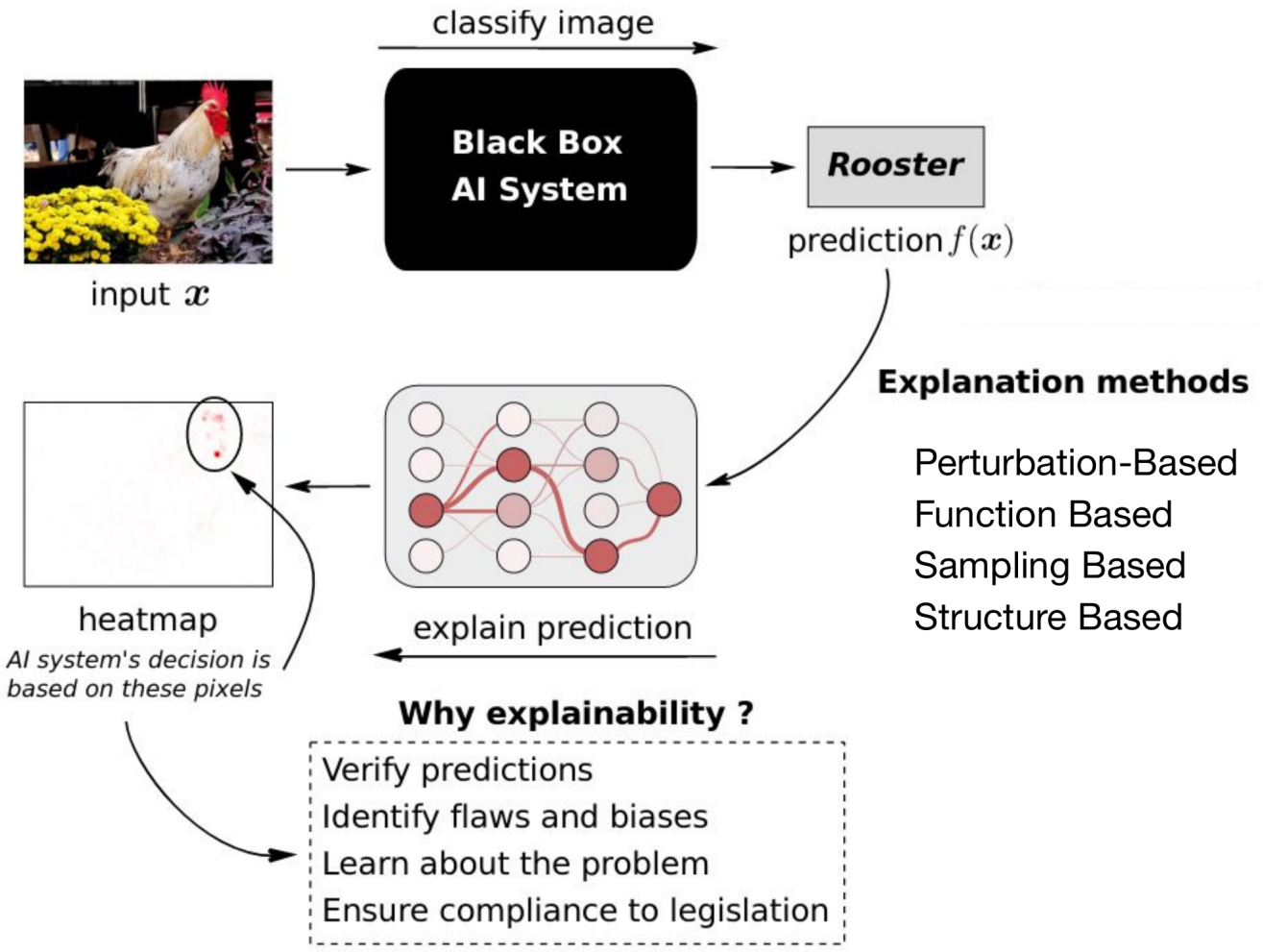
# Outline

- Recap
  - XAI
  - LRP
- Motivation
  - Beyond supervised settings
  - Beyond input heatmaps
  - Increasingly complex model structures - Advanced XAI
- Introduction Advanced XAI
  - Self-explainable models, natural XAI
  - concept-based explanations (neural network centric)
  - Model improvements with XAI
  - Beyond first-order explanations
  - Need to consider structure for complex models
- Advanced XAI examples
  - Similarity - BiLRP
  - Graphs - GNN-LRP
  - Anomaly
  - Clustering



# Recap: The standard scenario

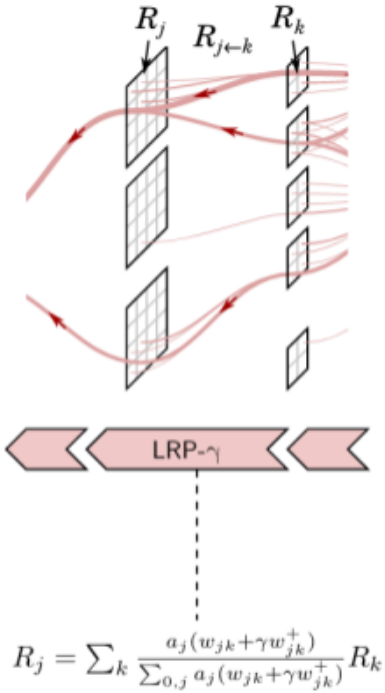
Explaining supervised classification settings, e.g. image classification:




Samek et al. (2017)

# Recap: Layer-wise Relevance Propagation (LRP)

LRP is a theoretically well-founded a widely used method to make deep neural networks explainable using propagation-based gradient information. Gradient Input is a special case of LRP ($\gamma = 0$).




**Example:** LRP-$\gamma$

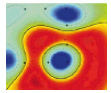
$$R_j = \sum_k \frac{a_j(w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j(w_{jk} + \gamma w_{jk}^+)} R_k$$

- $a_j(w_{jk} + \gamma w_{jk}^+)$: Contribution of neuron $a_j$ to the activation $a_k$.
- $R_k$ 'Relevance' of neuron $k$ available for redistribution.
- $\sum_{0,j} a_j(w_{jk} + \gamma w_{jk}^+)$ Normalization term that implements conservation.
- $\sum_k$: Pool all 'relevance' received by neuron $j$ from the layer above.

