A Practical Introduction to (Explainable) Graph Learning

Yong-Min Shin
School of Mathematics and Computing (Computational Science and Engineering)
Yonsei University
2025.04.02

수학계산학부(계산과학공학)
School of Mathematics and Computing
(Computational Science and Engineering)

이화여자대학교
EWHA WOMANS UNIVERSITY

Objectives

Part 1: A practical introduction to graphs and graph neural networks

1. Understanding of **graphs** as a **general data type**
2. Understanding of the general framework of **graph neural networks (GNNs)**

Part 2: Towards explainable graph learning with attention

1. Understanding the basic concepts of **explainable AI**
2. **Answer to the question: Can we understand graph attention networks using attention?**

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

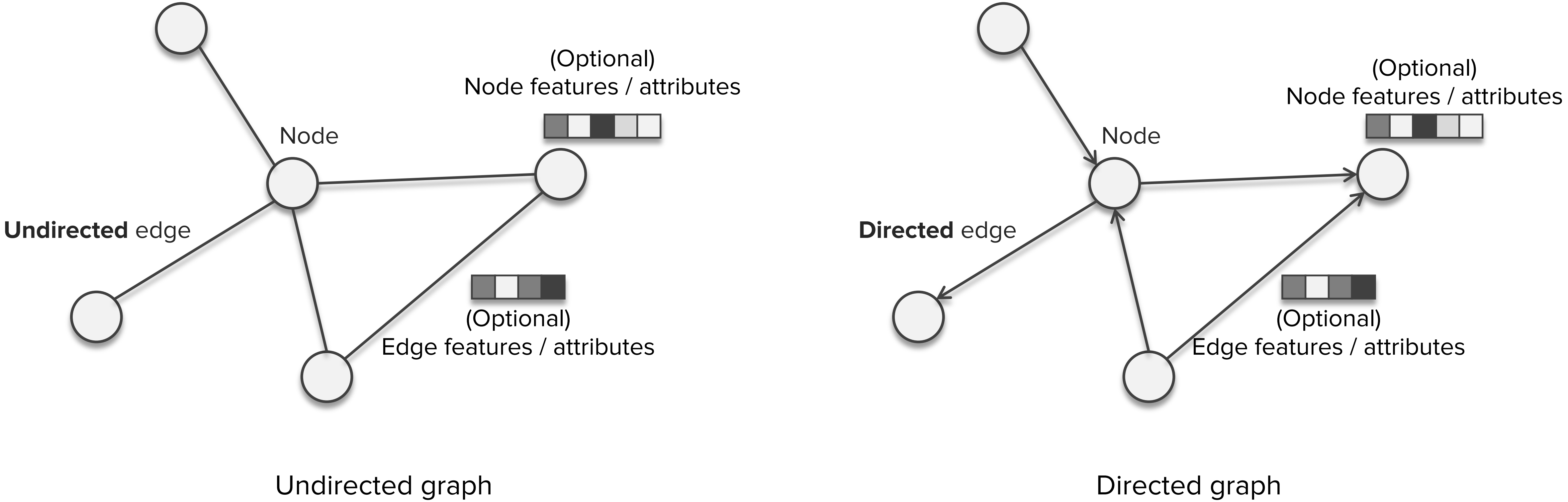
Part 1: A practical introduction to graphs and graph neural networks

Understanding of graphs as a general data type

*This part is heavily influenced by one of my academic heros, Petar Veličković. These are some materials from his public materials that I have referred to:
- (Slide) Everything is Connected: Graph Neural Networks from the Ground Up (2021)
- (Blog) Graph & Geometric ML in 2024: Where We Are and What's Next (Part II – Applications)

Graphs as an abstract datatype

Graphs are an abstract type of data where nodes (entities) are **connected** by edges (connections)



…But honestly, looking at this does not result in a **practical** understanding of graphs.

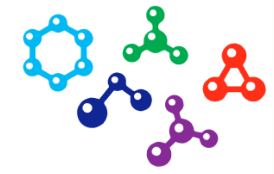
Therefore, we will look at **various applications** in the field of **graph machine learning** before moving our discussion further.

Area 1) Biology & Chemistry Research

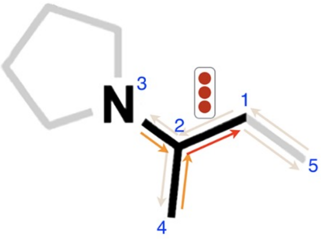
Example 1: The discovery of Halicin, GNN-guided antibiotic discovery

Node: Atom
Node feature: Atom type
Edge: Atom bonds
Edge feature: Bond type etc.

Training GNNs with real-world chemical dataset



Training set
(10^4 molecules)



Prediction task
E.coli growth inhibition

Apply

Drug dataset (~6K)

Acquire potential lead compounds that acts a gainst E.coli

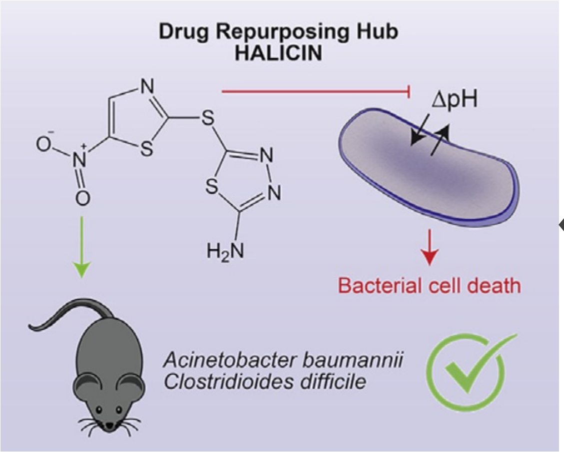
Further screening
& Empirical testing


Cell
Article

A Deep Learning Approach to Antibiotic Discovery

Jonathan M. Stokes,[1,2,3] Kevin Yang,[3,4,10] Kyle Swanson,[3,4,10] Wengong Jin,[3,4] Andres Cubillos-Ruiz,[1,2,5] Nina M. Donghia,[1,5] Craig R. MacNair,[6] Shawn French,[6] Lindsey A. Carfrae,[6] Zohar Bloom-Ackermann,[2,7] Victoria M. Tran,[2] Anush Chiappino-Pepe,[5,7] Ahmed H. Badran,[2] Ian W. Andrews,[1,2,5] Emma J. Chory,[1,2] George M. Church,[5,7,8] Eric D. Brown,[6] Tommi S. Jaakkola,[3,4] Regina Barzilay,[3,4,9,*] and James J. Collins[1,2,5,8,9,11,*]

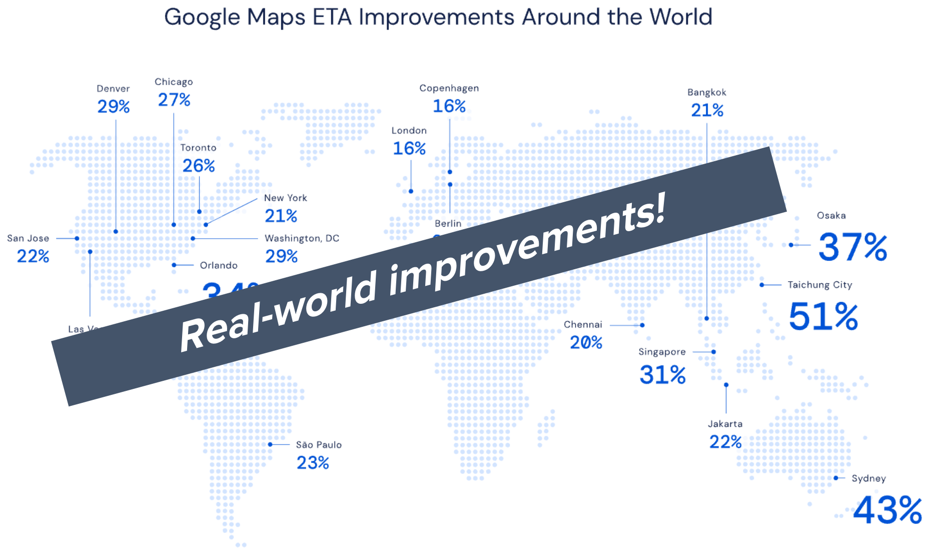
[1]Department of Biological Engineering, Synthetic Biology Center, Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA
[3]Machine Learning for Pharmaceutical Discovery and Synthesis Consortium, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[4]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[5]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA 02115, USA
[6]Department of Biochemistry and Biomedical Sciences, Michael G. DeGroote Institute for Infectious Disease Research, McMaster University, Hamilton, ON L8N 3Z5, Canada
[7]Department of Genetics, Harvard Medical School, Boston, MA 02115, USA
[8]Harvard-MIT Program in Health Sciences and Technology, Cambridge, MA 02139, USA
[9]Abdul Latif Jameel Clinic for Machine Learning in Health, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[10]These authors contributed equally
[11]Lead Contact
*Correspondence: regina@csail.mit.edu (R.B.), jimjc@mit.edu (J.J.C.)
https://doi.org/10.1016/j.cell.2020.01.021

