

Seminar Series on Graph Neural Networks 07

Explainable graph neural networks

Yong-Min Shin

School of Mathematics and Computing (Computational Science and Engineering)

Yonsei University

2025.05.26



Before going in....

Towards application of graph neural networks

Towards efficient graph learning

Explainable graph neural networks

Fundamental topics on graph neural networks

On the representational power of graph neural networks

A graph signal processing viewpoint of graph neural networks

From label propagation to graph neural networks

On the problem of oversmoothing and oversquashing

Introduction to graph mining and graph neural networks
(Basic overview to kick things off)

* Presentation slides are available at:
jordan7186.github.io/presentations/



Objectives

1. Understanding general concept of explainable AI: Why & How?
2. A general understanding of explainable AI in graph learning
3. Subtopic: Explaining GNNs with attention

Understanding the concepts of explainable AI

The early ‘why’ part was based on the content from
Samek & Müller: Towards Explainable Artificial Intelligence. Explainable AI 2019: 5-22

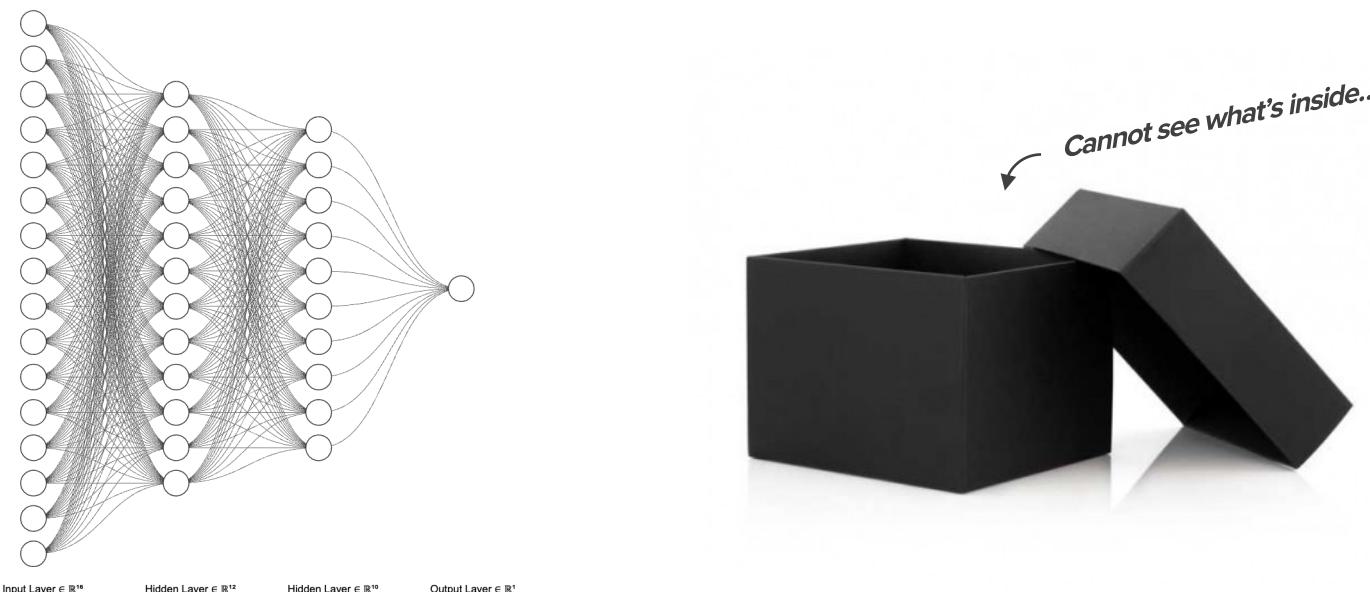
Why is interpretation an important question?

Generally, neural-network models are considered as 'black-box' models

1. Model weights are difficult to understand by humans.

[1]: "...due to their nested non-linear structure, these powerful models have been generally considered "black boxes"."

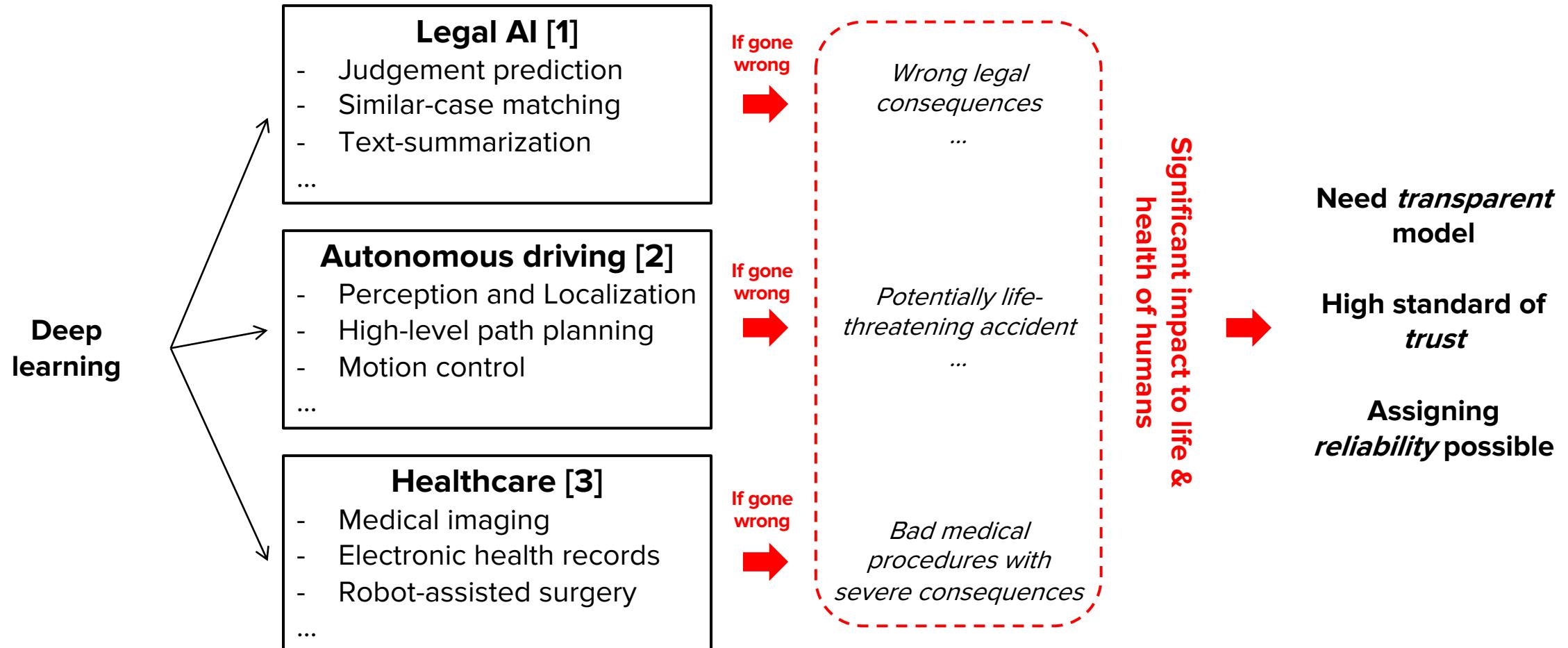
[2]: "These rules [model weights], because they're generated by the algorithm, can run counter to human intuition and be difficult, if not impossible, to decipher"



Why is interpretation an important question?

XAI becomes more critical in serious applications.

2. Interpretation brings **trust** and **reliability** to the table

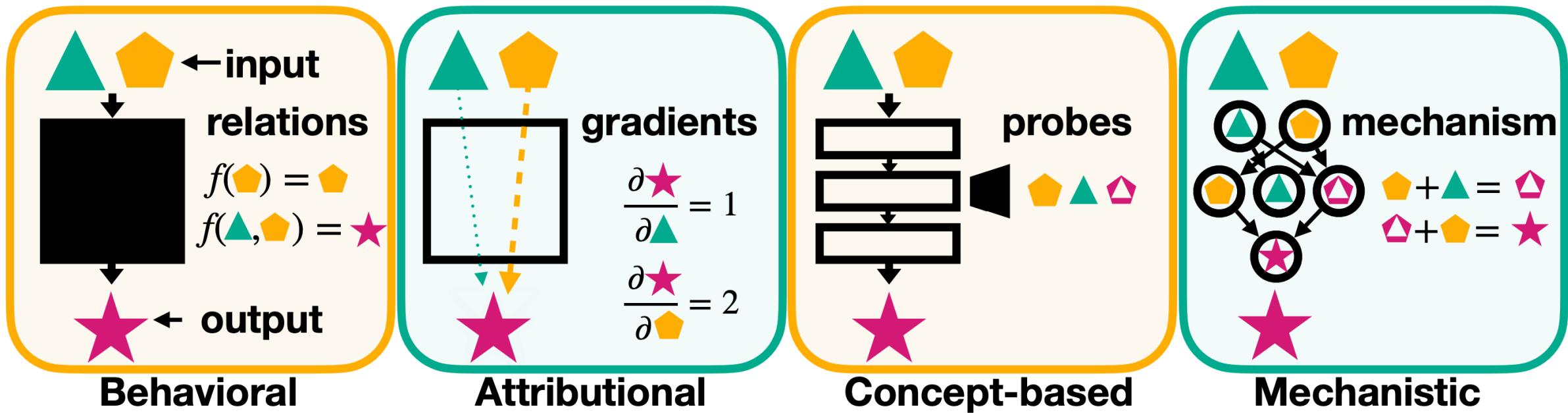


[1] Zhong, Haoxi, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. "How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence." arXiv preprint arXiv:2004.12158 (2020).

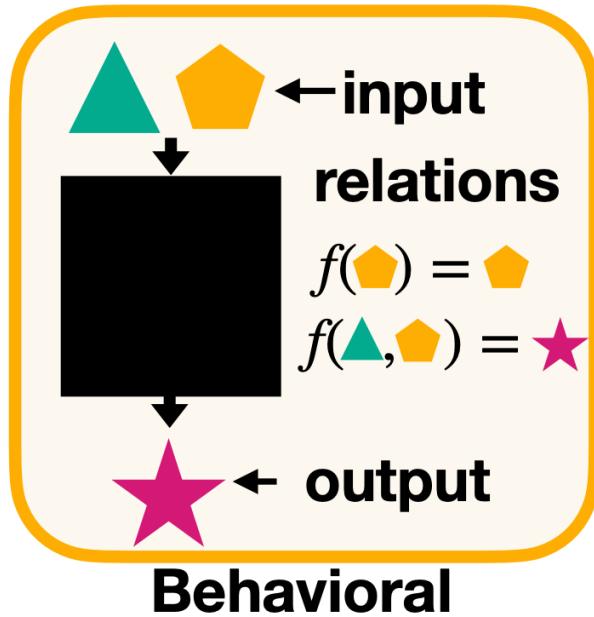
[2] Grigorescu, Sorin, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. "A survey of deep learning techniques for autonomous driving." Journal of Field Robotics 37, no. 3 (2020): 362-386.

[3] Esteva, Andre, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. "A guide to deep learning in healthcare." Nature medicine 25, no. 1 (2019): 24-29.