Samek Montavon (ECML/PKDD 2020)

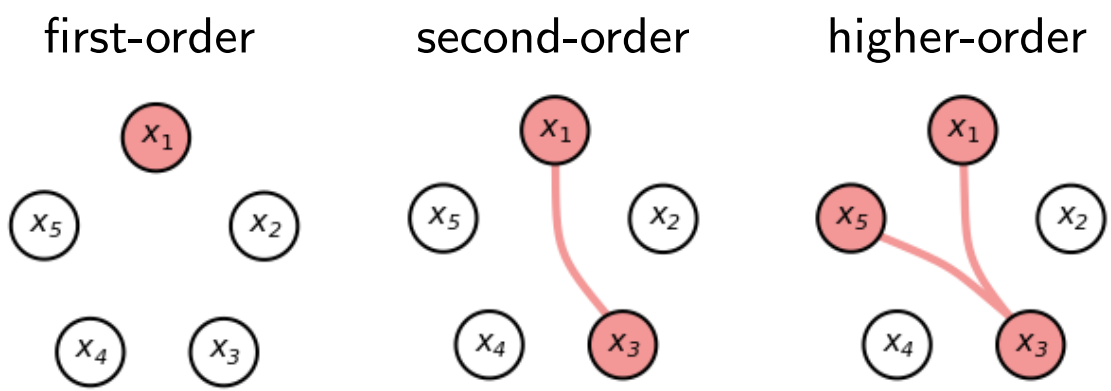
# Motivation

- **Beyond supervised settings**: unsupervised models, e.g. clustering or similarity models
- **Beyond standard model structure and heat-mapping**: GNNs, bilinear models (similarity models) $\rightarrow$ explanations using higher-order feature interactions
- **Neuralization**: clustering, anomaly detection, regression models
- **Utilizing explanations**: dataset-wide analyses $\rightarrow$ model improvement, insights, debugging



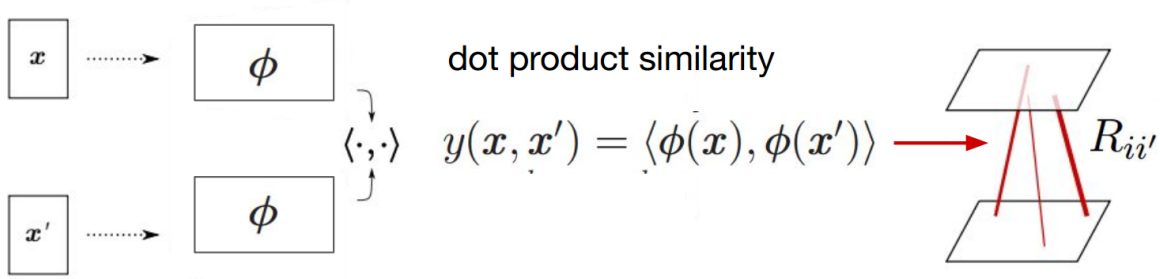
# Motivation: Beyond first-order explanations

- First-order explanations provide a natural way to describe model predictions via a decomposition onto input feature contributions, e.g. an activated feature in standard neural network classification models
- But, other scenarios utilize the conjunction of multiple features to compute a prediction, e.g.:
  - presence of two activated features in similarity models (second-order explanations)
  - presence of several activated features in GNNs (higher-order explanations)



# Second-order explanations: Explaining similarity models [6]

**Model:**




**Taylor Expansion:**

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

Further assume:

- piecewise linear and positively homogeneous function $\phi$
- well-chosen root points $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')$

**Hessian $\times$ Product:**

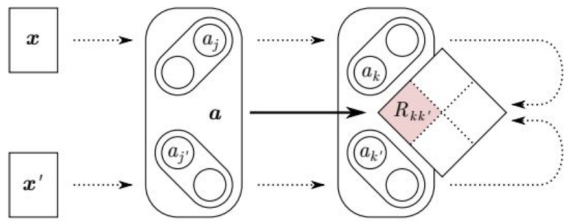
$$
y(\boldsymbol{x}, \boldsymbol{x}') = \sum_{ii'} [\nabla^2 y(\boldsymbol{x}, \boldsymbol{x}')]_{ii'} \, x_i \, x'_{i'}
$$

# 'Hessian x Product' explanations for deeper models

- This formulation can be seen as a second-order variant of 'Gradient x Input' explanations used to explain standard neural networks
- Using automatic differentiation software the Hessian can be conveniently computed
- **But**, similar to findings in first-order explanations the direct use of model gradients in deeper architectures suffers from
  - shattered gradients, i.e. the local gradient variations become stronger with increasing model depth [13, 4]
  - difficulty to robustly select root points, danger of adversarials that do not meaningfully represent the data [14, 7]

# Explaining deep similarity models more robustly

**Model:**




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

**Deep Taylor Decomposition:**

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

Further assume, e.g. for Linear/ReLU networks:

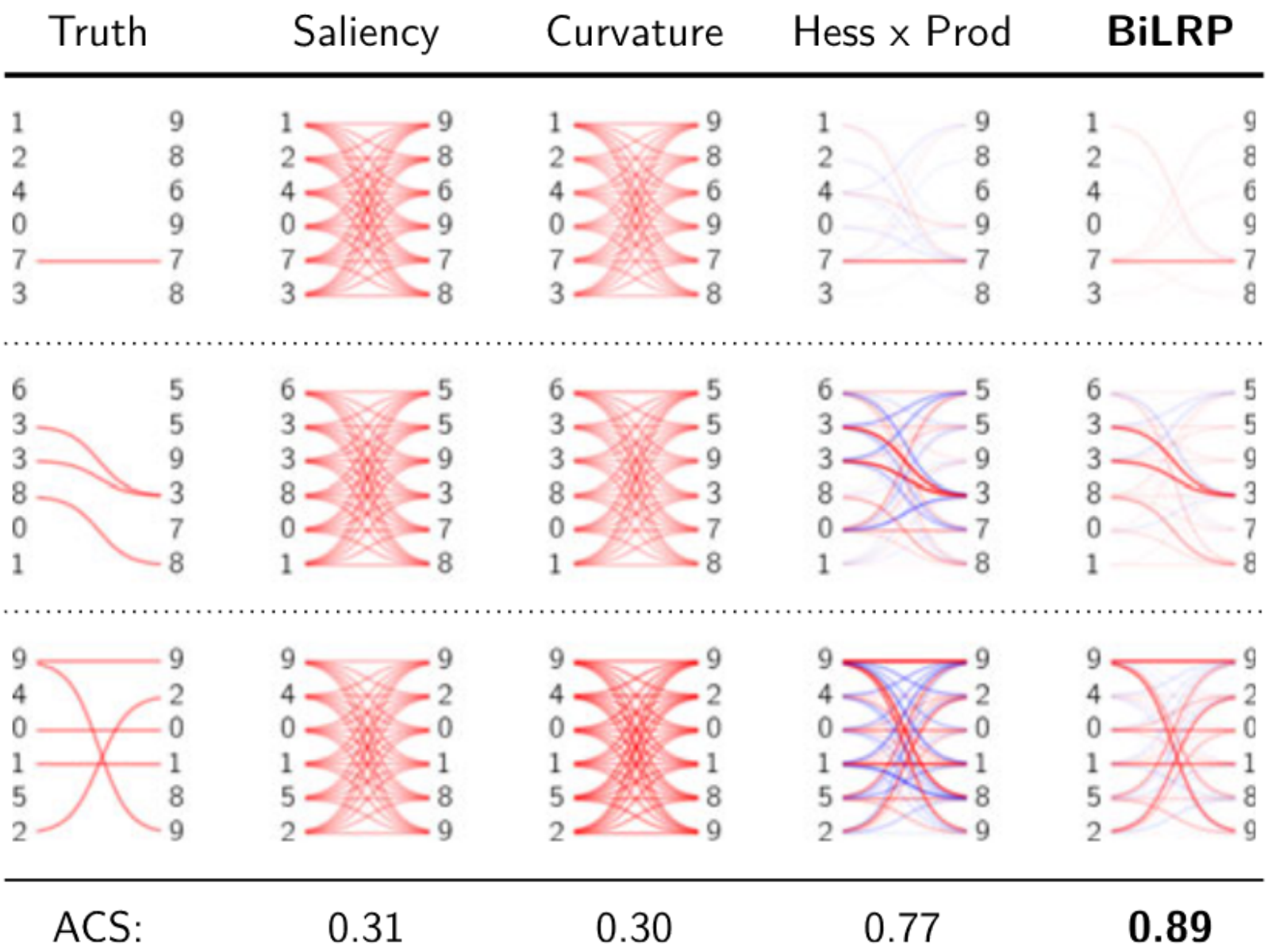
- Relevance model $\widehat{R}_{kk'}(\boldsymbol{a}) = a_k a_{k'} c_{kk'}$
- well-chosen root points $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{x}}')$