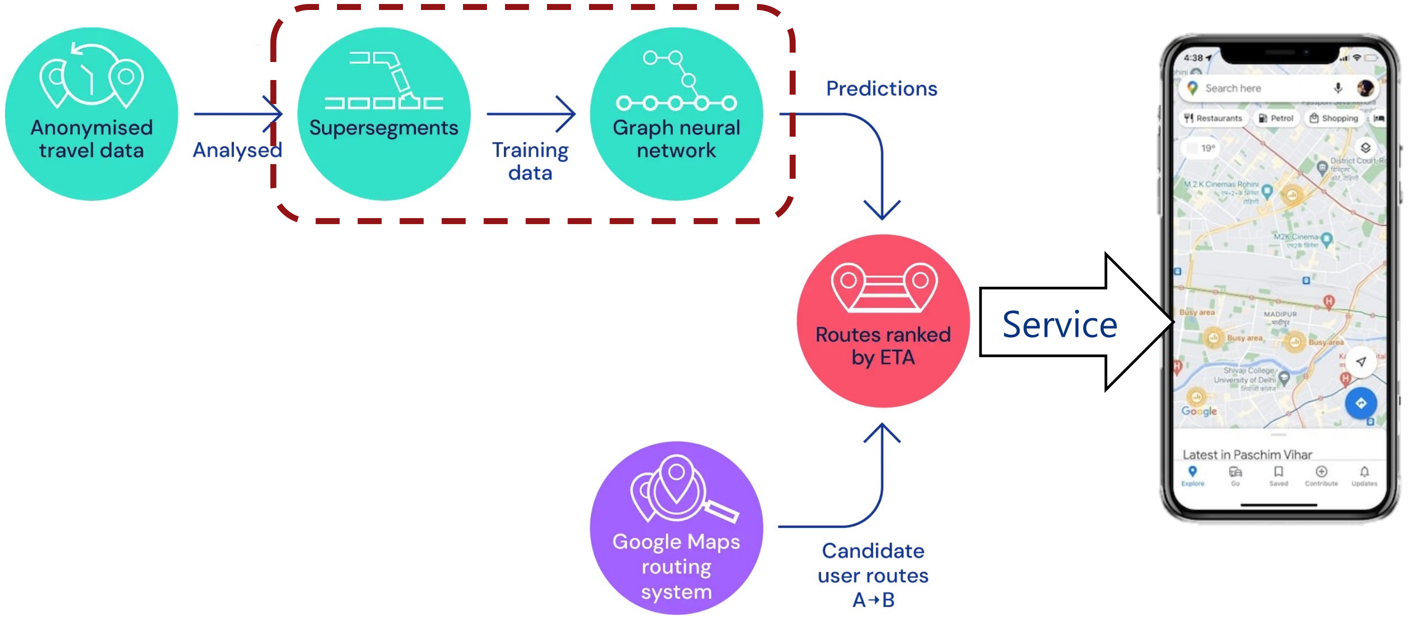


Stokes, Jonathan M., et al. "A deep learning approach to antibiotic discovery." Cell 180.4 (2020): 688-702.
Yang, Kevin, et al. "Analyzing learned molecular representations for property prediction." Journal of chemical information and modeling 59.8 (2019): 3370-3388.

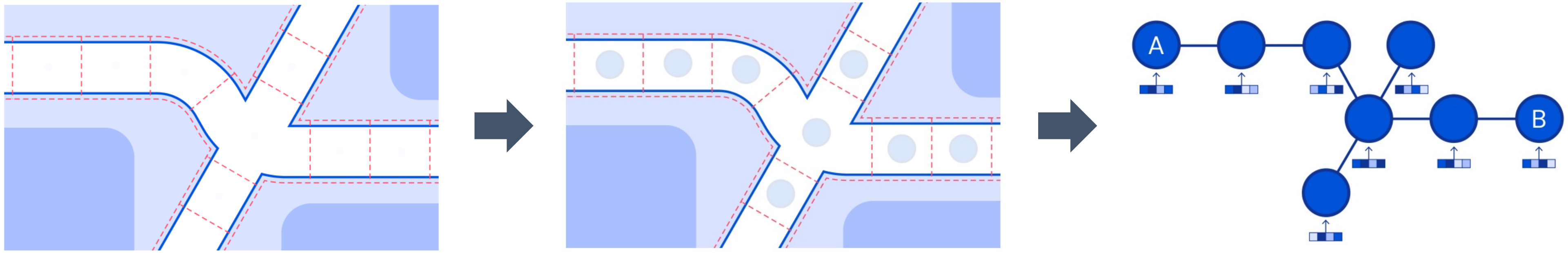
Area 2) ETA prediction

Example 2: DeepMind's improvement of Google map's ETA (Estimated Time of Arrival) prediction





Unlike chemical datasets, constructing a graph is less straightforward.
In these cases, **how to construct the graph** is also a crucial contribution.

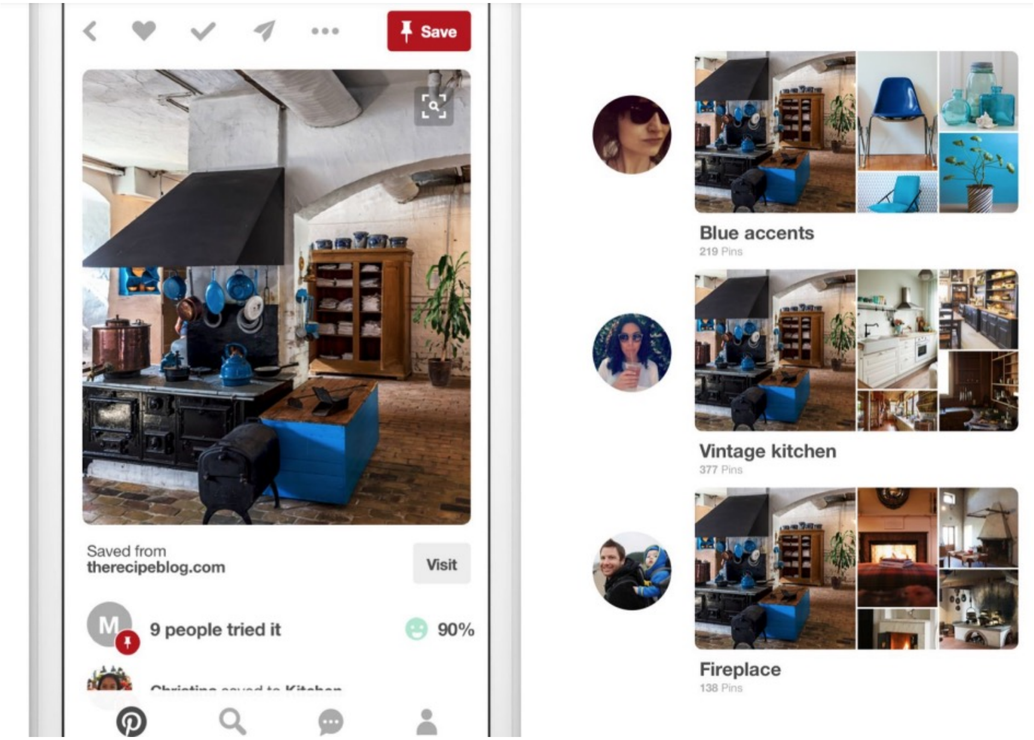


Derrow-Pinion, Austin, et al. "ETA prediction with graph neural networks in google maps.", CIKM 2021.
Deepmind, "Traffic prediction with advanced graph neural networks"