A general overview on the types of XAI methods [1]



A general overview on the types of XAI methods [1]



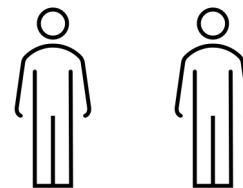
Example: Shapley value-based explanations

- Only interested in input-output relations
- Treats the model as a complete black-box
- Main limitation: Exponential computation & How to express the absence of a feature?
 - Many approaches are designed to approximate this
 - ex. KernelSHAP

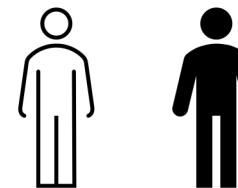
*Basic concept of Shapley values by a glove-selling game

The rules & setting of selling gloves

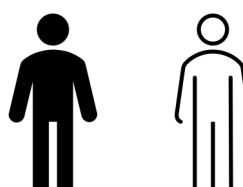
1. The gloves are **\$1** each.
2. They must be sold in pairs.
3. Person A has **9** gloves.
4. Person B has **3** gloves.



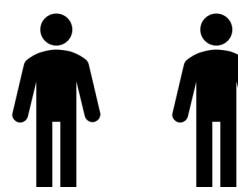
No A, no B



No A, B



A, no B

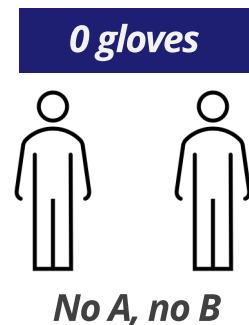


A & B

*Basic concept of Shapley values by a glove-selling game

The rules & setting of selling gloves

1. The gloves are **\$1** each.
2. They must be sold in pairs.
3. Person A has **9** gloves.
4. Person B has **3** gloves.

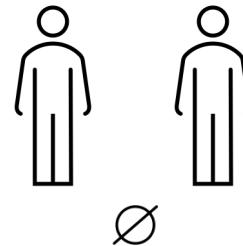


*Basic concept of Shapley values by a glove-selling game

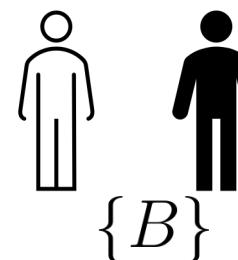
The rules & setting of selling gloves

1. The gloves are **\$1** each.
2. They must be sold in pairs.
3. Person A has **9** gloves.
4. Person B has **3** gloves.

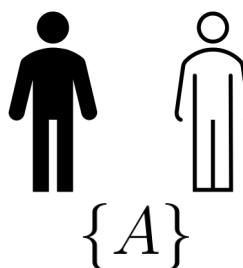
0 gloves



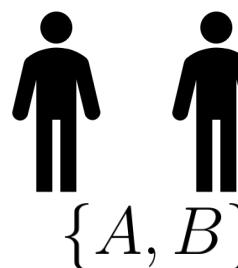
2 gloves



8 gloves



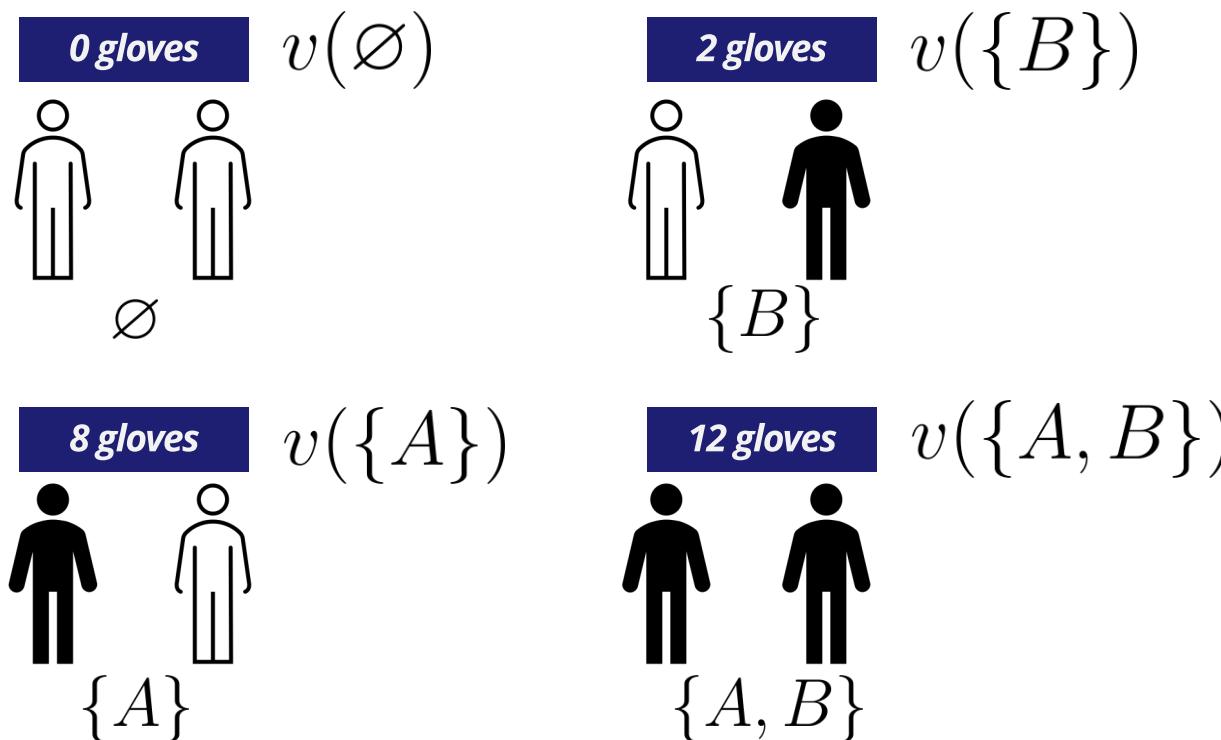
12 gloves



*Basic concept of Shapley values by a glove-selling game

The rules & setting of selling gloves

1. The gloves are **\$1** each.
2. They must be sold in pairs.
3. Person A has **9** gloves.
4. Person B has **3** gloves.



*Basic concept of Shapley values by a glove-selling game

The rules & setting of selling gloves

1. The gloves are **\$1** each.
2. They must be sold in pairs.
3. Person A has **9** gloves.
4. Person B has **3** gloves.

$$v(\emptyset) = 0 \xrightarrow{+2} v(\{B\}) = 2$$



$$v(\{A\}) = 8 \xrightarrow{+4} v(\{A, B\}) = 12$$

*Basic concept of Shapley values by a glove-selling game

What is the overall contribution of each player?

$$\begin{array}{ccc}
 v(\emptyset) = 0 & \xrightarrow{+2} & v(\{B\}) = 2 \\
 & \downarrow +8 & \downarrow +10 \\
 v(\{A\}) = 8 & \xrightarrow{+4} & v(\{A, B\}) = 12
 \end{array}$$

- As an example, concentrate on A.
- A contributes **+8** when there are no one.
- A contributes **+10** when there is B.
- How do we determine A's contribution overall?

- Take the average for all cases.

*Basic concept of Shapley values by a glove-selling game

What is the overall contribution of each player?

$$v(\emptyset) = 0 \xrightarrow{\Delta_v(B, \emptyset) \text{ +2}} v(\{B\}) = 2$$

$$\Delta_v(A, \emptyset) \text{ +8} \downarrow \qquad \qquad \qquad \downarrow \text{ +10 } \Delta_v(A, \{B\})$$

$$v(\{A\}) = 8 \xrightarrow{\Delta_v(B, \{A\}) \text{ +4}} v(\{A, B\}) = 12$$

$$\phi_v(A) = \frac{\Delta_v(A, \emptyset) + \Delta_v(A, \{B\})}{2} = 9$$

$$\phi_v(B) = \frac{\Delta_v(B, \emptyset) + \Delta_v(B, \{A\})}{2} = 3$$

*Basic concept of Shapley values by a glove-selling game

What is the overall contribution of each player?

$$\phi_v(i) = \sum_{S \subseteq N} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \Delta_v(i, S)$$

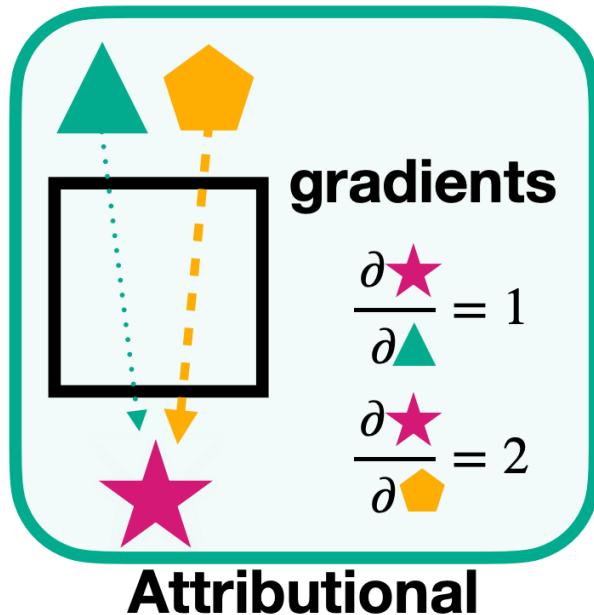
$$\frac{1}{|N|} \frac{1}{\binom{|N|-1}{|S|}}$$

$$\phi_v(i) = \frac{1}{n!} \sum_{\pi \subseteq \Pi(N)} \overbrace{[v(P_{\pi[:i]} \cup \{i\}) - v(P_{\pi[:i]})]}^{\text{Marginal contribution } \Delta_v(i, P_{\pi[:i]})}$$

All possible permutations of N players

Subset of players preceding player i

A general overview on the types of XAI methods [1]



- **Most XAI works (especially early works) fall into this category.**
- *"How much can we attribute the output back to the input?"*
- Tend to be highly heuristic (exceptions include Integrated Gradients & Deep Taylor Decomposition)
 - Lot of "Sanity check" work exposes this limitation (Refer to, for example, [2])
- For graphs, GNNExplainer-types belongs to this category
 - How much of the explanation generated from the XAI method is from the model vs. from the XAI method? (see [3] for similar argument)

[1] Bereska & Gavves, "Mechanistic interpretability or AI safety: A review", arXiv 2024

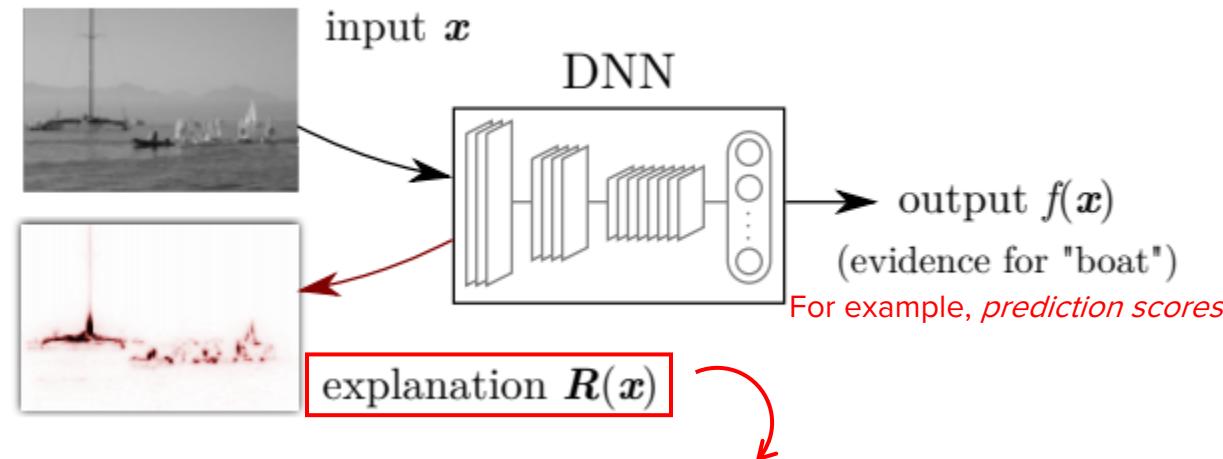
[2] Adebayo et al., "Sanity Checks for Saliency Maps", NeurIPS 2018 (+2000 citations)

[3] Miao et al., "Interpretable and generalizable graph learning via stochastic attention mechanism", ICML 2022

*Well-known classical approaches in attributional methods

Sensitivity analysis (SA) [1]

Consider a neural network where the input is an image and the task is image classification.