**BiLRP - propagation rule**

$$R_{jj'} = \sum_{kk'} \frac{a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})}{\sum_{jj'} a_j a_{j'} \rho(w_{jk}) \rho(w_{j'k'})} R_{kk'} \quad \text{with} \quad \rho(w_{jk}) = w_{jk} + \gamma w_{jk}^+$$

# Evaluation of second-order

**Setup** Toy task



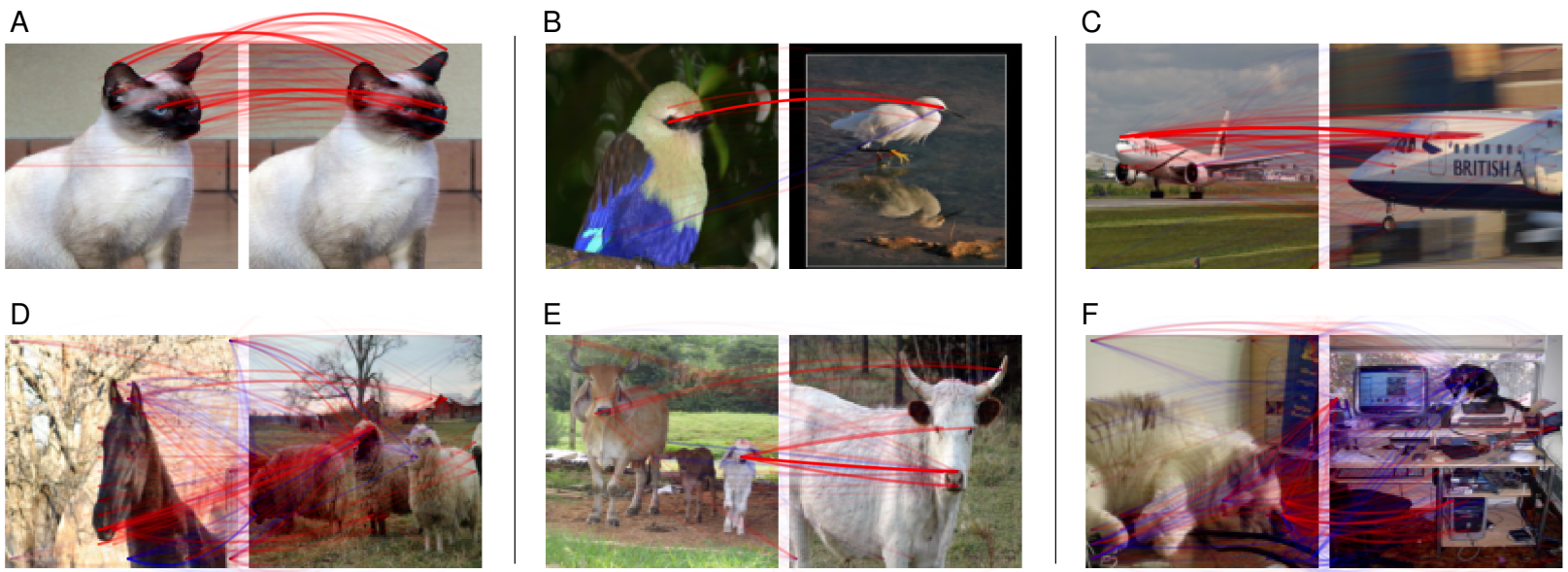
# Explaining similarity in deep convolutional models [16]

**Setup** Similarity between representations from VGG-16 pre-trained model for image classification. Define the following similarity model

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

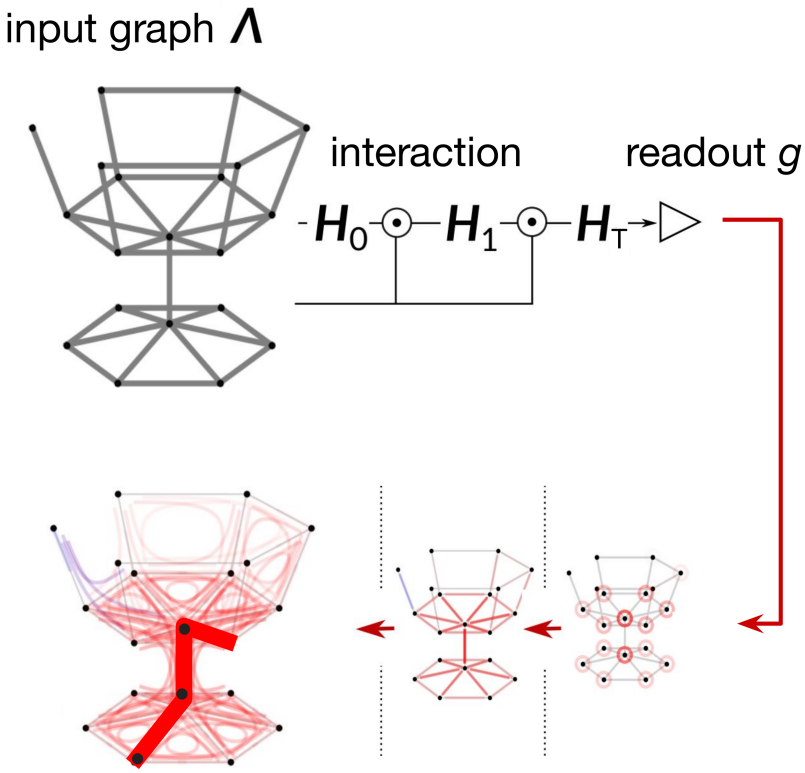
and use BiLRP to extract explanations.