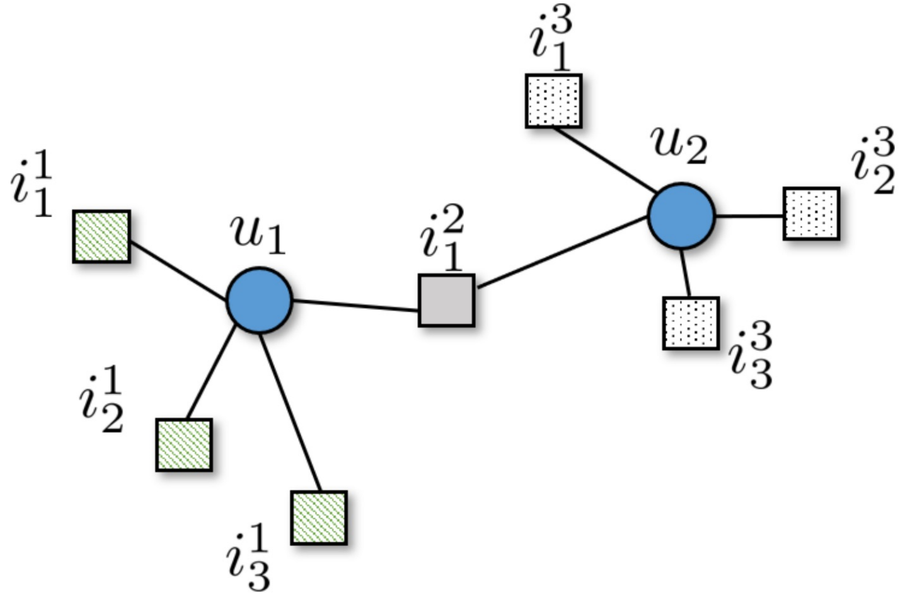
Area 3) Recommdender systems

Example 3: Pinterest (social platform)

Image & User interaction in Pinterest



User-item interaction graph



Source: Andrew Zhai (Pinterest) talk @WWW 2022 (link)
Right figure: Hou et al., Collaborative Filtering Based on Diffusion Models: Unveiling the Potential of High-Order Connectivity, SIGIR 2024

Area 3) Recommdender systems

Example 4: Other industry usecases

Applied Research Track — CIKM '20, October 19–23, 2020, Virtual Event, Ireland

P-Companion: A Principled Framework for Diversified Complementary Product Recommendation

Applied Data Science Track Paper — KDD '20, August 23–27, 2020, Virtual Event, USA

ConSTGAT: Contextual Spatial-Temporal Graph Attention Network for Travel Time Estimation at Baidu Maps

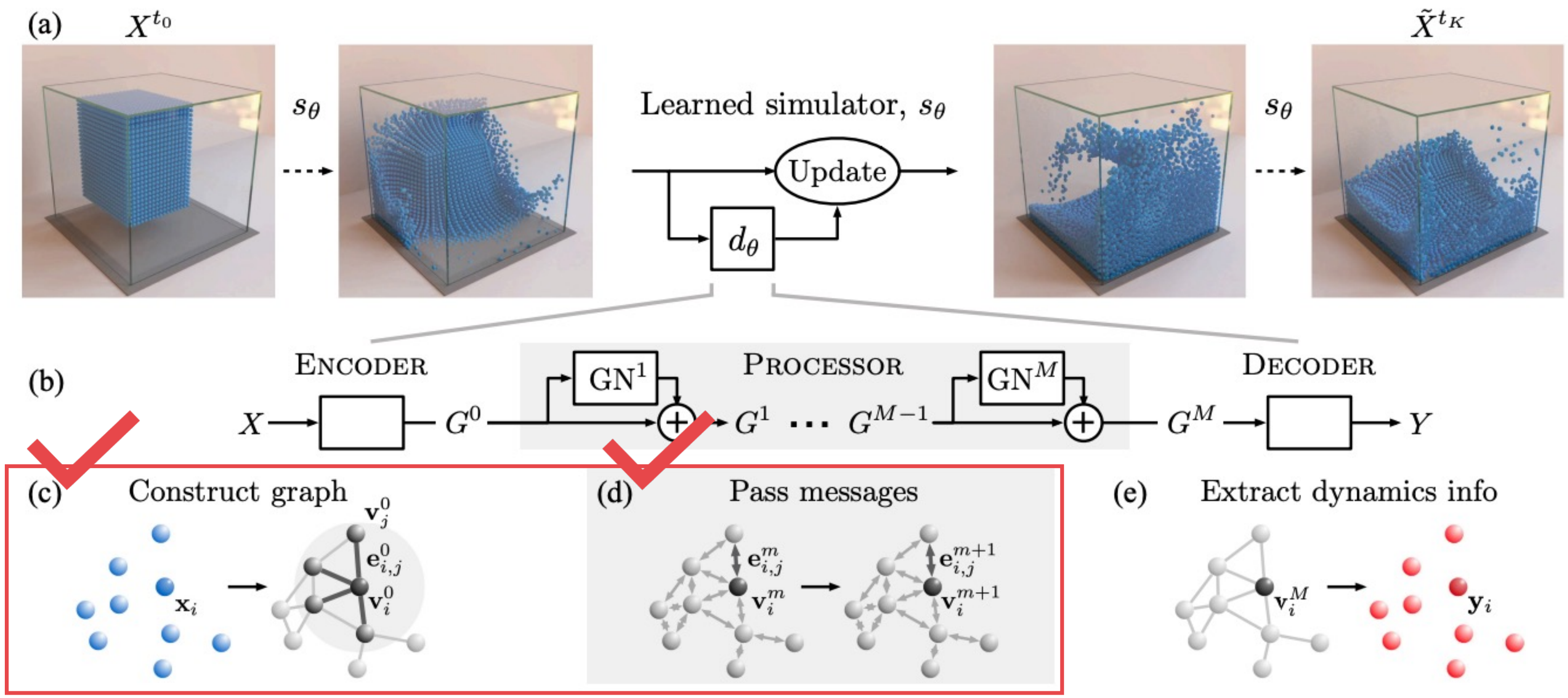
AliGraph: A Comprehensive Graph Neural Network Platform

Graph Neural Networks for Friend Ranking in Large-scale Social Platforms

Knowing your FATE: Friendship, Action and Temporal Explanations for User Engagement Prediction on Social Apps

Area 4) Modeling physical systems

Example 5: Simulation of complex physical systems

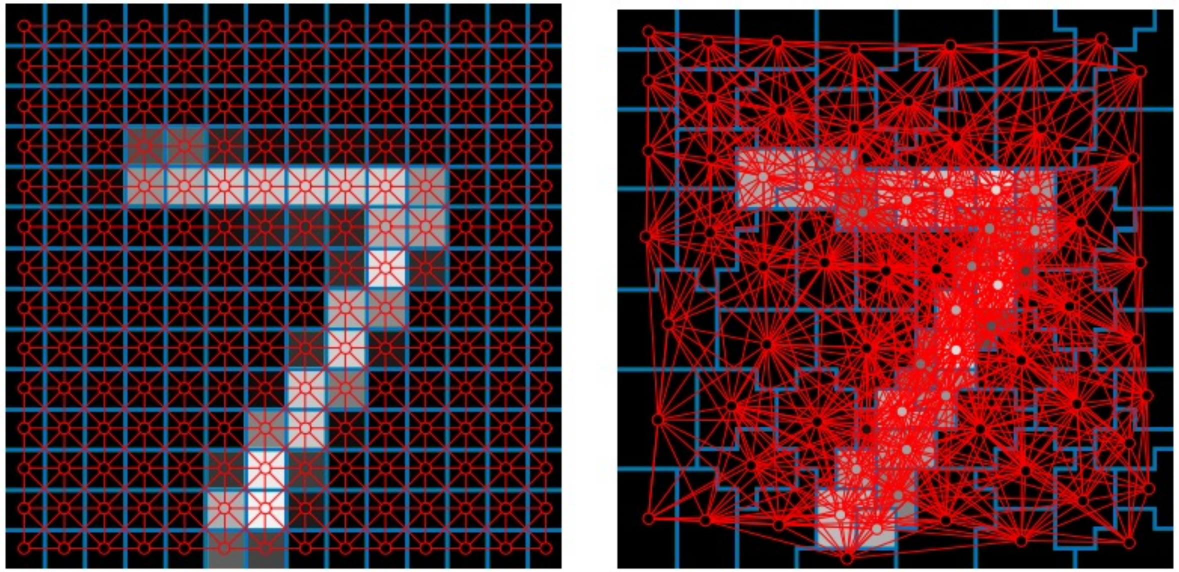


Similar to the ETA prediction task, <u>how to construct the edges between particles</u> will highly impact the rest of the learning process.