We can generate an “explanation” of the prediction as a form of **heatmap**.

In sensitivity analysis, the pixel-wise value of the heatmap is the derivative of the score with respect to the image.

$$R_i(\mathbf{x}) = \left(\frac{\partial f}{\partial x_i} \right)^2$$

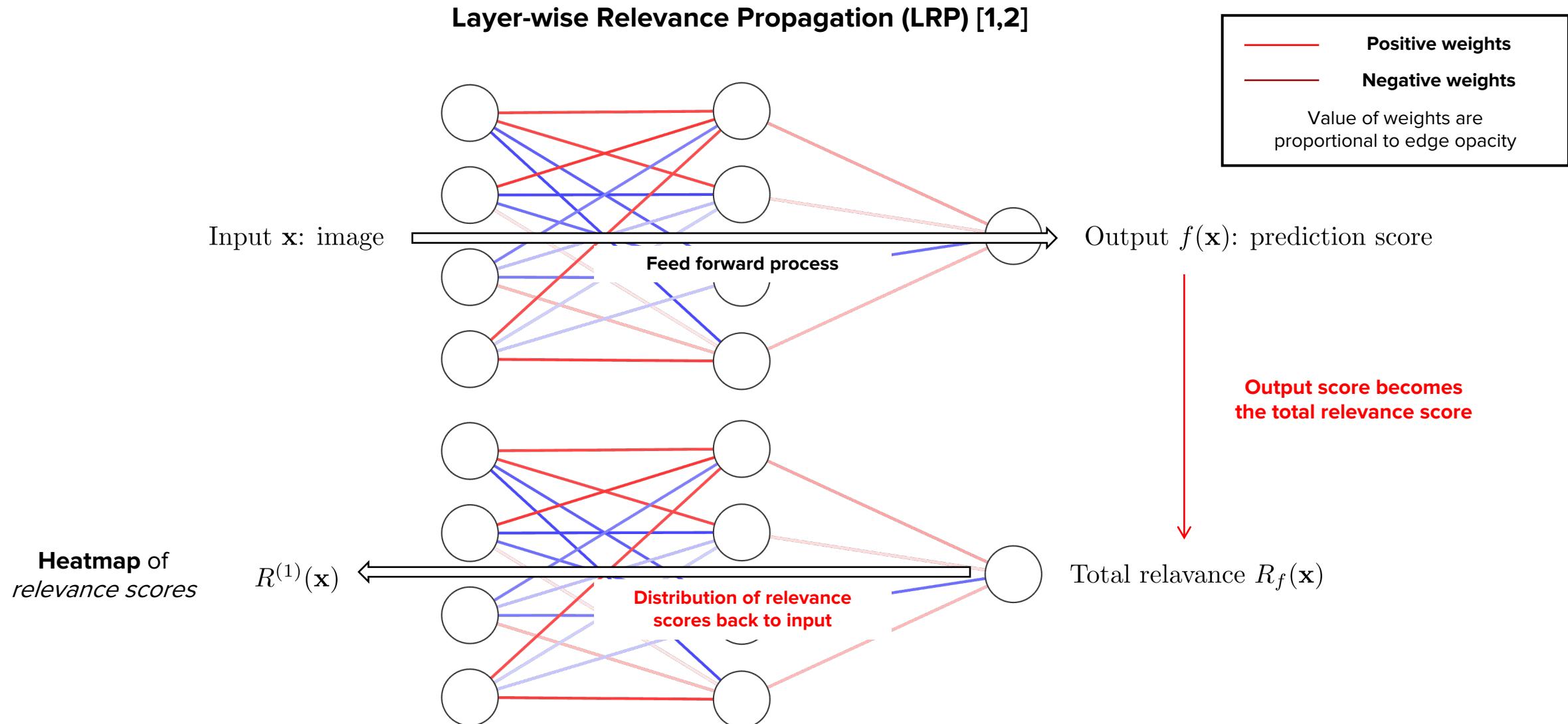
- Note that this is easily acquired via back-propagation via modern machine learning libraries
- Also, SA provides explanation of the *variation* of the function, not the function itself [2].
- Known to be vulnerable to ‘shattered gradient’ [3], where the gradients in standard feedforward networks increasingly resemble *white noise*.

[1] Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks." Digital Signal Processing 73 (2018): 1-15. (The image is also from the paper.)

[2] Włodzimierz Samek, Grégoire Montavon, and Klaus-Robert Müller. "Tutorial on Interpreting and Explaining Deep Models in Computer Vision". In CVPR 2018.

[3] Balduzzi, David, Marcus Frean, Lennox Leary, J. P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. "The shattered gradients problem: If resnets are the answer, then what is the question?." arXiv preprint arXiv:1702.08591 (2017).

*Well-known classical approaches in attributional methods

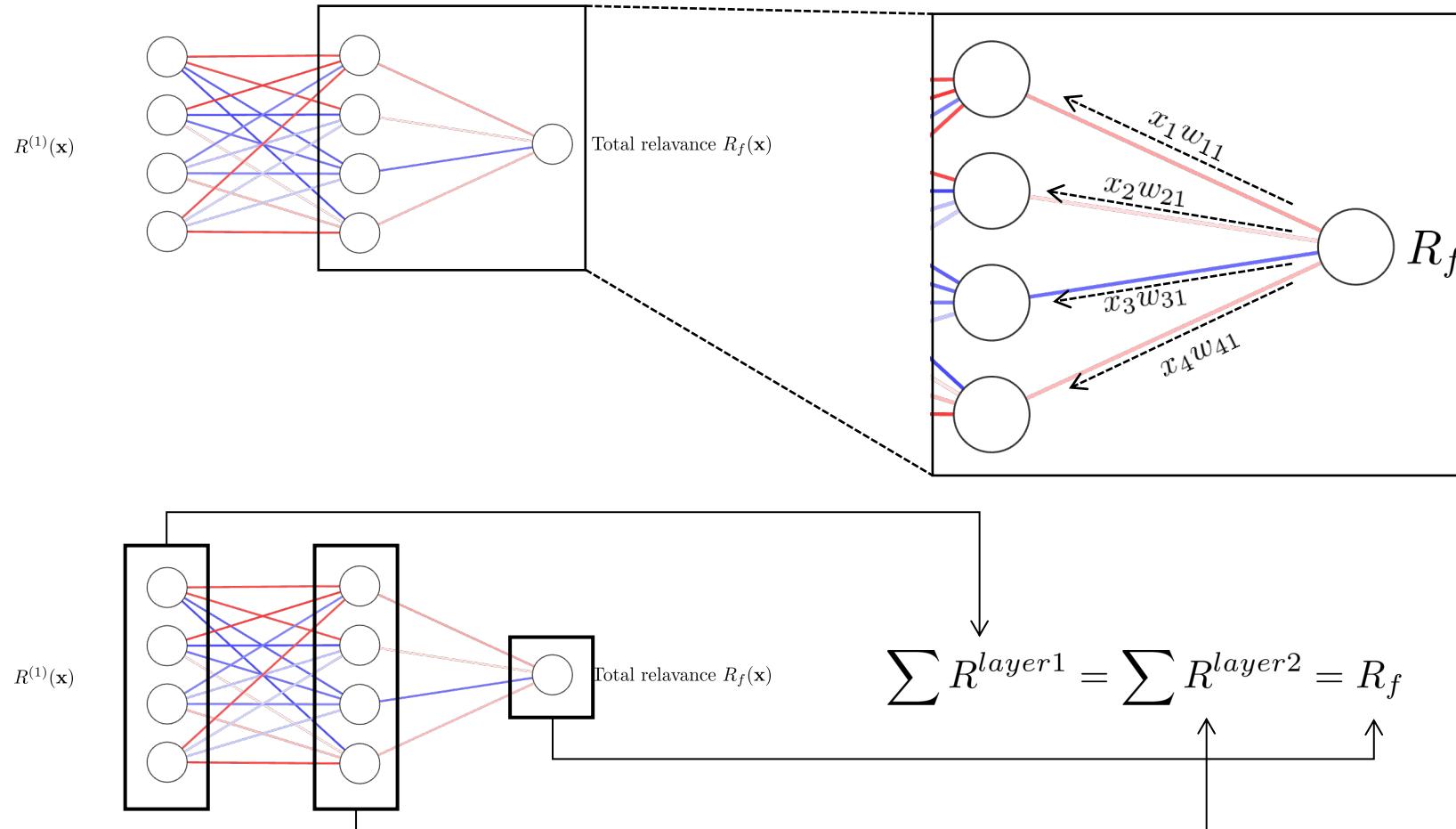


[1] Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one* 10, no. 7 (2015): e0130140.

[2] Binder, Alexander, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. "Layer-wise relevance propagation for deep neural network architectures." In *Information Science and Applications (ICISA) 2016*, pp. 913-922. Springer, Singapore, 2016.

*Well-known classical approaches in attributional methods

Layer-wise Relevance Propagation (LRP) [1,2]



— Positive weights
 - Negative weights
 Value of weights are proportional to edge opacity

$$R_i = \sum_j \frac{x_i w_{ij}}{\sum_{i'} x_{i'} w_{ij}} R_f$$

normalization

The relevance scores are distributed *proportional to the neurons' activation during feed-forwarding*.

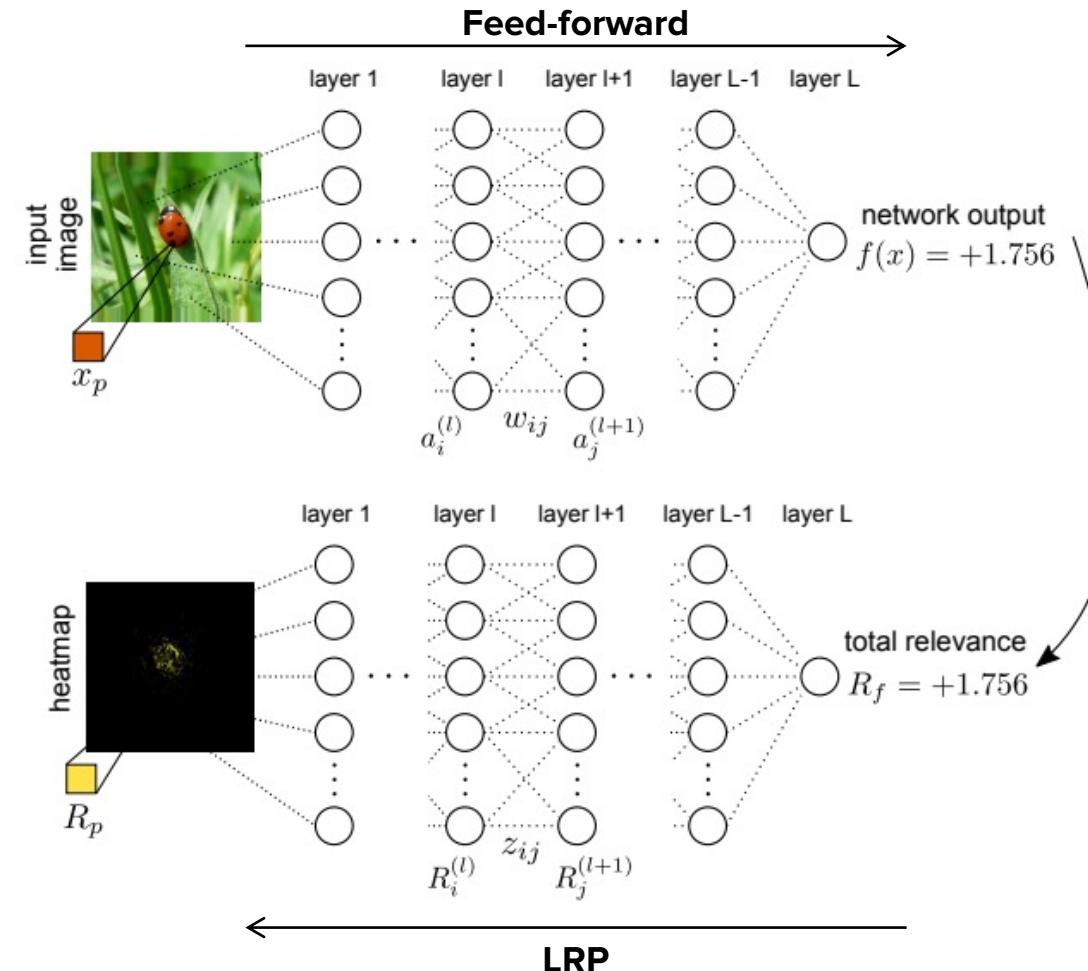
As a result, the total sum of relevance scores are preserved for all layers.