**Results**



# Explaining GNNs using higher-order explanations [16]

In GNNs the input graph is highly entangled with the model. Several layers of interaction blocks that consist of 'aggregate' and 'combine' steps are use to compute the GNN prediction. The adjacency matrix $\mathbf{\Lambda}$ is used repeatedly as an input in the computation blocks:




$R_{\mathcal{W}}$ walk relevance

Interaction block:

aggregate: $Z_t = \Lambda H_{t-1}$

combine: $H_t = \left(\mathcal{C}_t(Z_{t,K})\right)_K$

Walks:

ordered sequence of nodes

$$\mathcal{W} = (\ldots, J, K, L, \ldots)$$

# Computing walk-based relevance in GNNs [16]

**Walk-based relevance**

Sequential computation using first-order terms:

$$R_{\mathcal{W}} = \frac{\partial}{\partial \ldots} \left( \frac{\partial}{\partial \lambda^{\star}_{JK}} \left( \frac{\partial \ldots}{\partial \lambda^{\star}_{KL}} \cdot (\lambda^{\star}_{KL} - \widetilde{\lambda}^{\star}_{KL}) \right) (\lambda^{\star}_{JK} - \widetilde{\lambda}^{\star}_{JK}) \right) \cdot \ldots \tag{1}$$

with

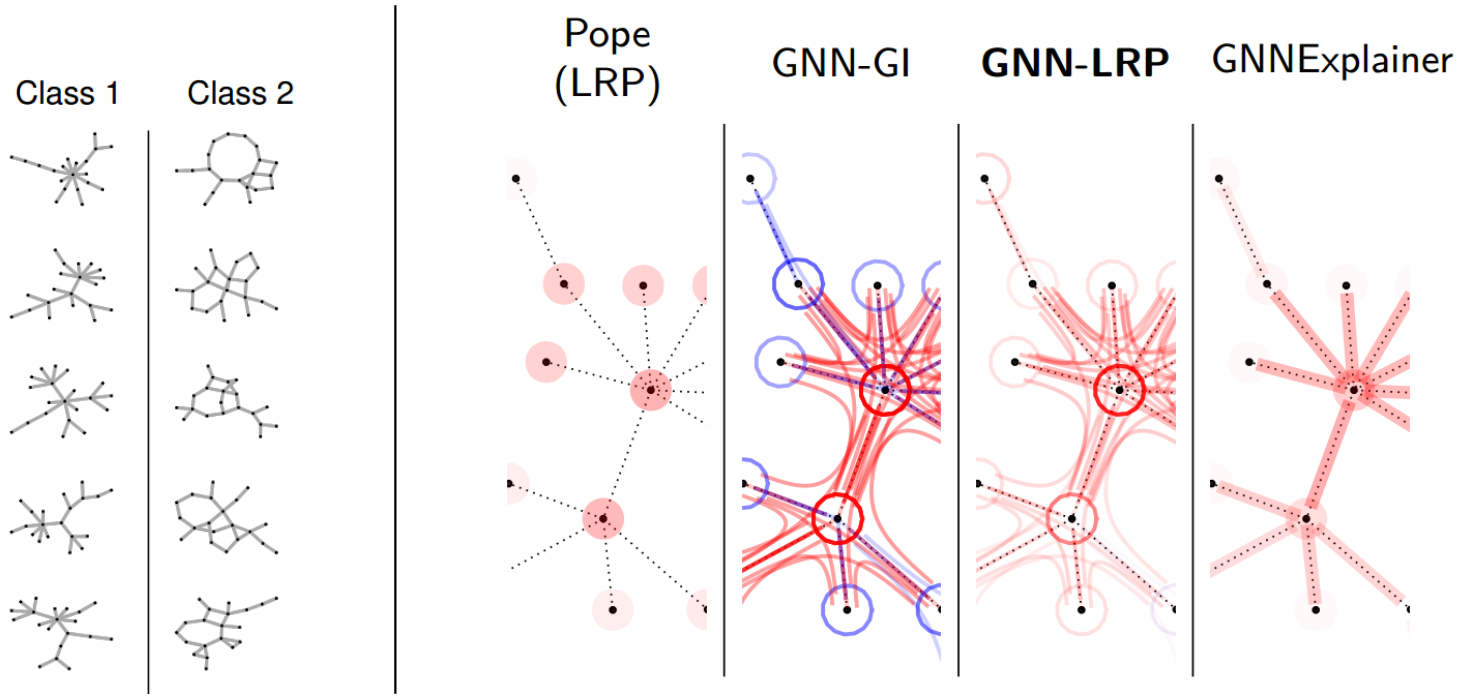
| | |
|---|---|
| $\lambda^{\star}_{KL}$ | connection weight of node $K$ and $L$ |
| '...' | leading and trailing terms |

# Explaining scale-free graphs

**Setup** Barabási-Albert graphs [1] with different growth beahavior:

| | |
|---|---|
| class 1 | growth parameter 1, new nodes attach preferably to higher-degree nodes |
| class 2 | higher growth parameter, new nodes attach preferably to low-degree nodes |

**Results**



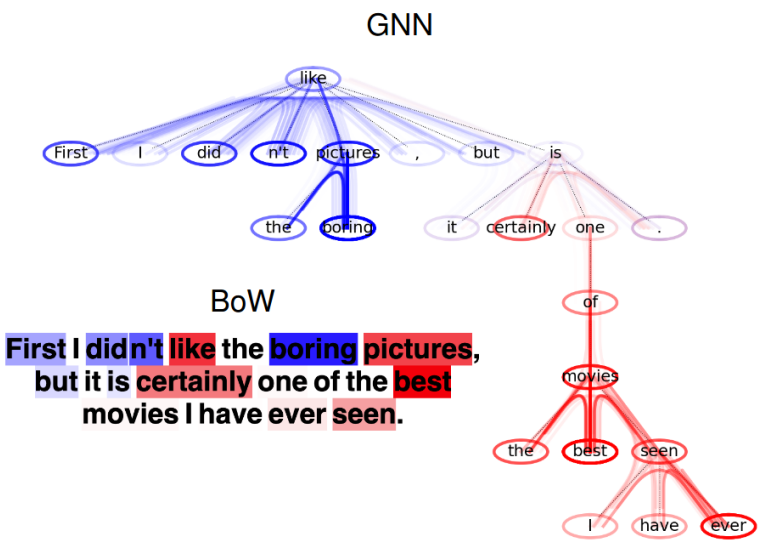
# Explaining graph predictions on real-world data [16]

## NLP

### Setup

- Sentiment classification on movie reviews (SST-2 [20])
- 2-block GCN to predict positive/negative sentiment using the sentence's parse-tree structure