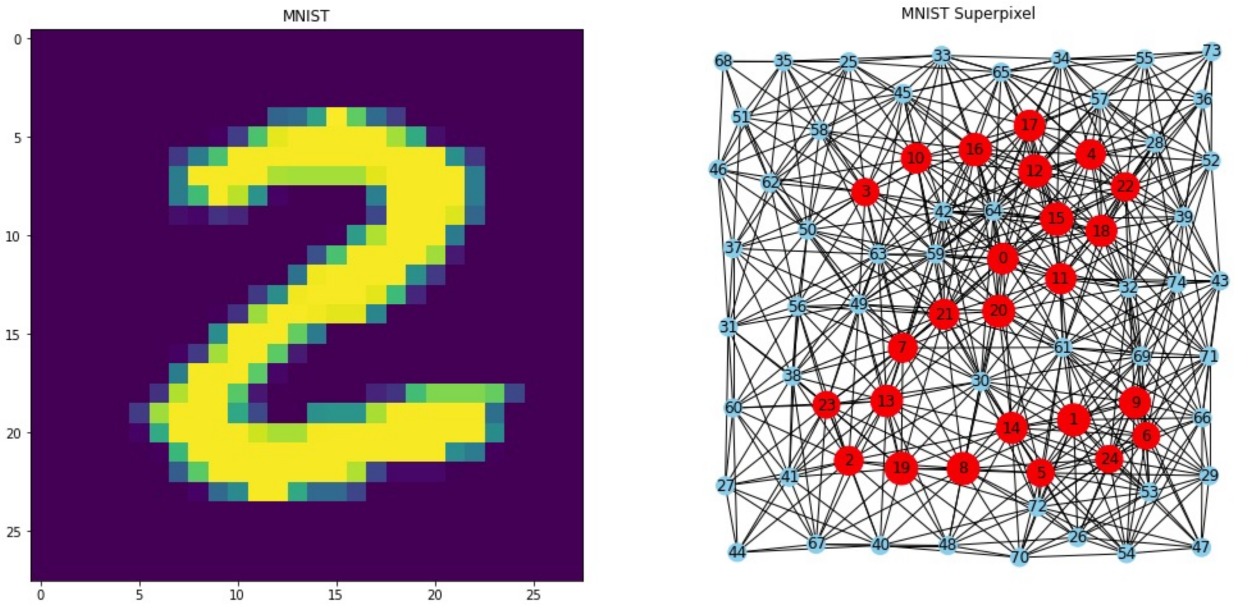
Sanchez-Gonzalez et al., Learning to Simulate Complex Physics with Graph Networks, ICML 2020

Area 5) Images are actually grid-like graphs

Example 6: MNIST and MNIST-sp



Regular grid Superpixels



MNIST-sp is quite commonly used as a benchmark dataset in the graph domain.

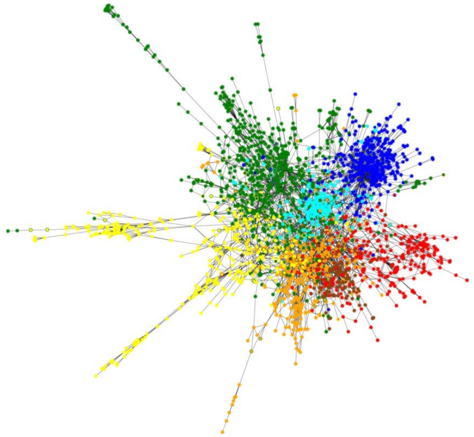
(Left) Monti et al., "Geometric deep learning on graphs and manifolds using mixture model CNNs", CVPR 2017
(Right) https://github.com/pyg-team/pytorch_geometric/issues/320

In academia: Benchmark datasets in the literature

Social

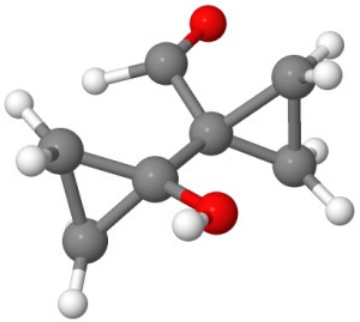
Node: People / Account
Edge: Connection
Node feature: Metadata

Citation / Web



Node: Paper
Edge: Citation
Node feature: Abstract

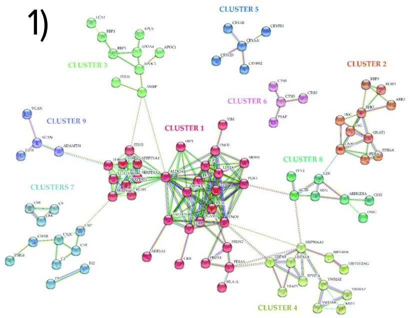
Molecules



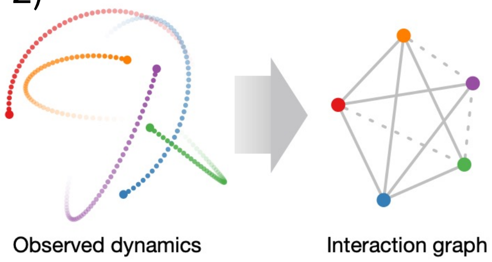
Node: Atom
Edge: Bond
Node feature: Atom type
Edge feature: Bond type

Biology / Simulation / etc.

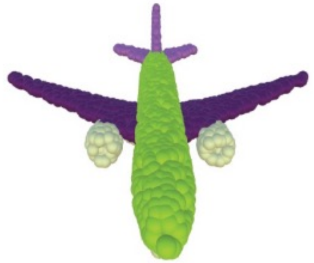
1)



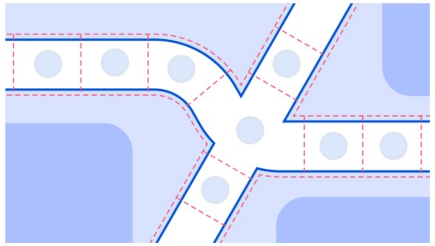
2)



3)



4)



Select benchmark datasets

- Reddit
- Ego-Facebook
- Github

- *Planetoid dataset (Cora/Citeseer/Pubmed)
- Coauthor
- WebKB (Texas/Cornell/etc.)

- QM9
- Zinc
- MUTAG

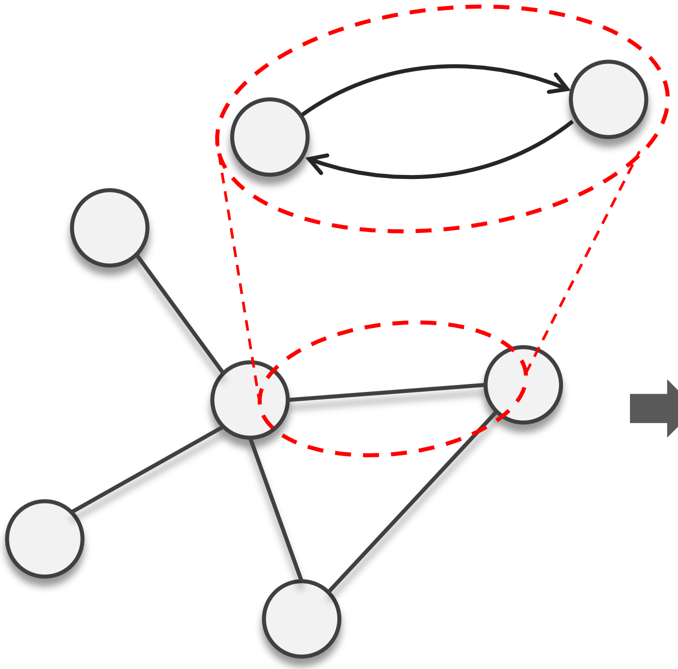
1) **PPI (protein-protein interaction)
2) Physical simulation (Kipf et al., 2018)
3) 3D point cloud (Wang et al., 2019)
4) Road network (Derrow-Pinion et al., 2021)

...and so much more

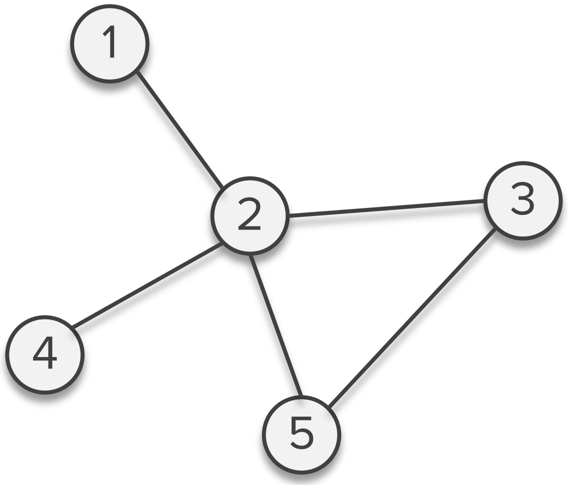
Yang et al., Revisiting Semi-Supervised Learning with Graph Embeddings, ICML 2016
Kipf et al., Neural Relational Inference for Interacting Systems, ICML 2018
Wang et al., Dynamic Graph CNN for Learning on Point Clouds, ACM Transactions on Graphics 2019
Derrow-Pinion et al., ETA Prediction with Graph Neural Networks in Google Maps, CIKM 2021
**Image source: https://www.researchgate.net/publication/324457787_iTRAQ_Quantitative_Proteomic_Analysis_of_Vitreous_from_Patients_with_Retinal_Detachment/figures?lo=1