[1] Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one* 10, no. 7 (2015): e0130140.

[2] Binder, Alexander, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. "Layer-wise relevance propagation for deep neural network architectures." In *Information Science and Applications (ICISA) 2016*, pp. 913-922. Springer, Singapore, 2016.

*)Well-known classical approaches in attributional methods

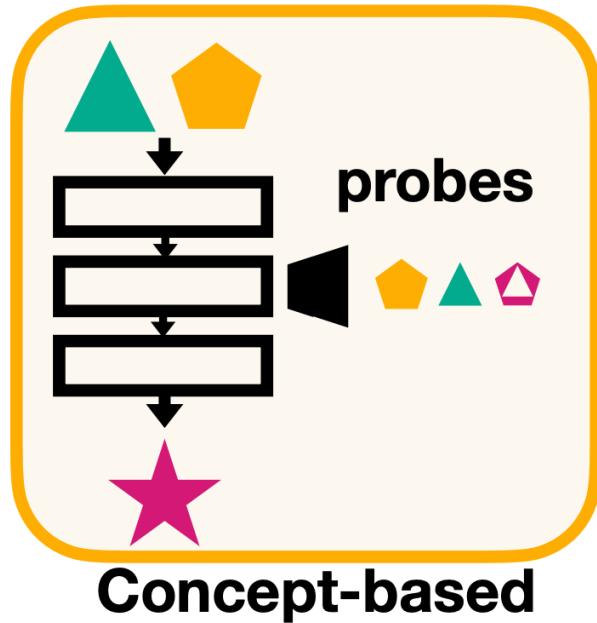
Layer-wise Relevance Propagation (LRP) [1,2]



[1] Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one* 10, no. 7 (2015): e0130140.

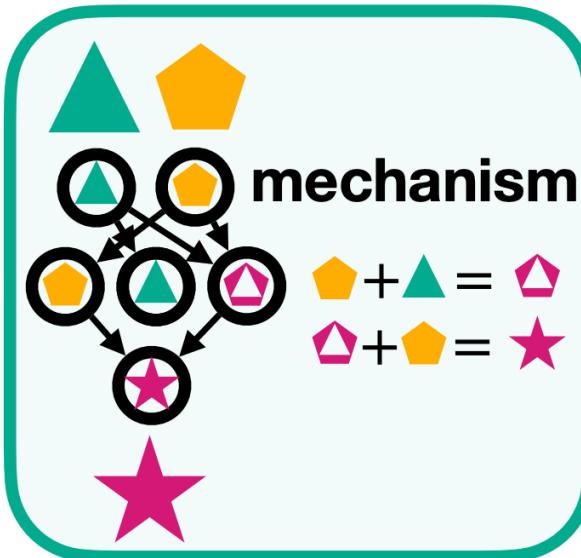
[2] Binder, Alexander, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. "Layer-wise relevance propagation for deep neural network architectures." In *Information Science and Applications (ICISA) 2016*, pp. 913-922. Springer, Singapore, 2016.

A general overview on the types of XAI methods [1]



- Learns a method to extract explainable information from the internal representations
- Works include learning a probe with some unsupervised loss
- For graphs, PAGE [2] also directly utilizes the node & graph level representations.
- However, the concept-based method cannot escape the previous criticism: *Does it really explain the model? How much is the explanation from the explanation method itself?*

A general overview on the types of XAI methods [1]



Hypothesis of Mechanistic Interpretability

- Models learn human-comprehensible algorithms and can be understood.
- They are not comprehensible by default, and we need to do some work to make it legible.

- A (relatively new) sub-field of interpretability
- Mechanistic interpretability is done by...
 - **Rigorous (and almost surgical) observations** of the model ‘without tricking ourselves’
 - Most works are case studies, and does not know what it would find at the start of the investigation. **Most discoveries is the authors ‘noticing common trends’**
 - Since they are case studies, Transformers [2] are typically the model of interest
- Goal: Reverse engineer neural networks
(Analogy: Binary of a program → Source code. But is it just an analogy though? [3])

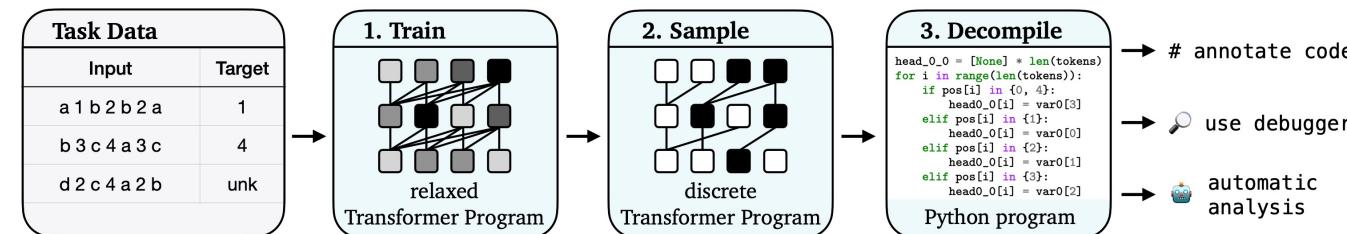


Figure 1: We design a modified Transformer that can be trained on data and then automatically discretized and converted into a human-readable program. The program is functionally identical to the Transformer, but easier to understand—for example, using an off-the-shelf Python debugger.

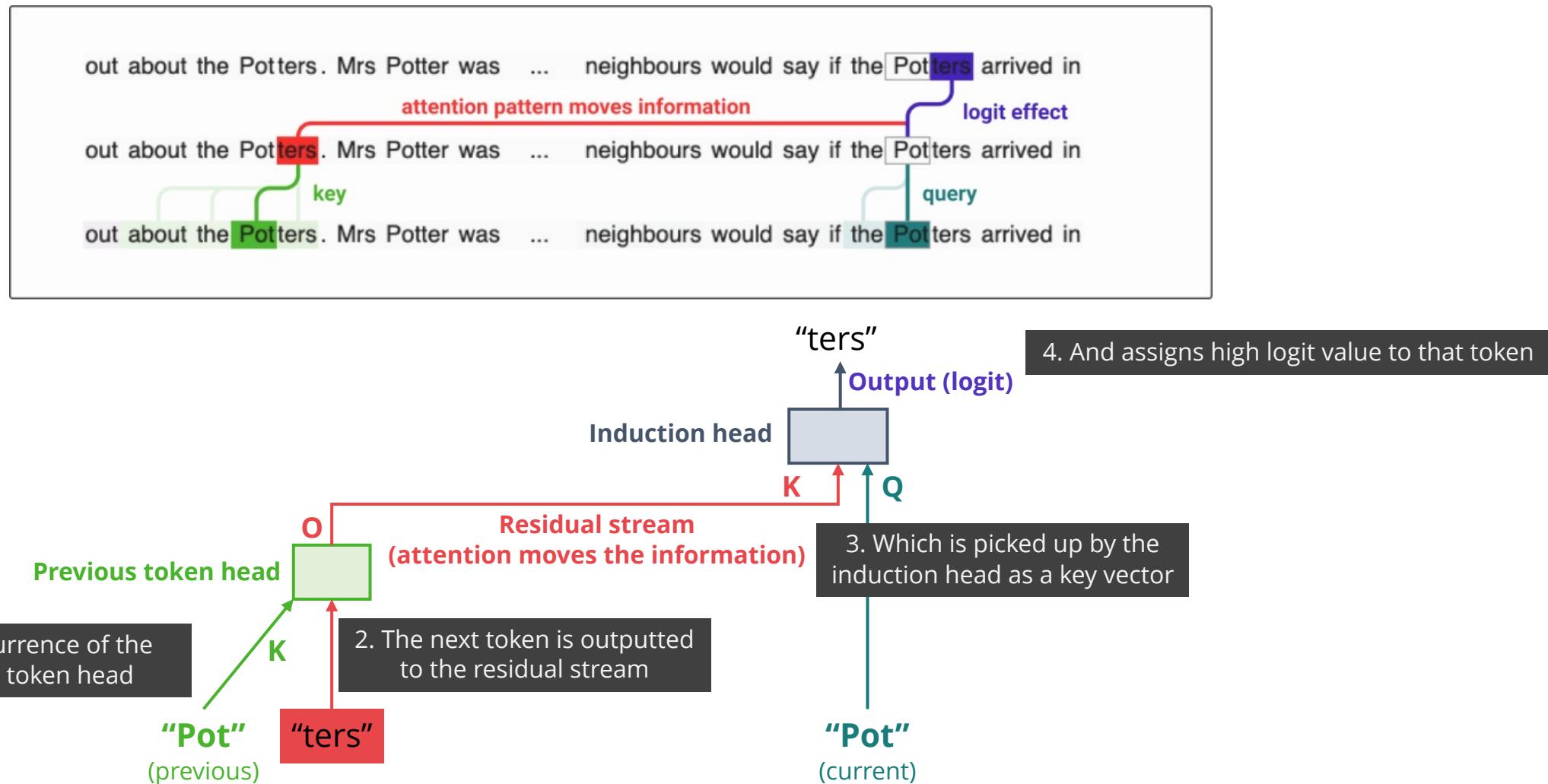
[1] Bereska & Gavves, “Mechanistic interpretability or AI safety: A review”, arXiv 2024

[2] Vaswani et al., “Attention is all you need”, NeurIPS 2017

(Bottom right figure) [3] Friedman et al., “Learning transformer programs”, NeurIPS 2024 (Oral)

*A bulk of the content of this slide is from Neel Nanda’s talk “Open Problems in Mechanistic Interpretability: A Whirlwind Tour | Neel Nanda | EAGxVirtual 2023” on Youtube.