### Results

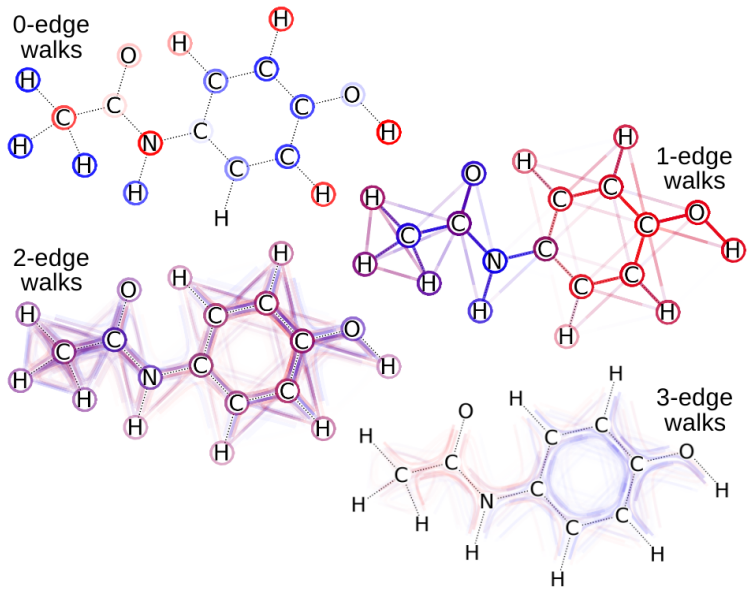



## Chemistry

### Setup

- Dipole moment prediction of the paracetamol molecule (QM9 dataset [15])
- 3-block GNN to predict molecular properties (SchNet model [18, 17])

### Results



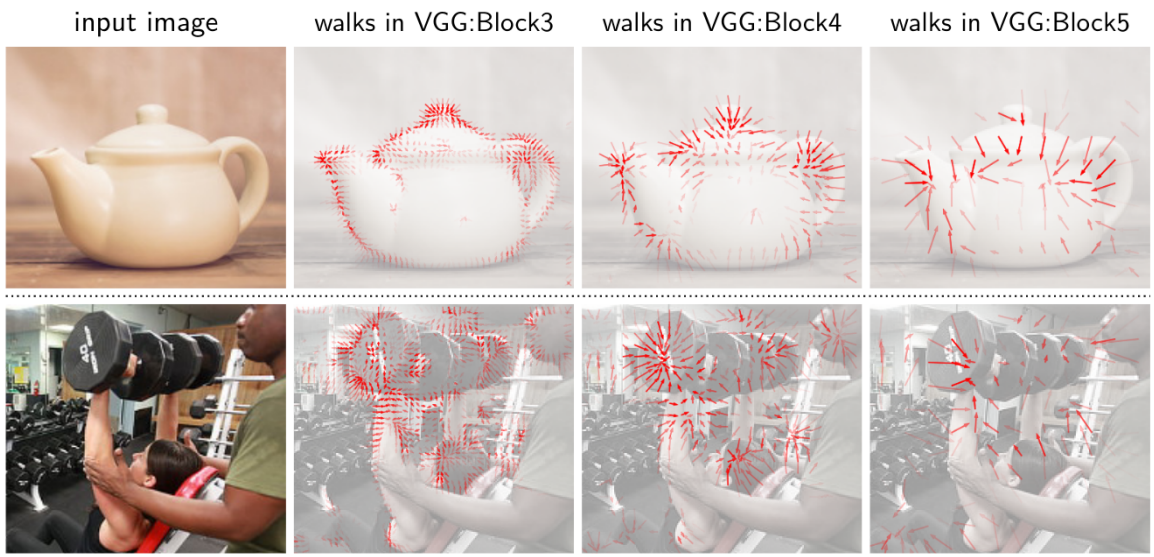
# Explaining graph predictions on real-world data

## Vision

### Setup

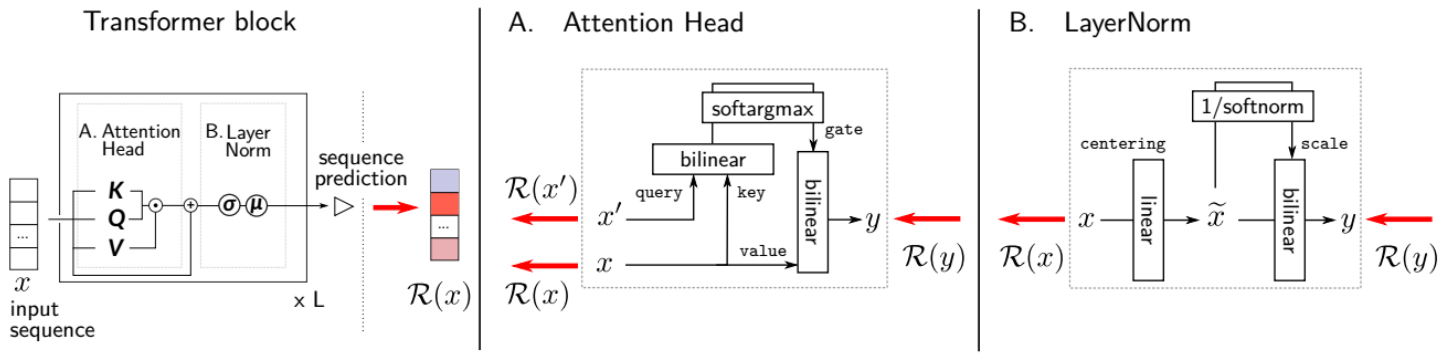
- View CNN as a GNN operating on pixel lattices using VGG-16 [19]
- Analyze relevance flow across convolution blocks from block input to output
- Visualize relevance as a vector field over the image to understand how information is processed throughout the model

### Results



# Explaining Transformer models

## Transformers




**Conservation**: Desired principle of conservation is not met when computing standard Gradient×Input in Transformers:

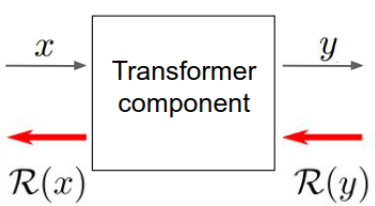
**Naive relevance approach**

Gradient x Input

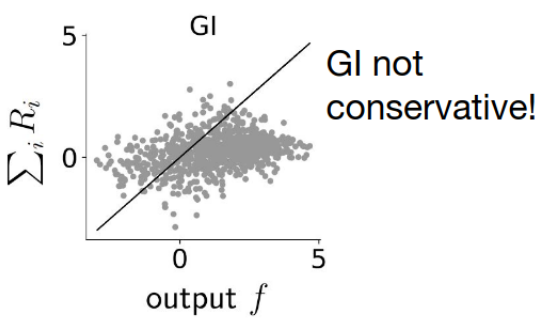
$$\mathcal{R}(x_i) = x_i \cdot (\partial f / \partial x_i)$$
$$\mathcal{R}(y_j) = y_j \cdot (\partial f / \partial y_j)$$