Representing the graph as a adjacency matrix

*We treat undirected edges as
two directed edges going in both directions



Undirected graph



Assign arbitrary node ordering

- **Graphs with canonical node ordering is not common**
- Related research topic: Positional encoding in graphs (Maskey et al., NeurIPSW 2022)

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	1	1
3	0	1	0	0	1
4	0	1	0	0	0
5	0	1	0	1	0

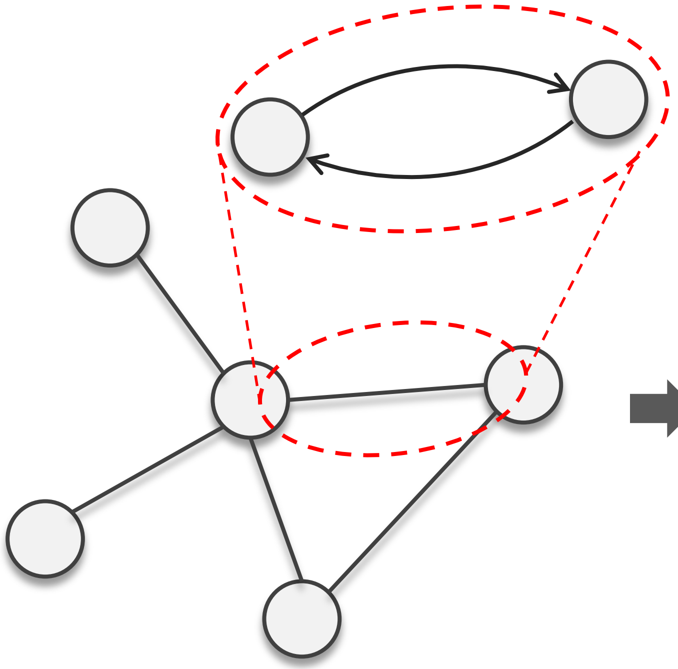
Adjacency matrix

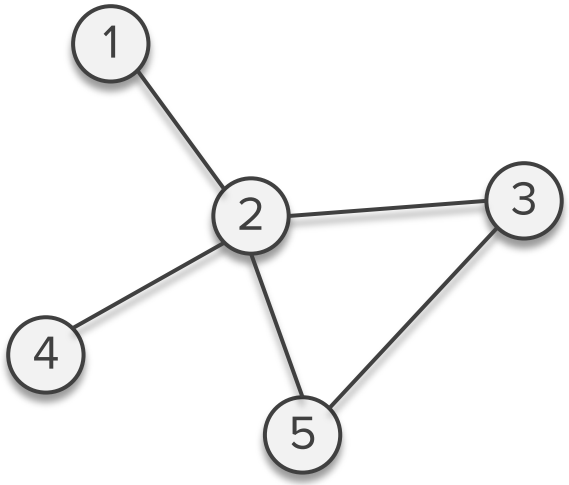
- Represent edge by assigning 1 for (i, j)-th element if node i and j are connected
- For weighted graphs: Assign a real number
- For graphs with multiple edges: Assign integers
- For directed graphs: Asymmetric matrix

Maskey et al., Generalized Laplacian Positional Encoding for Graph Representation Learning, NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations

Representing the graph as a adjacency matrix

*We treat undirected edges as
two directed edges going in both directions





(1, 2), (2, 1), (2, 3), (3, 2), …

Undirected graph

Assign arbitrary node ordering

- **Graphs with canonical node ordering is not common**
- Related research topic: Positional encoding in graphs (Maskey et al., NeurIPSW 2022)

Edge list

- Represent graph by listing all edges
- Notice that for undirected edges, (i, j) and (j,i) both appear
- More **memory efficient** than (dense) adjacency matrix

Maskey et al., Generalized Laplacian Positional Encoding for Graph Representation Learning, NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations

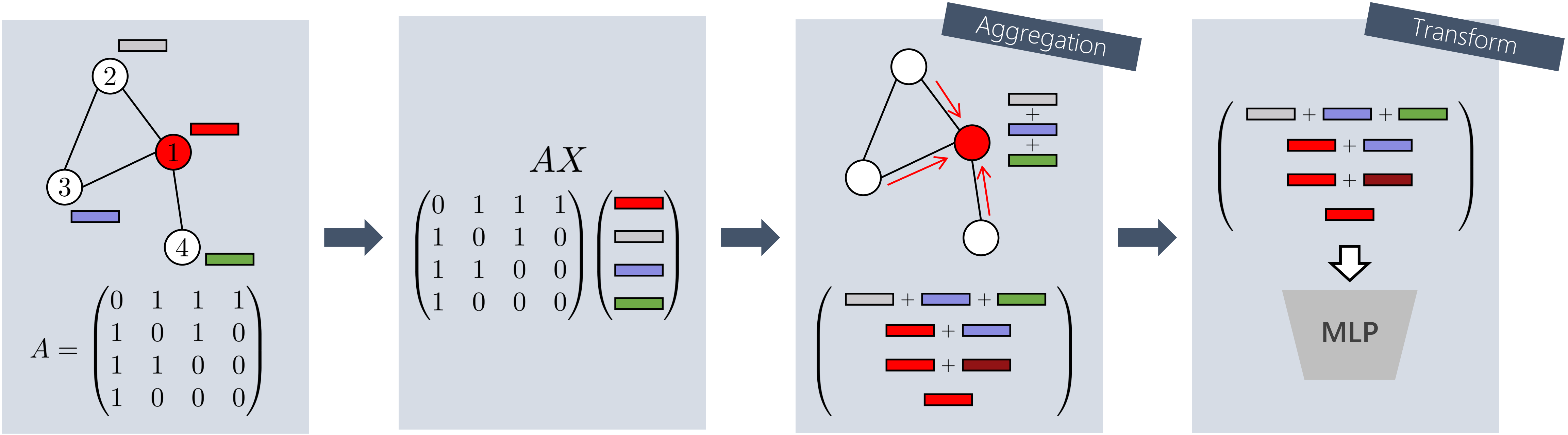
Part 1: A practical introduction to graphs and graph neural networks

Understanding of the general framework of graph neural networks (GNNs)

A simple, popular, and straightforward GNN

GCN (Graph Convolutional Network): Kipf & Welling, ICLR 2017

We are now ready to understand the basic principles of GNN, by looking at the most popular architecture.