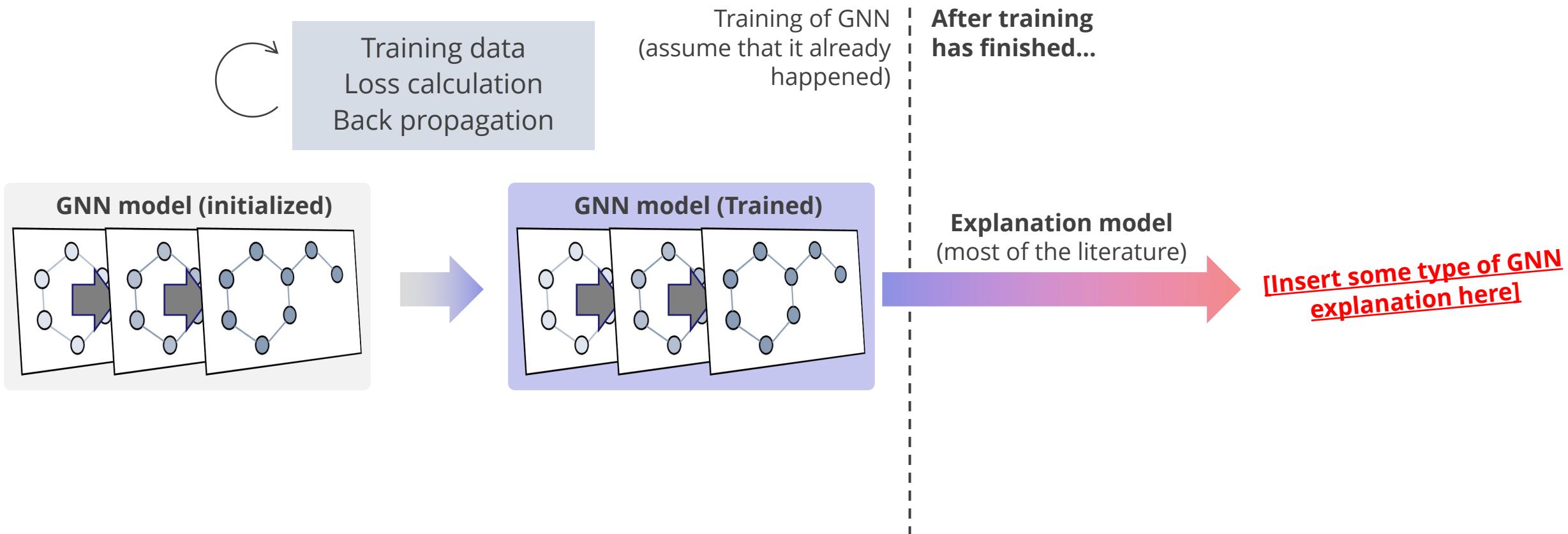
*The most famous discovery in Mech. Interp.: Induction heads [1]



An expanded introduction to explaining graph learning

What does it mean to explain GNN models?

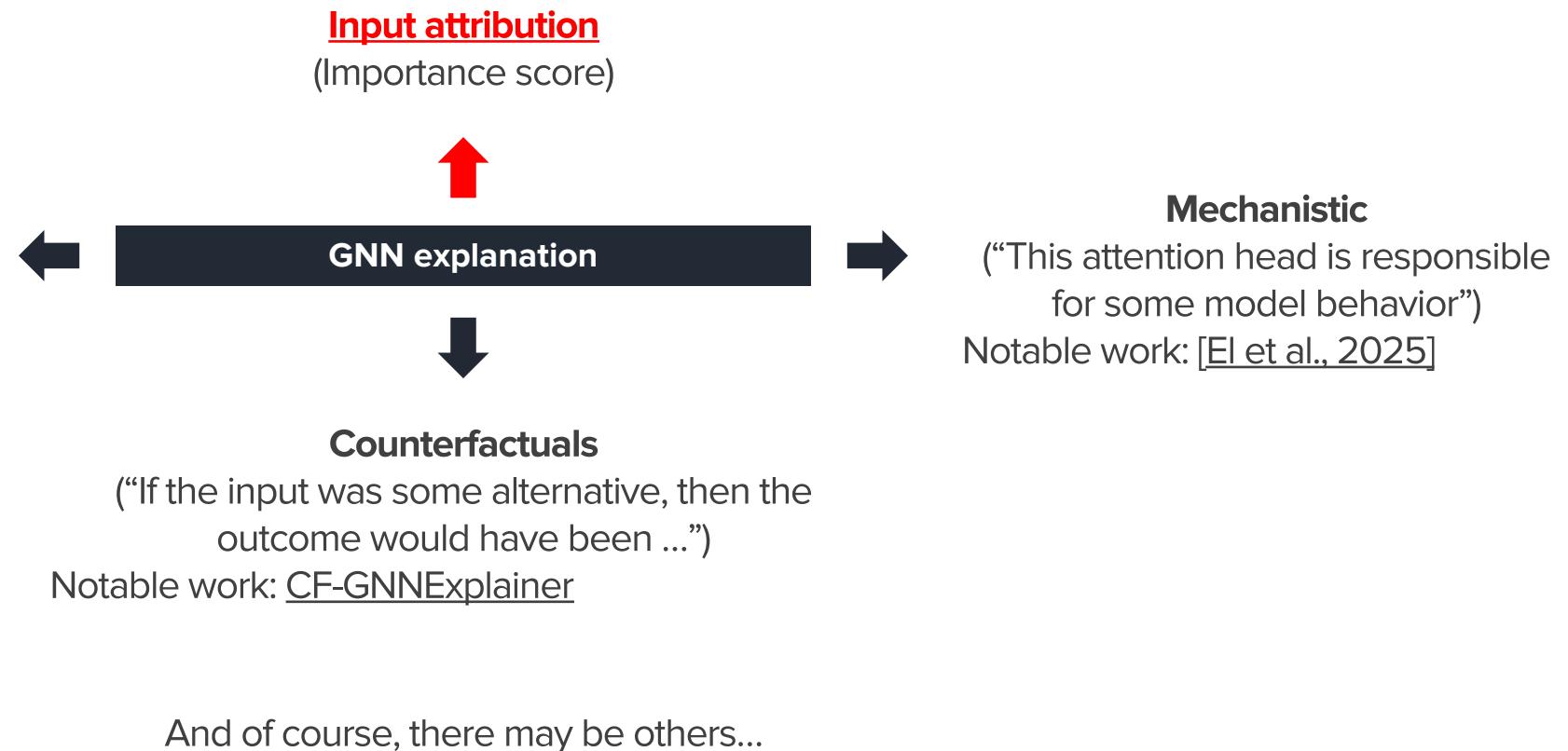
Basic scenario: When to explain? (*Post-hoc explanations)



What does it mean to explain GNN models?

Explaining GNNs mean providing additional information on the decision process in a **human-comprehensible way**

This implies: HOW to explain is up to the designer's choice.



And of course, there may be others...

Extension: Input attribution of GNN models

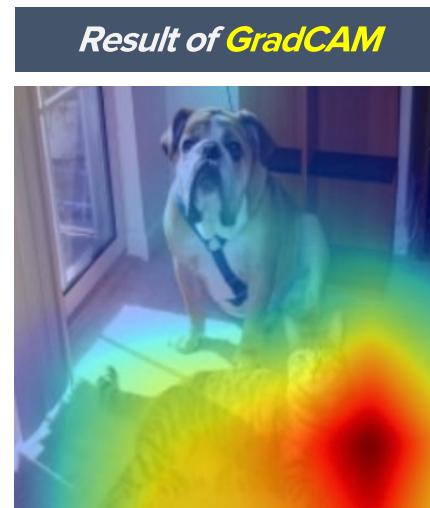
What does it mean to explain GNN models via assigning importance scores?

Attribution maps are one of the most popular ways, especially in CV and NLP.

*DTD [Montavon 2017], LRP [Bach 2018],
LIME [Ribeiro 2016], GradCAM [Selvaraju 2017], ...*



Output:
“Cat”

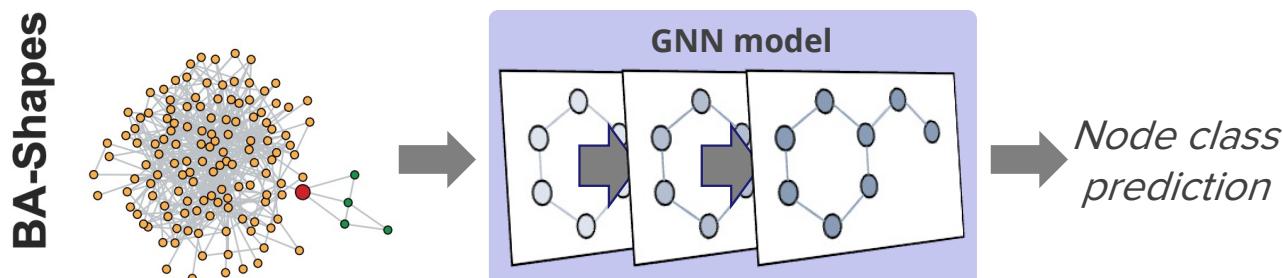


Highlights relevant pixels

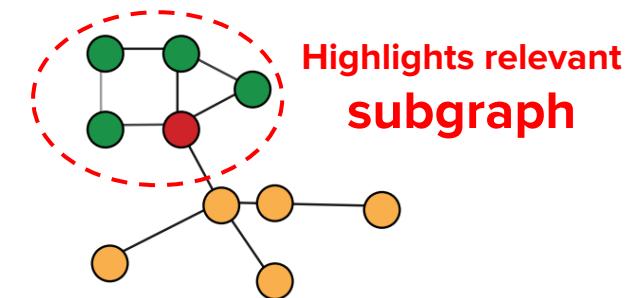
✓ Similar approaches are also popular in **GNN explanations**, too.

*GNNExplainer [Ying 2019],
PGEExplainer [Luo 2020], FastDnX [Pereira 2023]...*

Computation graph



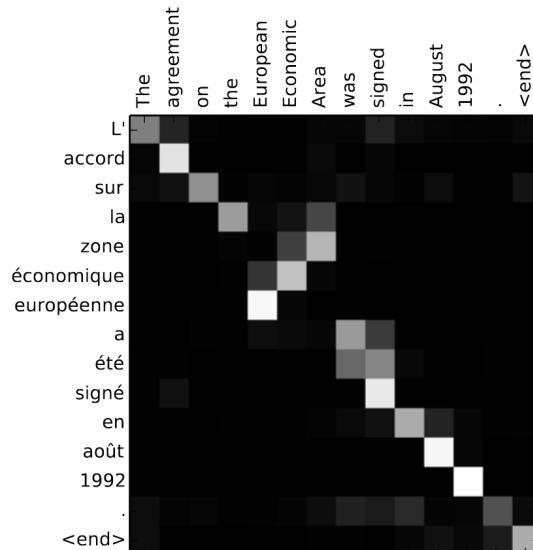
Result of Explanation



Sub-topic: Can we use attention to explain GNNs?

Motivation: Un-answered question of attention in the GNN literature

Attention as an explanation has been extensively studied in the CV & NLP literature,
due to their natural interpretation and the universal usage of transformers.