Relevance attribution

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

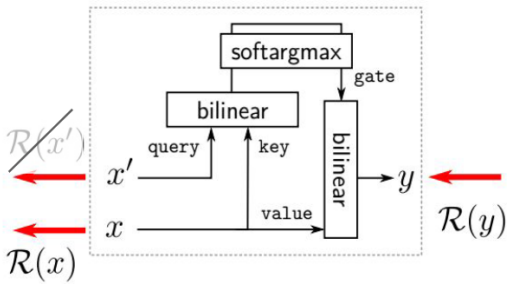



**Conservation?**



# Explaining Transformer models better [2]

## A. Self-Attention




Model function

$$y_j = \sum_i x_i \cdot \underbrace{\frac{\exp(q_{ij})}{\sum_{i'} \exp(q_{i'j})}}_{p_{ij}} \quad \text{with} \quad q_{ij} = \tfrac{1}{\sqrt{d_K}} x_i^\top W_K W_Q^\top x'_j$$

$p_{ij}$ → Detach in forward pass

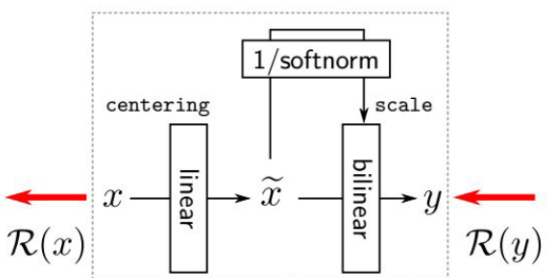
Relevance conservation

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

Improved propagation

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

## B. Layer normalization




Model function

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

→ Detach in forward pass

Relevance conservation

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

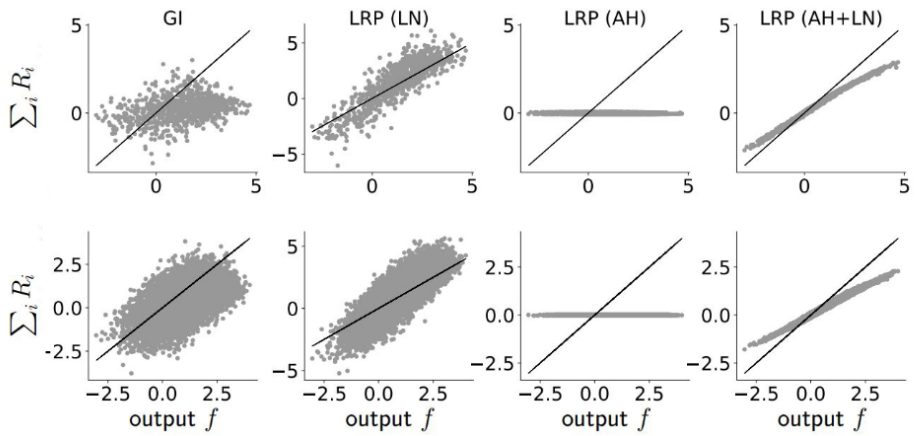
Improved propagation

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

# Better explanations for Transformers [2]

## Conservation

**Results**




## Faithfulness:

**Setup**

**Activation**

- Adding nodes to an empty sequence
- Observe true class output and compute area under the curve
- Higher scores are more faithful

**Pruning**

- Removing tokens from graph
- Observe change in output vector (MSE) and compute area under the curve
- Lower scores are better

**Results**

| 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** |

| 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** |

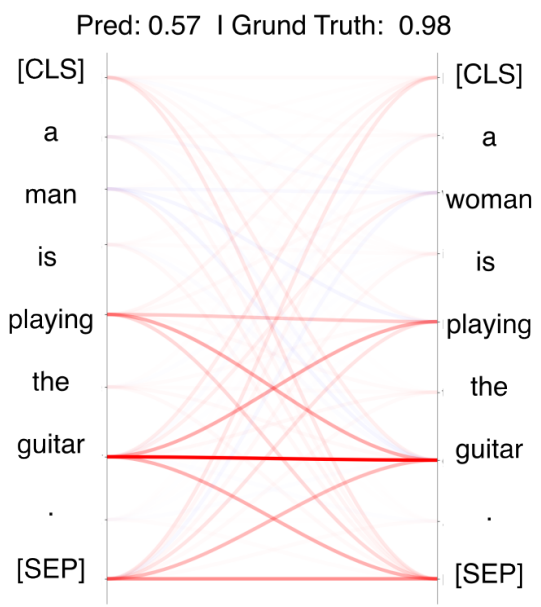
# Explaining Semantic Similarity in Language Models

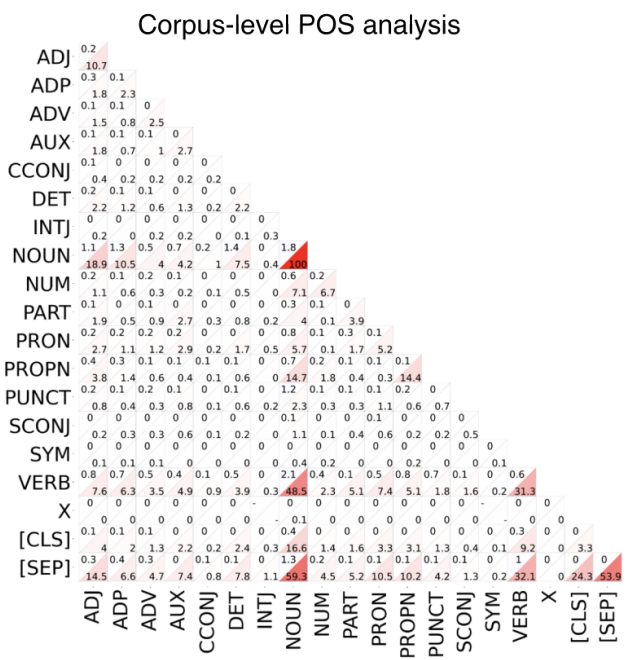
## Setup

- Use BiLRP to analyze text similarity models, e.g. Sentence BERT.[21]
- Resulting feature interactions provide insights into model strategies and can reveal simplified task-solving via simple token matching.
- An aggregation of different parts of speech, e.g. nouns or verbs, enables corpus-level insights

## Results

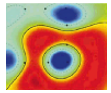
SBERT + Mean Pooling



# Motivation: Beyond neural networks

- Non-neural network algorithms such as kernel machines remain popular for unsupervised tasks, e.g. kernel density estimation, one-class SVMs and kernel k-means
- 'Neuralization' of these approaches allows to use explanations developed for neural networks, e.g. propagation-based methods such as LRP
- Allows their integration into existing theoretical frameworks and evaluation approaches for XAI



# Neuralizing Kernel Density Models [11, 10]

**Model:** Kernel density estimation (KDE) and one-class SVMs are non-neural network models for density estimation/ anomaly detection. The inlier score can be generically written as a weighted sum of kernel scores:

$$f(\boldsymbol{x}) = \sum_{j}^{N} \alpha_j \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}_j||^2)$$

For the detection of outliers (or *anomalies*) we can instead use $o(\boldsymbol{x}) = -\log f(\boldsymbol{x})$.