Notice that, this whole procedure can be neatly expressed as: $\sigma(AX\Theta)$

Non-linear activation function $\sigma(\cdot)$

Adjacency matrix $A \in \mathbb{R}^{n \times n}$

Node feature matrix $X \in \mathbb{R}^{n \times d}$

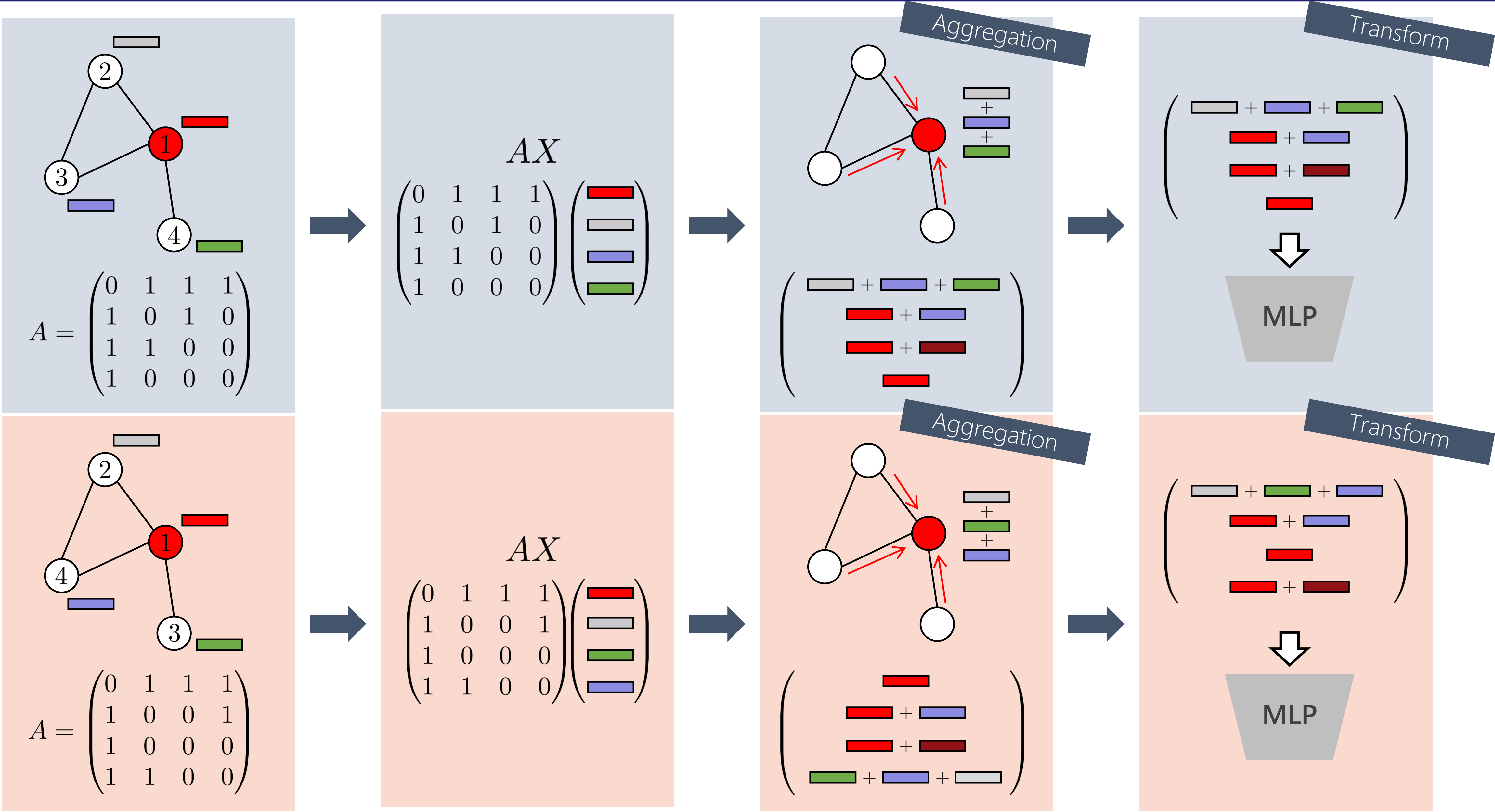
Learnable matrix $\Theta \in \mathbb{R}^{d \times d'}$

n: # of nodes

d: node feature dimensions

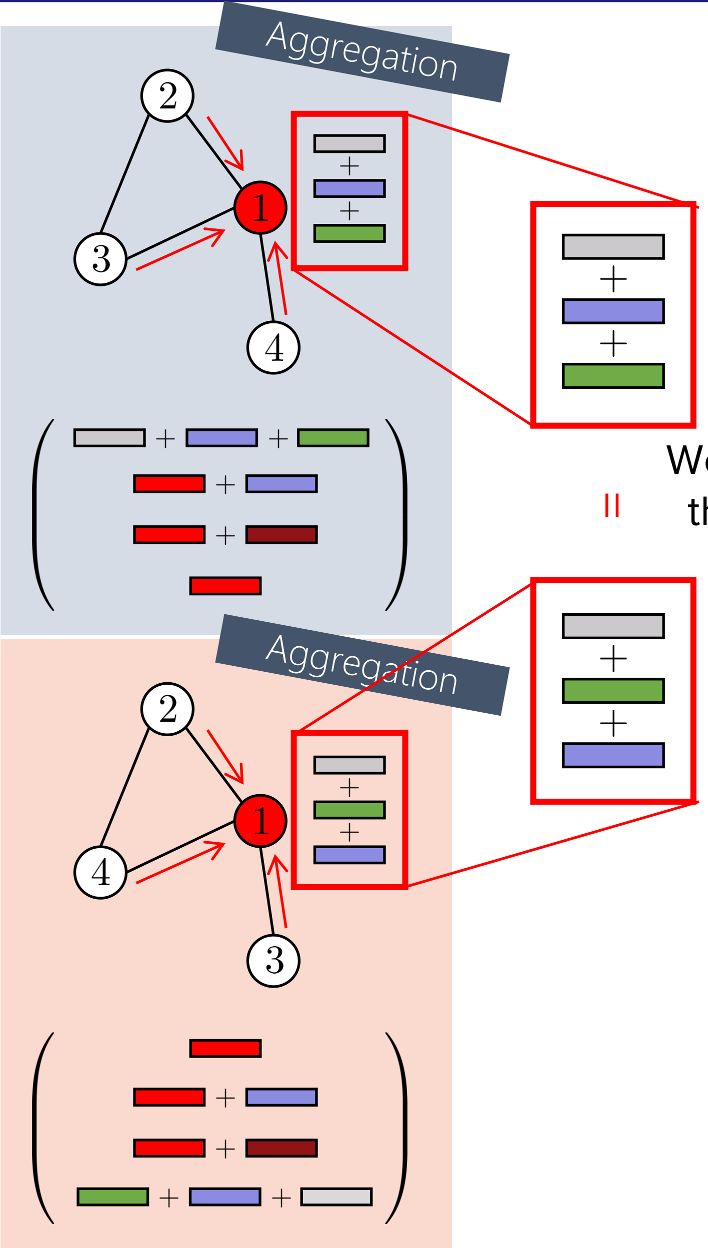
d': dimension for the next layer

A deeper look into the node ordering problem

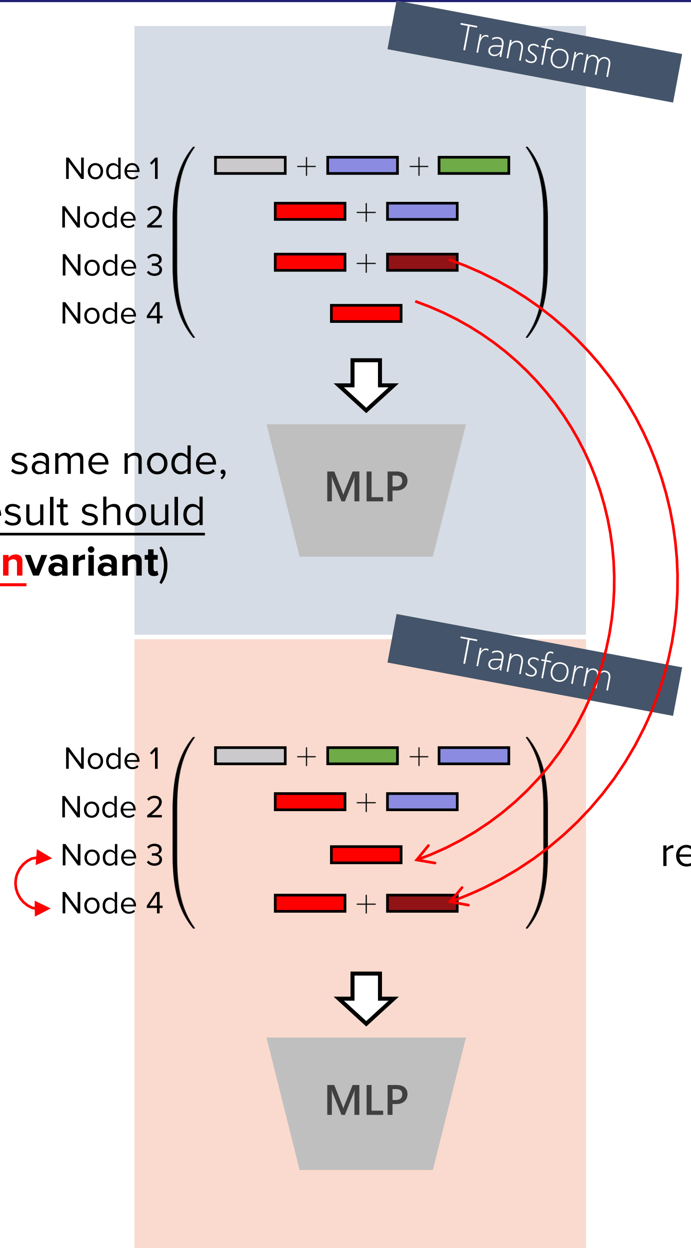


Alternate universe? (Different node ordering)

A deeper look into the node ordering problem



We are still computing for the same node, therefore the aggregation result should not change (**permutation invariant**)



As we have changed the node order, this should also be reflected in the embedding matrix (**permutation equivariant**)