(Bahdanau et al., 2015)



(Chefer et al., 2021)

Question. How much is attention **adequate as explanation?**

Attention is not Explanation

Attention is not not Explanation

Is Attention Explanation? An Introduction to the Debate

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Vargas, Thomas François* and Patrick Watrin
CENTAL, IL&C, University of Louvain,
{adrien.bibal, remi.cardon, david.alfter, thomas.francois, patrick.watrin}@uclouvain.be

greffe*
ve Computing
Yuval Pinter*
School of Interactive Computing

How Much Does Attention Actually Attend? Questioning the Importance of Attention in Pretrained Transformers

Michael Hassid[◇] Hao Peng^{◇*} Daniel Rotem[◇] Jungo Kasai[◆] Ivan Montero^{★*}
Noah A. Smith^{◆◇} Roy Schwartz[◇]
[◇]School of Computer Science & Engineering, Hebrew University of Jerusalem
[◆]Allen Institute for Artificial Intelligence [★]Apple, Inc.
[◆]Paul G. Allen School of Computer Science & Engineering, University of Washington
{michael.hassid, daniel.rotem, roy.schwartz1}@mail.huji.ac.il
haop@allenai.org {jkasai, nasmith}@cs.washington.edu ivamon@apple.com

Question. How to generate **better attention heatmaps** in transformers?

Quantifying Attention Flow in Transformers

Samira Abnar
ILLC, University of Amsterdam
s.abnar@uva.nl

Transformer Interpretability Beyond Attention Visualization

Hila Chefer¹ Shir Gur¹ Lior Wolf^{1,2}
Science, Tel Aviv University
Research (FAIR)

Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers

Hila Chefer¹ Shir Gur¹ Lior Wolf^{1,2}
¹The School of Computer Science, Tel Aviv University
Facebook AI Research (FAIR)

Motivation: Un-answered question of attention in the GNN literature

Attention as an explanation also has the natural appeal of being a **white-box method**, since we just need to post-process the attention weights

1. Acquire attention weights from the pre-trained model



2. Simply apply further calculations

Quantifying Attention Flow in Transformers

Samira Abnar
ILLC, University of Amsterdam
s.abnar@uva.nl

Willem Zuidema
ILLC, University of Amsterdam
w.h.zuidema@uva.nl

Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers

Hila Chefer¹ Shir Gur¹ Lior Wolf^{1,2}
¹The School of Computer Science, Tel Aviv University
²Facebook AI Research (FAIR)

Transformer Interpretability Beyond Attention Visualization

Hila Chefer¹ Shir Gur¹ Lior Wolf^{1,2}
¹The School of Computer Science, Tel Aviv University
²Facebook AI Research (FAIR)

$$\hat{\mathbf{A}}^{(b)} = I + \mathbb{E}_h \mathbf{A}^{(b)}$$

$$\text{rollout} = \hat{\mathbf{A}}^{(1)} \cdot \hat{\mathbf{A}}^{(2)} \cdot \dots \cdot \hat{\mathbf{A}}^{(B)}$$

$$\bar{\mathbf{A}} = \mathbb{E}_h((\nabla \mathbf{A} \odot \mathbf{R}^{\mathbf{A}})^+)$$

$$\bar{\mathbf{A}}^{(b)} = I + \mathbb{E}_h(\nabla \mathbf{A}^{(b)} \odot R^{(n_b)})^+$$

$$\mathbf{C} = \bar{\mathbf{A}}^{(1)} \cdot \bar{\mathbf{A}}^{(2)} \cdot \dots \cdot \bar{\mathbf{A}}^{(B)}$$

Note that all calculations are explicit, interpretable, and computed deterministically.

Question: Is there any similar work for graph **attention** network type models?

Core question

Q1. Are attention explanations for attention-based GNNs?

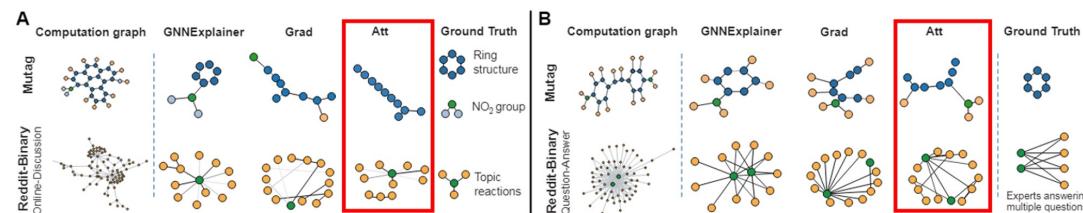
Q2. What methods have been developed to produce better attribution from attention in GNNs?

...Both questions are **not properly answerable**, since **attention-based GNN models** were only used as a **naïve baseline** in the literature.

Example 1)

		Node-level tasks				Edge-level tasks			
		BA-Shapes		BA-Community		Tree-Grid			
		GCN	MPNN	GraphNets	GAT	GCN	MPNN	GraphNets	GAT
Random Baseline		0.27	0.27	0.27	0.27	0.38	0.38	0.38	0.38
GradInput		0.58	0.64	0.39	0.72	0.52	0.51	0.5	0.5
SmoothGrad(GI)		0.58	0.64	0.39	0.72	0.52	0.51	0.51	0.49
GradCAM-last		0.79	0.84	0.86	0.8	0.7	0.67	0.68	0.61
GradCAM-all		0.67	0.78	0.65	0.76	0.67	0.71	0.73	0.67
IG	--	--	--	--	--	0.81	0.75	0.72	0.62
Attention Weights	--	--	--	0.5	--	--	0.5	--	--

Example 2)



Example 3)

Explanation AUC					
GRAD	0.882	0.750	0.905	0.612	0.717
ATT	0.815	0.739	0.824	0.667	0.674
Gradient	-	-	-	-	0.773
GNNExplainer	0.925	0.836	0.948	0.875	0.742
PGExplainer	0.963±0.011	0.945±0.019	0.987±0.007	0.907±0.014	0.926±0.021
Improve	4.1%	13.0%	4.1%	3.7%	24.7%

Inference Time (ms)					
GNNExplainer	650.60	696.61	690.13	713.40	934.72
PGExplainer	10.92	24.07	6.36	6.72	80.13
Speed-up	59x	29x	108x	106x	12x

Problem: Can we calculate a more faithful and accurate explanation using attention weights from graph attention network types?

GNN-XAI evaluation

(Sanchez-Lengeling et al., 2020)

“...have several blocks and attention heads, so for each component we take their average to combine them to a scalar value assigned to each edge.”

GNNExplainer
(Ying et al., NeurIPS 2019)

“...it is not obvious which attention weights need to be used for edge importance, Each edge’s importance is thus computed as the average attention weight across all layers.”

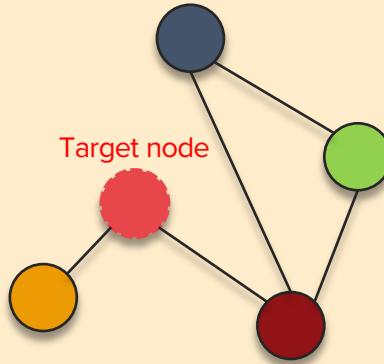
PGExplainer
(Luo et al., NeurIPS 2020)

“Each edge’s importance is obtained by averaging its attention weights across all attention layers.”

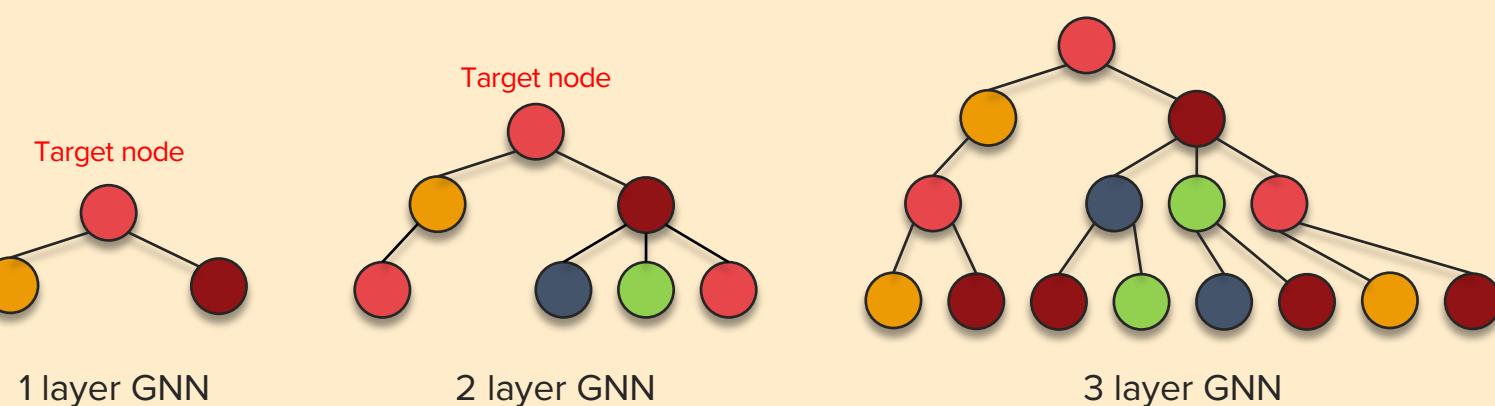
Our solution: Switch to the computation tree viewpoint!

We found that attention weights reliably represent edge importance after post-calculations **based on the computation tree**.

What is a ‘computation tree’?



Input graph



Computation tree of the target node with different GNN layers

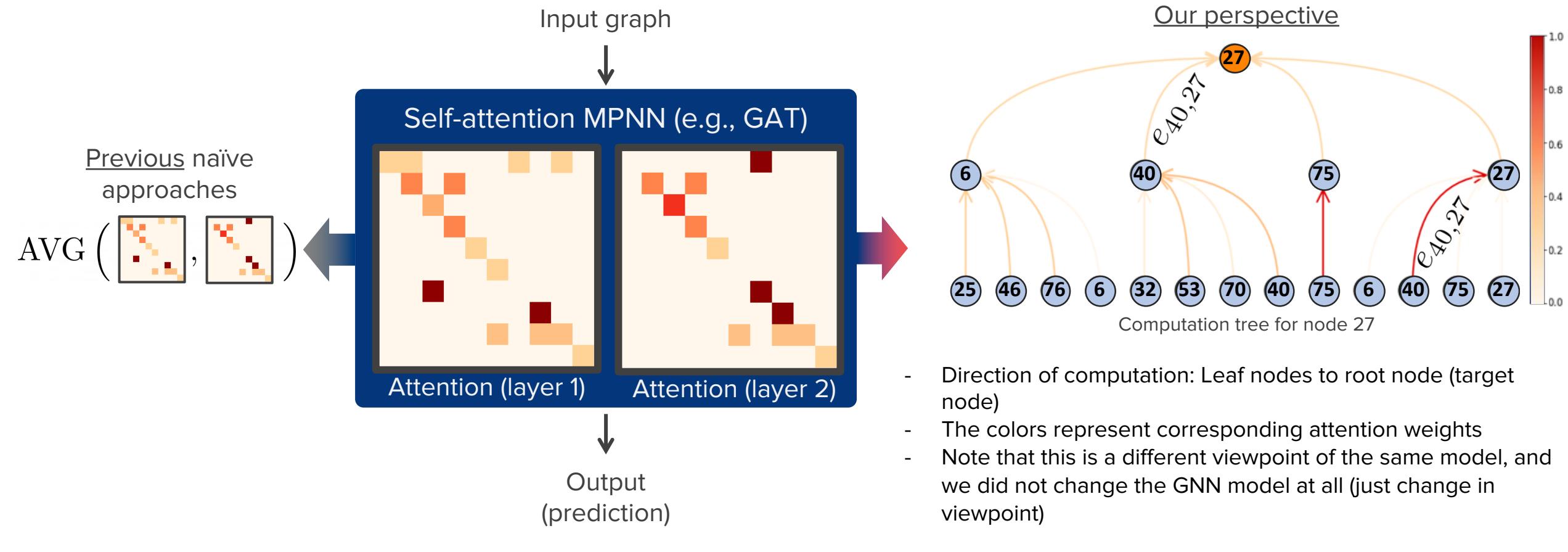
$$\text{Representation of node } u \quad \mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v) \right)$$

- \oplus : Permutation invariant operator (e.g., sum)
- ϕ : Combine function (e.g., small neural network)
- ψ : Message function (e.g., scaling function)
- \mathcal{N}_u : Set of neighbors of node u

- Due to the aggregation-based design of GNNs, it is often beneficial to visualize how the information flows as a **computation tree**.