**Neuralization:** This quantity can be rewritten as a strictly equivalent two-layer neural network:

$$h_j = \gamma ||\boldsymbol{x} - \boldsymbol{x}_j|| - \log \alpha_j \qquad \text{(layer 1)}$$

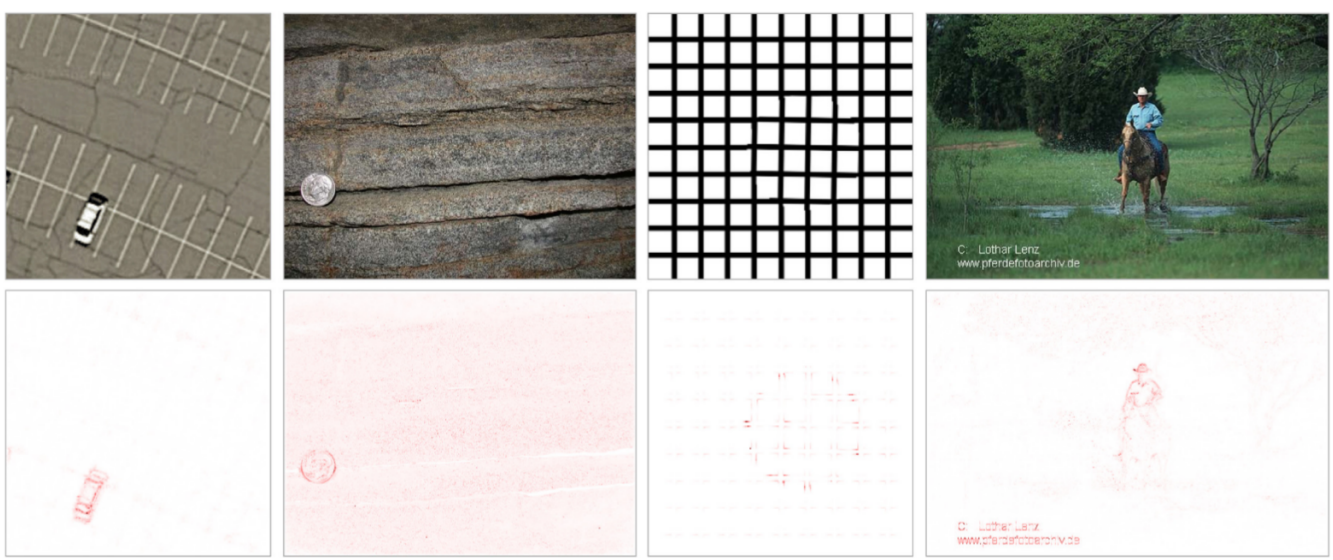
$$o(\boldsymbol{x}) = -\log(\sum_{j}^{N} \exp(-h_j)) \qquad \text{(layer 2)}$$

# Explaining Anomalies

## Setup

- Once-Class Learning. Goal: separate origin from data using the kernel space
- Build a One-Class Support Vector Machine model (OC-SVM) by extracting $7 \times 7$ patches from larger images, identify prototypes and train for binary task (inlinear vs outlier)
- Explain what makes a certain image an outlier in this model using OC-DTD (One Class Deep Taylor Decomposition [10].

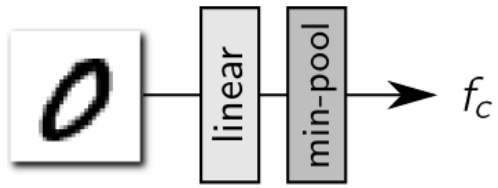
## Results



# Neuralizing k-Means Clustering [9]

**Model:** Assign cluster membership of $\boldsymbol{x}$ according to the distance of the closest cluster centroid $\boldsymbol{\mu}_c$:

$$\forall_{k \neq c}: \quad ||\boldsymbol{x} - \boldsymbol{\mu}_c||^2 < ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2$$




**Neuralization:**
Neuralized k-Means.

$$h_k = \boldsymbol{w}_k^T \boldsymbol{x} + b_k \qquad \text{(layer 1)}$$
$$f_c = \min_{k \neq c} \{h_k\} \qquad \text{(layer 2)}$$

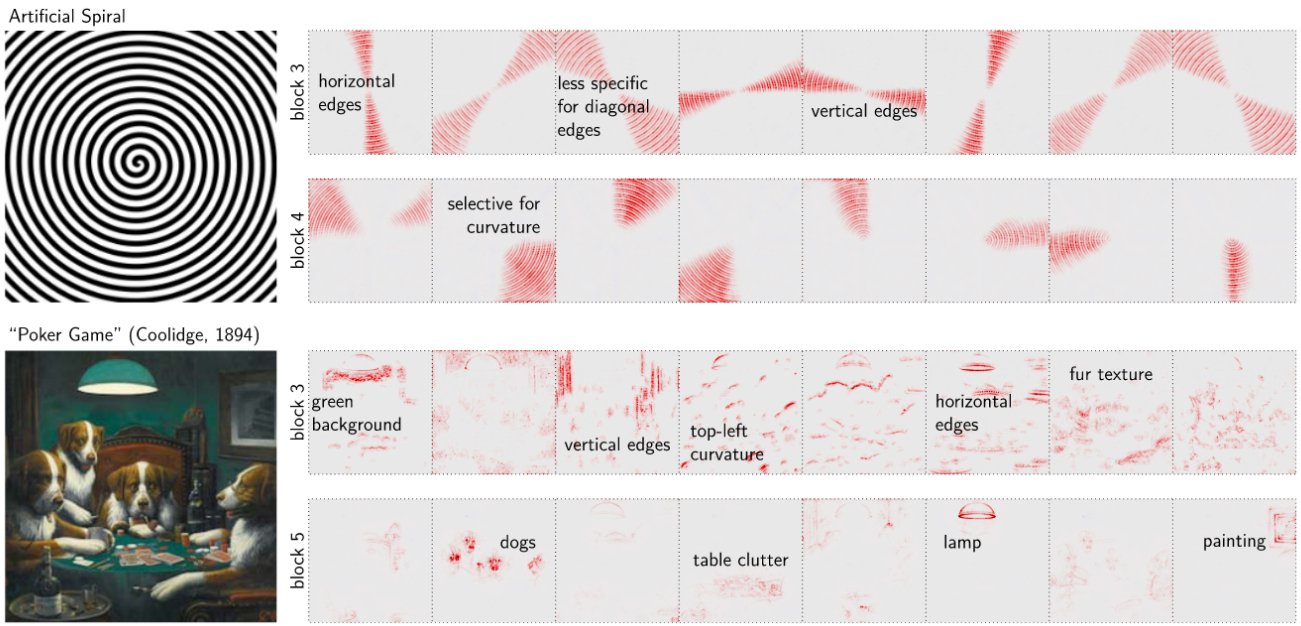
with $\boldsymbol{w}_k = 2(\boldsymbol{\mu}_c - \boldsymbol{\mu}_k)$ and $b_k = ||\boldsymbol{\mu}_k||^2 - ||\boldsymbol{\mu}_c||^2$ and assignment to cluster $c$ if $f_c(\boldsymbol{x}) > 0$.