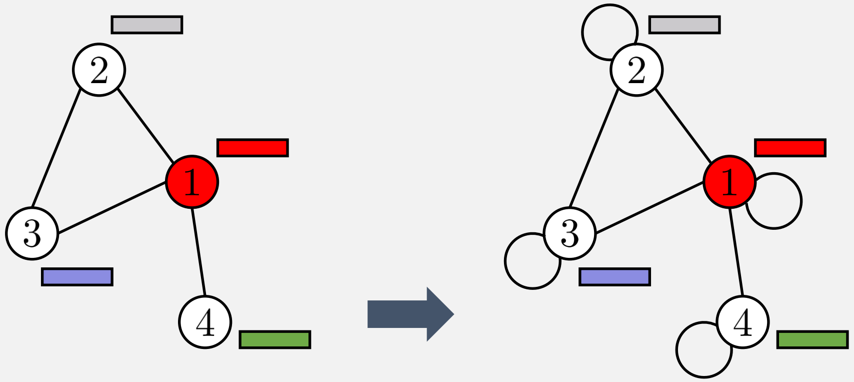
Practical design choices of GCN

Of course, we can get creative with the graph structure to solve some practical issues

Problem 1: The information of the neighbor nodes are aggregated but <u>not the node itself</u>.
Problem 2: The adjacency matrix is <u>not normalized</u>, and the scale of the feature vectors may explode for repeated layers.

Resolution to problem 1

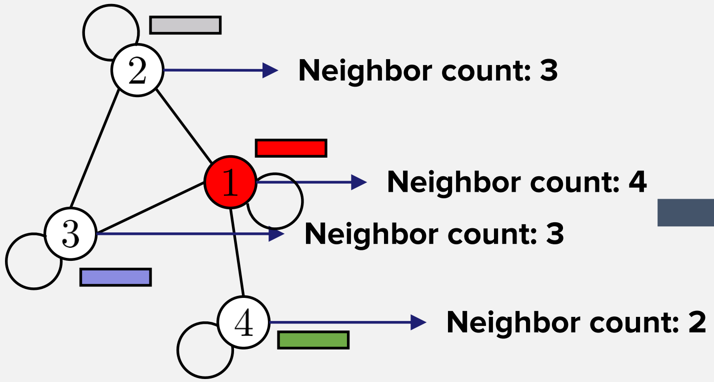
Add self-loops to each node



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \qquad \hat{A} = A + I = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Resolution to problem 2

Normalization of $\hat{A}$:



$$D = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\tilde{A} = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} = \begin{pmatrix} \frac{1}{4} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{12}} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{\sqrt{12}} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{\sqrt{8}} & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

Final layer of GCN: $\sigma(\tilde{A} X \Theta)$

Abstraction: A general message-passing layer of GNNs

1. Message passing phase (Aggregation)

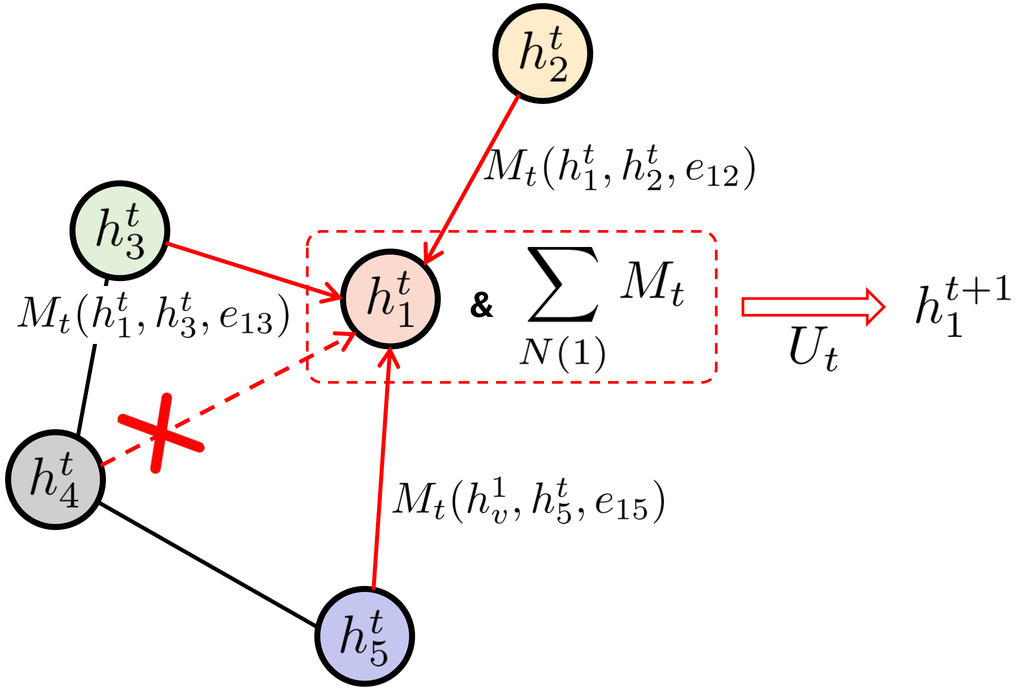
$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

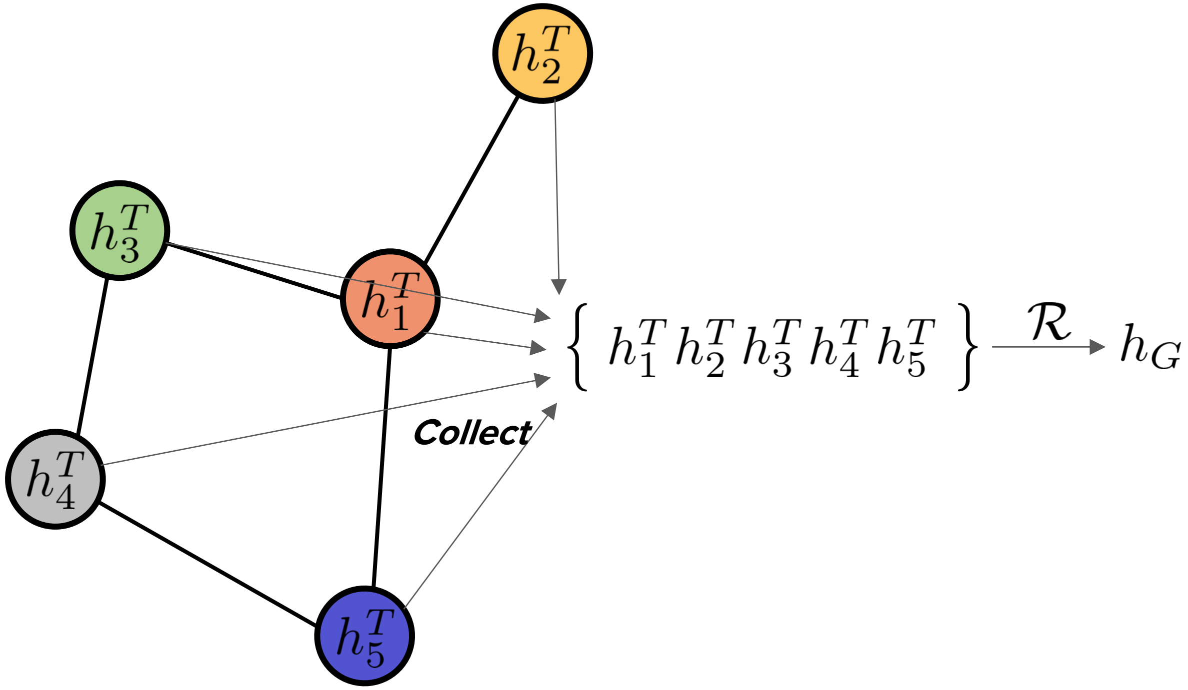
2. Update phase (Transformation)

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

3. Readout phase (Only for graph-level tasks)

$$h_G = \mathcal{R}(h_1^T, \cdots, h_{\mathcal{V}}^T)$$

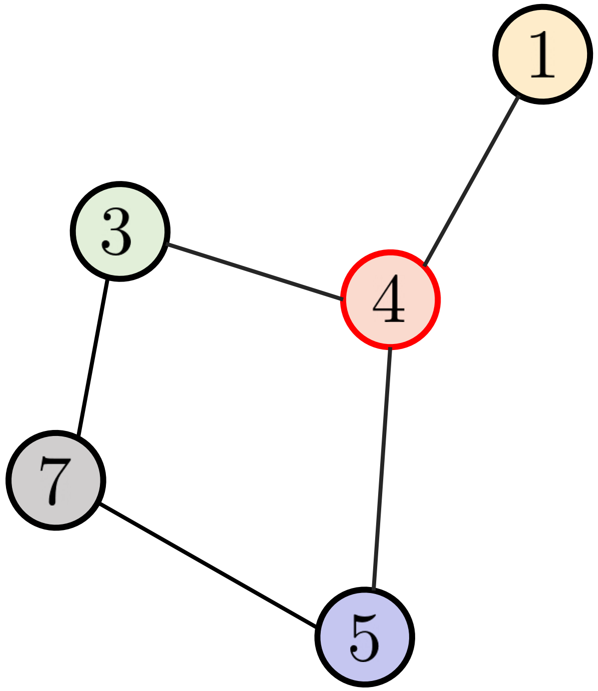




*Usually, we cite these papers for the term "message-passing"
[First formal introduction of the concept] Gilmer et al., "Neural Message Passing for Quantum Chemistry", ICML 2017
[Comprehensive discussion & abstraction] Bronstein et al., Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, arXiv 2021

Abstraction: A general message-passing layer of GNNs

GNN layer (Message-passing neural networks)



$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v)\right)$$

$$\bigoplus$$

This operation must be permutation invariant to ensure the same result for different node orderings!
Summation / Average / Max pooling etc.