Our solution: Switch to the computation tree viewpoint!

We found that attention weights reliably represent edge importance after post-calculations **based on the computation tree**.

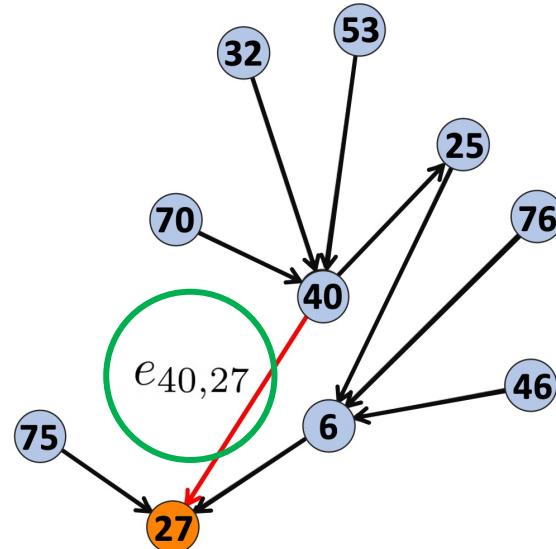


Our solution: Switch to the computation tree viewpoint!

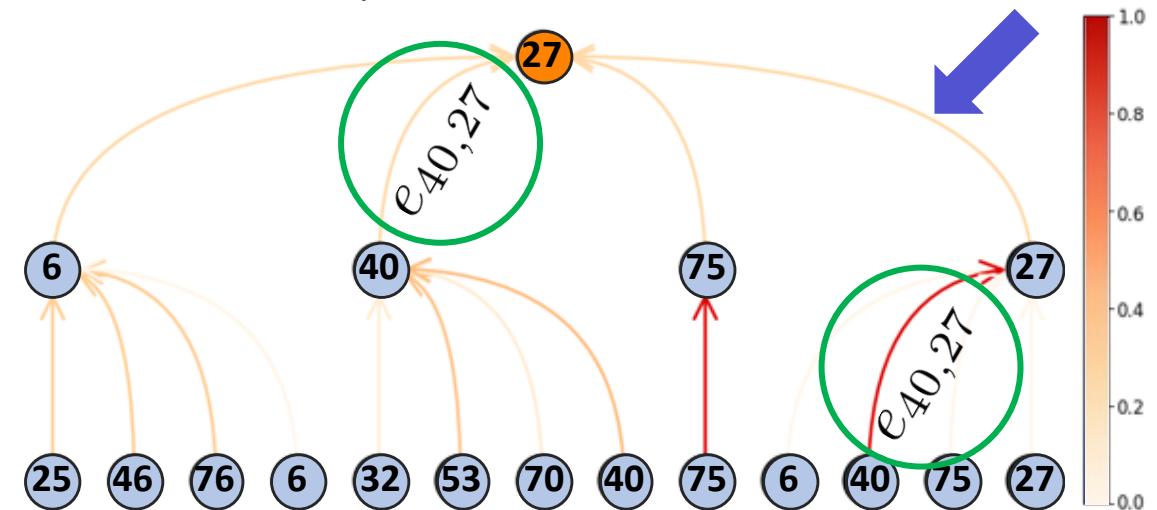
We found that attention weights reliably represent edge importance after post-calculations **based on the computation tree**.

Two design principles: **Summation** and **adjustment**

2-hop neighbor of node 27



Computation tree for node 27



Two observations

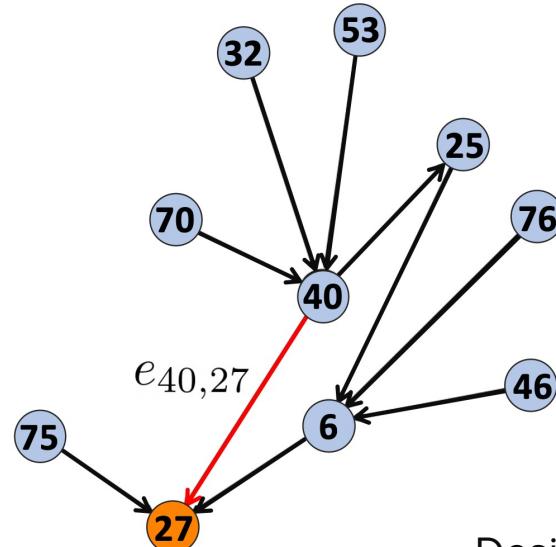
- Proximity effect: **Edges can appear multiple times**, and (likely to be) related with proximity.
- Contribution adjustment: The contribution of an edge in the computation tree should be **adjusted by its position**.

Our solution: Switch to the computation tree viewpoint!

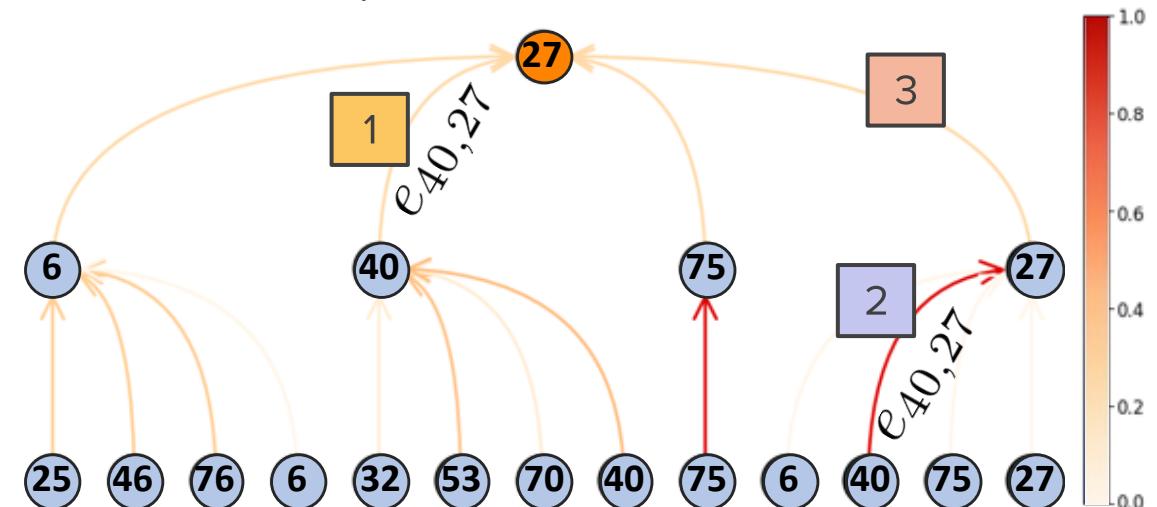
We found that attention weights reliably represent edge importance after post-calculations **based on the computation tree**.

Two design principles: **Summation** and **adjustment**

2-hop neighbor of node 27



Computation tree for node 27



Design philosophy of GAtt from our observations

- Proximity effect: **Need to sum all occurrences of an edge!** (Averaging will offset the number of appearances)
- Contribution adjustment: Each attention should be multiplied by all attention weights along the path towards the root.

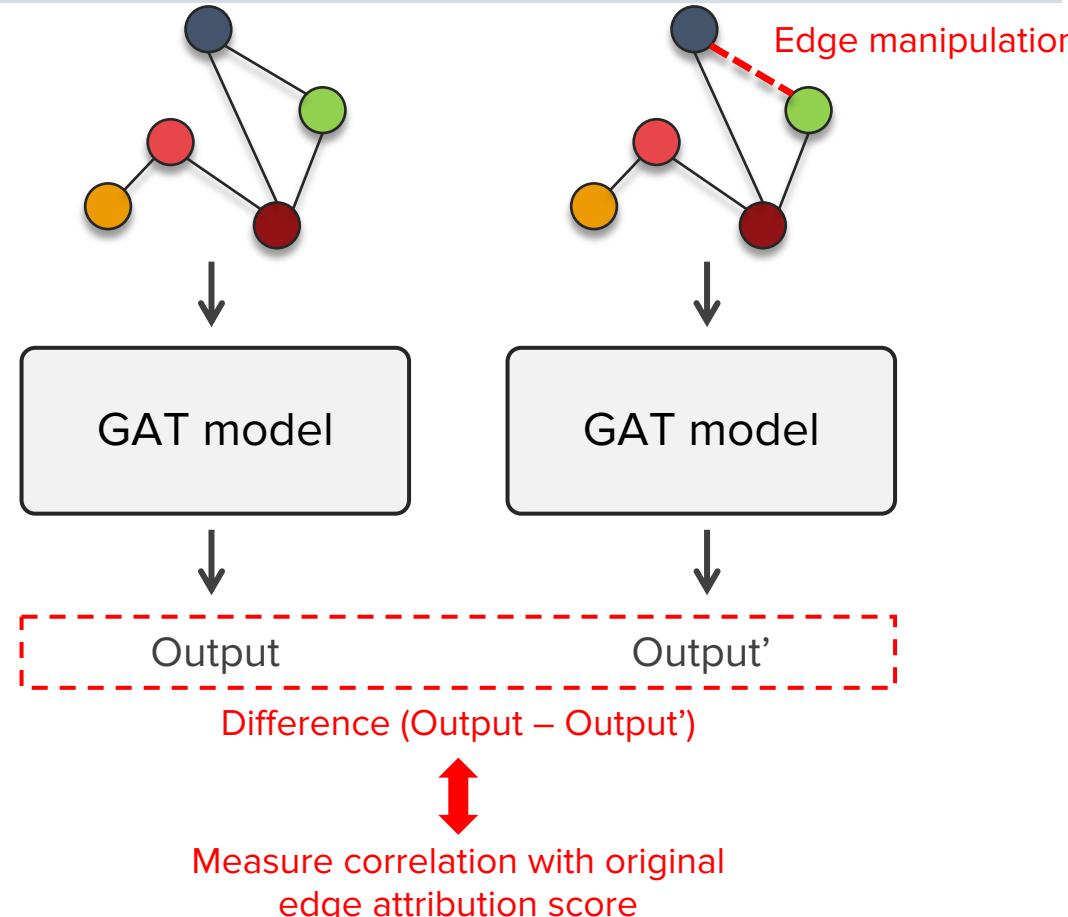
Attribution of edge (40, 27)
when target node is 27

$$= \boxed{1} + \boxed{3} \times \boxed{2}$$

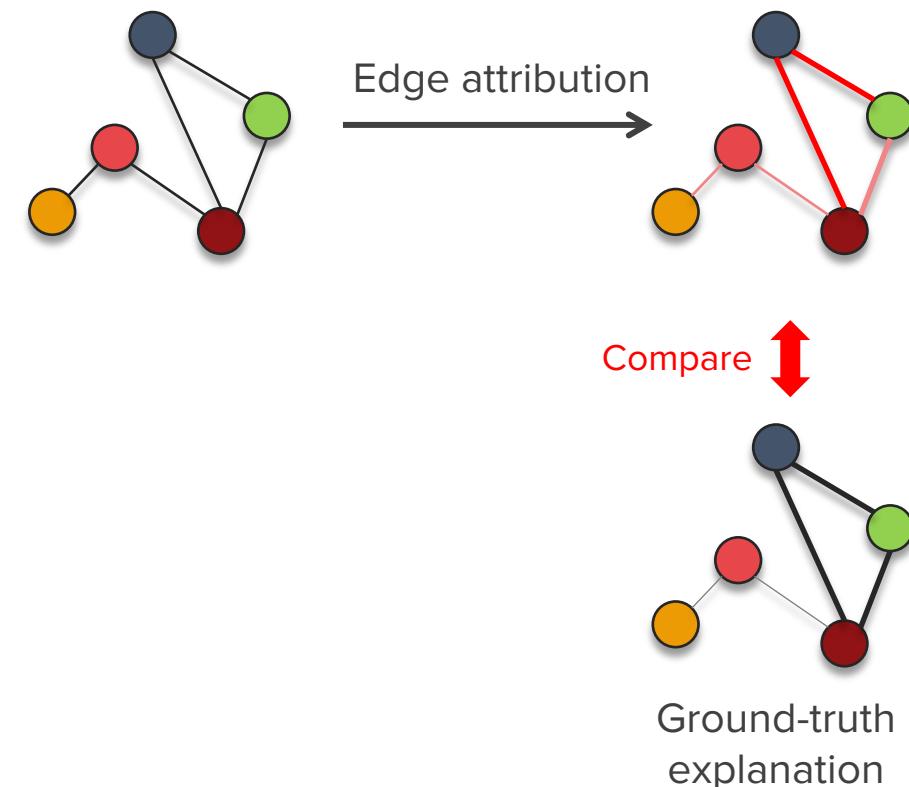
Experimental results (Evaluation metrics)