# Explaining Deep k-Means Clustering

## Setup

- Cluster VGG-16 [19] activations for different processing blocks
- Explain each of the resulting clustering ($k = 8$ clusters, each column corresponds to one cluster)

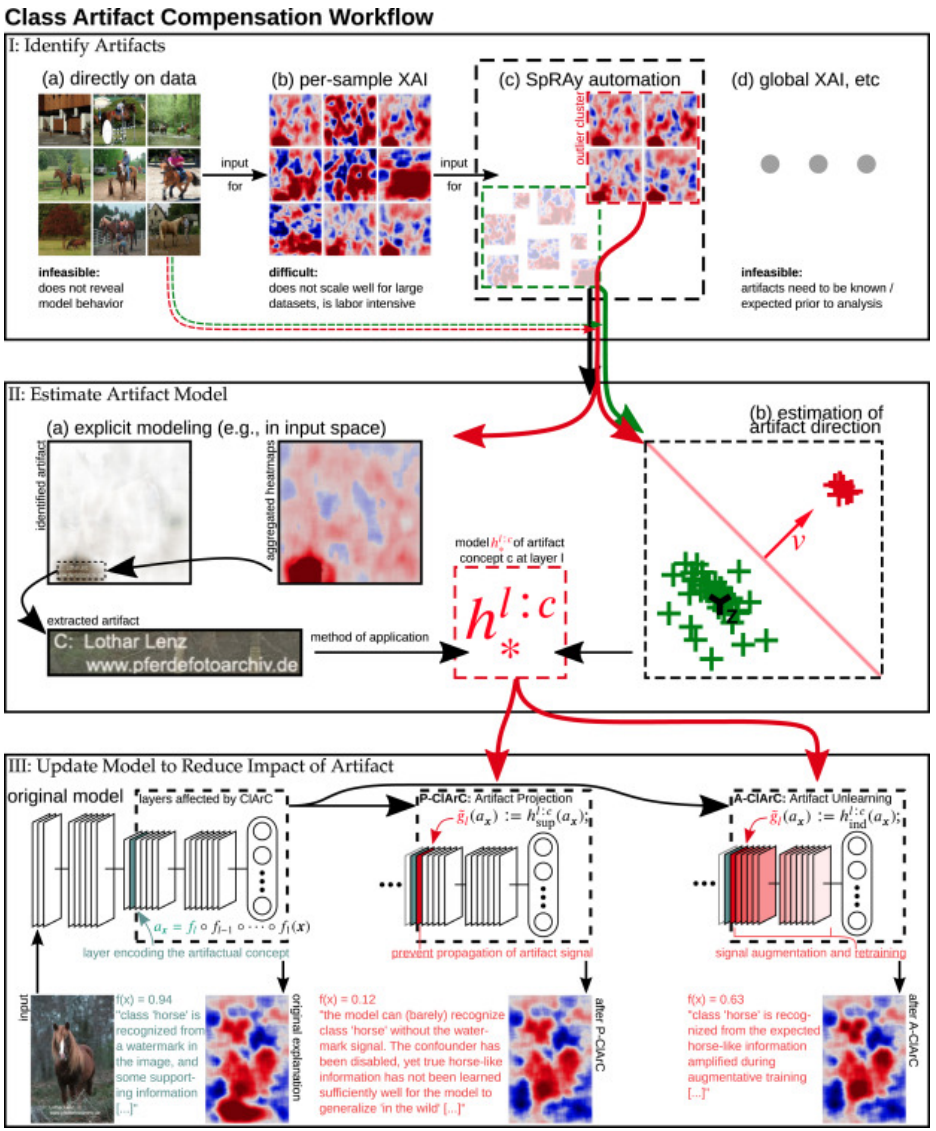
## Results



# Applications of XAI

## Improving Models

Debugging and Robustification [3]

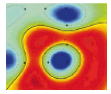



## Scientific Insights

- Detection of new structures, e.g. in multi-modal models and XAI [8]
- Investigating model bias, e.g. in language models [2]
- Understanding patterns in data, e.g. correlations in histological, clinical and molecular data for cancer profiling [5]
- XAI-informed materials modeling, catalysis, and drug design [12]
- ...

# Summary

- Careful analysis of model structure is important and, e.g., motivates to go beyond first-order explanations:
  - Second-order in deep similarity models
  - Higher-order in graph neural networks
- Neuralizing non-neural network models can be used to make use of existing explanation techniques, e.g. LRP for k-means clustering and kernel density estimation
- Faithful explanations for widely used models are crucial to support the safe and robust use of machine learning, e.g. for insights and sensitive applications.
- Beyond explaining predictions, explanations can be used to improve models, support insights and understand large data characteristics.



# Bibliography I


[1] R. Albert and A.-L. Barabási.
Statistical mechanics of complex networks.
*Reviews of Modern Physics*, 74(1):47–97, Jan. 2002.

[2] 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. JMLR.org, 2022.

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

[4] 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*, volume 70 of *Proceedings of Machine Learning Research*, pages 342–350. PMLR, 2017.

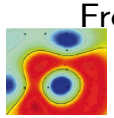
[5] A. Binder, M. Bockmayr, M. Hägele, S. Wienert, D. Heim, K. Hellweg, M. Ishii, A. Stenzinger, A. Hocke, C. Denkert, K.-R. Müller, and F. Klauschen.
Morphological and molecular breast cancer profiling through explainable machine learning.
*Nature Machine Intelligence*, 3:1–12, 04 2021.

[6] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani, and G. Montavon.
Building and interpreting deep similarity models.
*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1149–1161, 2022.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy.
Explaining and harnessing adversarial examples.
In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] A. Holzinger.
Explainable ai and multi-modal causability in medicine.
*i-com*, 19(3):171–179, 2020.

[9] J. Kauffmann, M. Esders, L. Ruff, G. Montavon, W. Samek, and K.-R. Müller.
From clustering to cluster explanations via neural networks, 2021.

# Bibliography II


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

[11] J. R. Kauffmann, L. Ruff, G. Montavon, and K. Müller.
The clever hans effect in anomaly detection.
*CoRR*, abs/2006.10609, 2020.

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

[13] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio.
On the number of linear regions of deep neural networks.
*Advances in Neural Information Processing Systems*, 4(January):2924–2932, 2014.

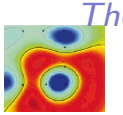
[14] A. Nguyen, J. Yosinski, and J. Clune.
Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.
In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[15] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld.
Quantum chemistry structures and properties of 134 kilo molecules.
*Scientific Data*, 1(40022), 2014.

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

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

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

# Bibliography III


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

[20] 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*, pages 1631–1642. Association for Computational Linguistic, 2013.

[21] A. Vasileiou and O. Eberle.
Explaining text similarity in transformer models.
In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.