Faithfulness and Accuracy

Faithfulness: How much does the edge attribution truly reflect the model's behavior?



Accuracy: How much does the edge attribution highlight ground-truth explanations?



Intuition: If the edge was truly important, the model should drastically change its output when deleted.

Experimental results (Faithfulness)

Faithfulness experiments on real-world datasets shows the superiority of our method

Dataset	2-layer GAT/GATv2			3-layer GAT/GATv2		
	GATT	AVGATT	Random	GATT	AVGATT	Random
Cora	Δ_{PC} 0.8468/0.1040	0.1764/0.0121	-0.0056/-0.0036	Δ_{PC} 0.8642/0.1696	0.0967/0.0168	0.0045/0.0045
	Δ_{NE} 0.7112/0.0930	0.1526/0.0100	-0.0076/0.0019	Δ_{NE} 0.7690/0.1664	0.0859/0.0186	0.0040/0.0037
	Δ_P 0.9755/0.9623	0.7251/0.6226	0.4389/0.4891	Δ_P 0.9875/0.9966	0.7075/0.8897	0.5235/0.6107
Citeseer	Δ_{PC} 0.8516/0.0658	0.3096/0.0180	0.0012/-0.0043	Δ_{PC} 0.8711/0.0432	0.2110/0.0107	-0.0073/-0.0034
	Δ_{NE} 0.7653/0.0700	0.2780/0.0186	0.0021/0.0019	Δ_{NE} 0.8291/0.0551	0.2006/0.0140	0.0015/0.0025
	Δ_P 0.9846/0.9771	0.9213/0.9510	0.3695/0.4258	Δ_P 0.9920/0.9961	0.8979/0.9692	0.4039/0.7569
Pubmed	Δ_{PC} 0.8812/0.0631	0.1648/0.0126	-0.0064/0.0021	Δ_{PC} 0.8489/0.0367	0.0592/0.0023	0.0015/-0.0016
	Δ_{NE} 0.8201/0.0915	0.1477/0.0169	-0.0068/0.0078	Δ_{NE} 0.8612/0.0484	0.0600/0.0028	0.0009/-0.0015
	Δ_P 0.9915/0.9972	0.8834/0.9361	0.3974/0.1327	Δ_P 0.9993/0.9996	0.8932/0.9153	0.5172/0.5242
Arxiv	Δ_{PC} 0.7790/0.0546	0.0794/-0.0593	0.0007/0.0028	Δ_{PC} 0.7721/0.0508	0.0465/-0.0252	-0.0004/-0.0003
	Δ_{NE} 0.8287/0.0164	0.0804/-0.0390	0.0016/-0.0067	Δ_{NE} 0.8282/-0.0012	0.0478/-0.0216	-0.0017/ 0.0000
	Δ_P 0.9908/0.8995	0.8470/0.2560	0.4962/0.5107	Δ_P 0.9985/0.9366	0.8331/0.3934	0.5004/0.5034
Cornell	Δ_{PC} 0.8089/0.2660	0.3391/0.0209	-0.0284/0.0421	Δ_{PC} 0.7173/0.0899	0.3065/-0.0512	-0.0273/-0.0129
	Δ_{NE} 0.7820/0.1526	0.3199/-0.0488	-0.0231/0.0235	Δ_{NE} 0.7160/0.0520	0.3491/-0.0294	-0.0060/-0.0017
	Δ_P 0.9532/0.8372	0.7416/0.5130	0.5074/0.5660	Δ_P 0.9270/0.6406	0.6907/0.3969	0.4787/0.4953
Texas	Δ_{PC} 0.7818/0.0801	0.3676/-0.0406	-0.0762/0.0025	Δ_{PC} 0.6866/0.1504	0.2443/0.0486	0.0414/0.0040
	Δ_{NE} 0.7977/0.1443	0.3809/ 0.1478	-0.0659/0.0145	Δ_{NE} 0.6132/0.0896	0.1645/0.0579	0.0202/0.0149
	Δ_P 0.8726/0.7299	0.6803/0.3669	0.4733/0.5198	Δ_P 0.9197/0.8195	0.7072/0.5565	0.5562/0.5426
Wisconsin	Δ_{PC} 0.6898/0.1751	0.2649/0.0556	0.0596/0.0120	Δ_{PC} 0.7616/0.0323	0.2443/0.0486	0.0414/0.0040
	Δ_{NE} 0.6421/0.1554	0.2340/0.0636	0.0414/0.0157	Δ_{NE} 0.7409/0.0243	0.1645/0.0579	0.0202/0.0149
	Δ_P 0.8985/0.8501	0.7067/0.6060	0.5427/0.5006	Δ_P 0.8982/0.7582	0.6906/0.3980	0.5119/0.5333

Higher the better!

- We compared our method against the naïve baseline where the attention matrices are averaged across layers (i.e., AvgAtt, see left figure)
- All results show our method (GAtt) outperforms all baselines in all 7 datasets on GAT (Veličković et al., 2017), GATv2 (Brody et al., 2022), and SuperGAT (Kim et al., 2021) (shown in paper).

Experimental results (Accuracy)

Accuracy experiments on real-world datasets shows the superiority of our method

Higher the better!

Naïve attention-based baseline

Although a different category, we expanded the list of baselines to include 7 other non-attention-based XAI methods

Model	Dataset	GATT	AvgATT	SA	GB	IG	GNNEEx	PGEEx	GM	FDnX	Random
GAT	BA-Shapes	0.9591	0.7977	0.9563	0.6231	0.6231	0.8916	0.8289	0.5316	0.9917	0.4975
	Infection	0.9976	0.8786	0.8237	0.8949	0.9472	0.9272	0.7173	0.6859	0.6574	0.4811
GATv2	BA-Shapes	0.9617	0.7876	0.9626	0.5260	0.5232	0.9318	0.5000	0.5123	0.9923	0.4976
	Infection	0.8628	0.4719	0.7711	0.7250	0.7849	0.7611	0.8178	0.5355	0.5059	0.5002

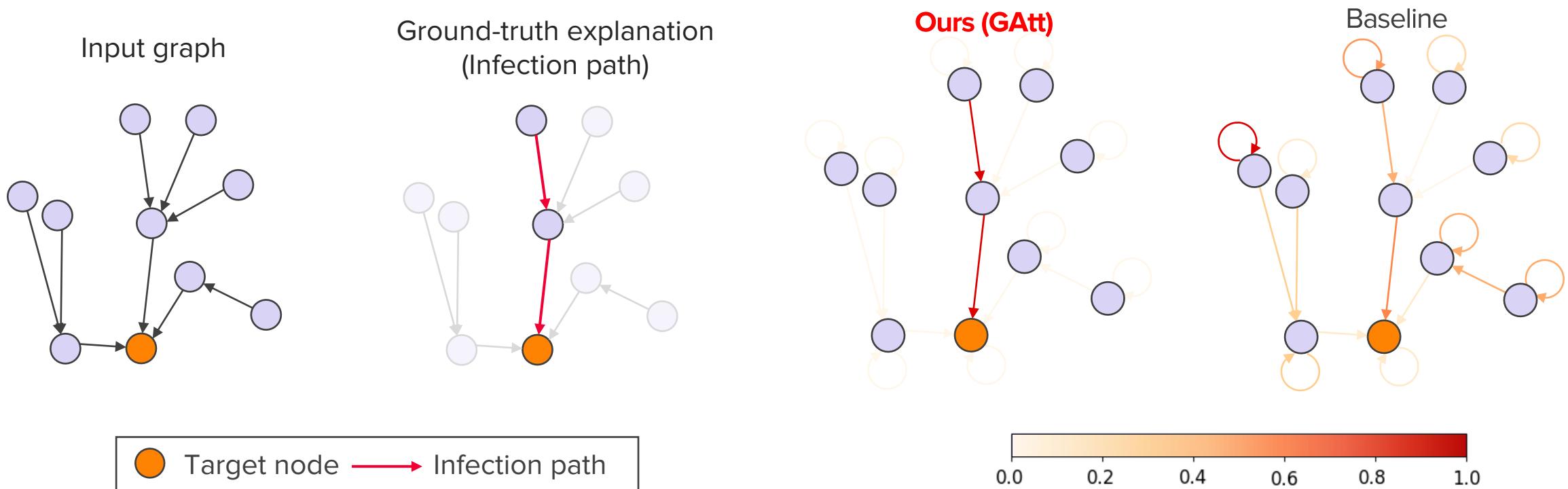
All results show our method (GAtt) outperforms all 9 baselines in terms of explanation accuracy.

Short description of other XAI methods

- SA (Saliency): Gradient-based explanation (Simonyan et al., 2014)
- GB (Guided Backpropagation): Propagate output signals back to the input according to model activations (Springenberg et al., 2015)
- IG (Integrated Gradients): Numerical integration of gradients from a baseline to the actual input (Sundararajan et al., 2017)
- GNNEEx (GNNEExplainer): Optimize edge masks using a mutual-information based loss function with gradient descent (Ying et al., 2019)
- PGEEx (PGEExplainer): Train a neural network using the loss function from GNNEExplainer (Luo et al., 2020)
- GM (GraphMask): Train a classifier that masks certain messages in the GNN that does not change the output (Schlichtkrull et al., 2021)
- FDnX (FastDnX): Train a simpler surrogate GNN, and use that GNN for explanation (Pereira et al., 2021)

Experimental results (Visualizations)

Case study reveals the model highlights ground-truth explanations when using GAtt
(Infection dataset)



Takeaways

1. Understanding explainable AI
 1. Why? Black-box nature, serious application, model debugging
 2. Types of XAI: Attribution is the basic form of explanation (with a touch of Mech. Interp. + Shapley)
2. Extension to graph learning: The basic concepts can naturally be extended to graphs
3. Subtopic: Can we explain GNNs with attention? (Yes, but with some effort of course)

Thank you!

Please feel free to ask any questions :)

jordan7186.github.io