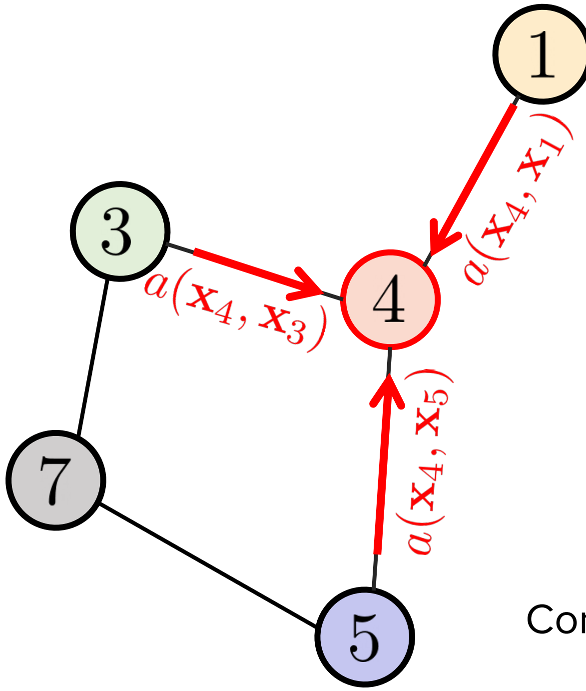
So if we re-describe GCN for node 4, it would be...

$$\mathcal{N}_u = \{1, 3, 5\} \cup \{4\} \quad \psi(\mathbf{x}_u, \mathbf{x}_1) = \frac{1}{\sqrt{2 \times 4}}\mathbf{x}_1 \quad \phi = \mathrm{MLP}$$

*Usually, we cite these papers for the term "message-passing"
[First formal introduction of the concept] Gilmer et al., "Neural Message Passing for Quantum Chemistry", ICML 2017
[Comprehensive discussion & abstraction] Bronstein et al., Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, arXiv 2021

Abstraction: A general message-passing layer of GNNs

Example: GAT (Veličković et al., ICLR 2018)



$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \boxed{\psi(\mathbf{x}_u, \mathbf{x}_v)}\right)$$

The model decides the strength of the 'propagation'

$$\mathcal{N}_u = \{1, 3, 5\} \cup \{4\} \quad \psi(\mathbf{x}_u, \mathbf{x}_1) = \boxed{a(\mathbf{x}_u, \mathbf{x}_1)}\mathbf{x}_1 \quad \phi = \mathrm{MLP}$$

Compare this part with GCN, the role of attention will be much more clear:

$$\psi(\mathbf{x}_u, \mathbf{x}_1) = \frac{1}{\sqrt{2 \times 4}} \mathbf{x}_1$$

The node degree decides the strength of the 'propagation'

There are a lot of fun & fundamental topics in the GNN literature

To name a few…

Relationship between GNNs and Graph Signal Processing

Industry-scale GNNs
(usually recommendation systems)

Explainable GNN
(this is part 2)

Basic GNN architectures

(GCN, GraphSAGE, GAT, GIN, etc.)

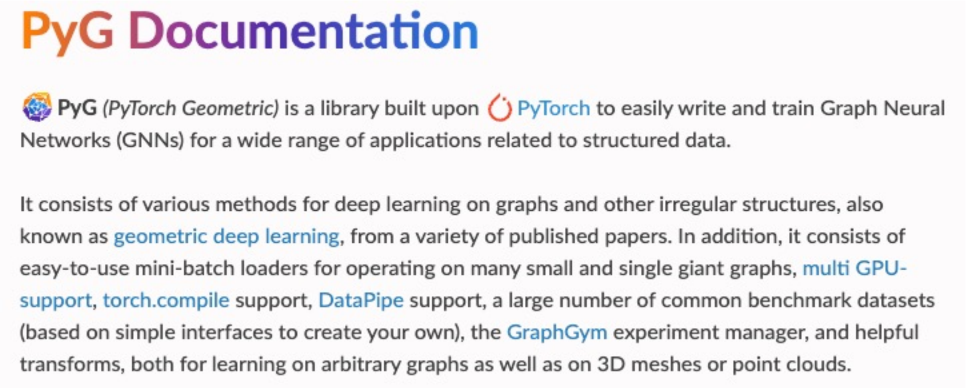
Overcoming limitations: Oversmoothing & Oversquashing in deep GNNs

GNN-to-MLP knowledge distillation for fast inference

Expressivity of GNNs: Connections to the Weisfeiler-Lehman isomorphism test

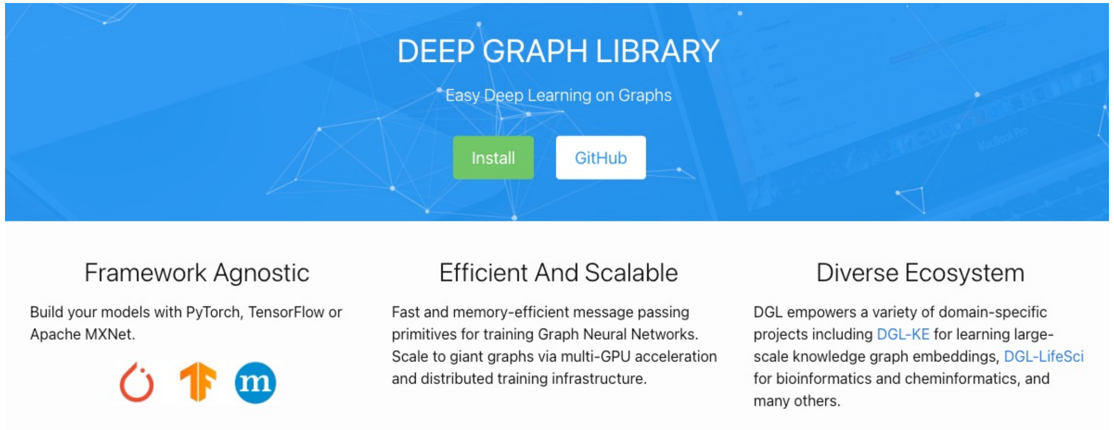
Before moving on, one slide on the library for graph learning

PyTorch Geometric (link)



- Jure Leskovec (Standford/KumoAI/Snapchat)
- Faster library updates
- (Seems like) A larger community

Deep Graph Library (link)



- Slower library updates
- Variable framework support
- Can be tricky to install older versions

NetworkX
Network Analysis in Python

- Additonal library: NetworkX (link) – Library for **graphs in general**
  - Not a library for ML/DL
  - Often used in junction with PyG/DGL

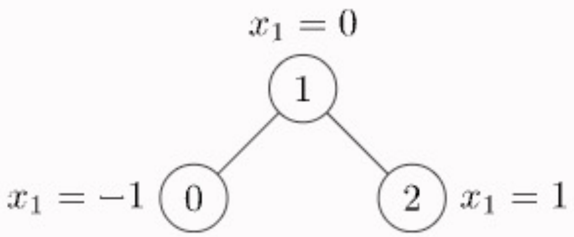
Wait, just one more slide on the library for graph learning

A very small PyG example

```
import torch
from torch_geometric.data import Data

edge_index = torch.tensor([[0, 1, 1, 2],
                           [1, 0, 2, 1]], dtype=torch.long)
x = torch.tensor([[-1], [0], [1]], dtype=torch.float)

data = Data(x=x, edge_index=edge_index)
>>> Data(edge_index=[2, 4], x=[3, 1])
```



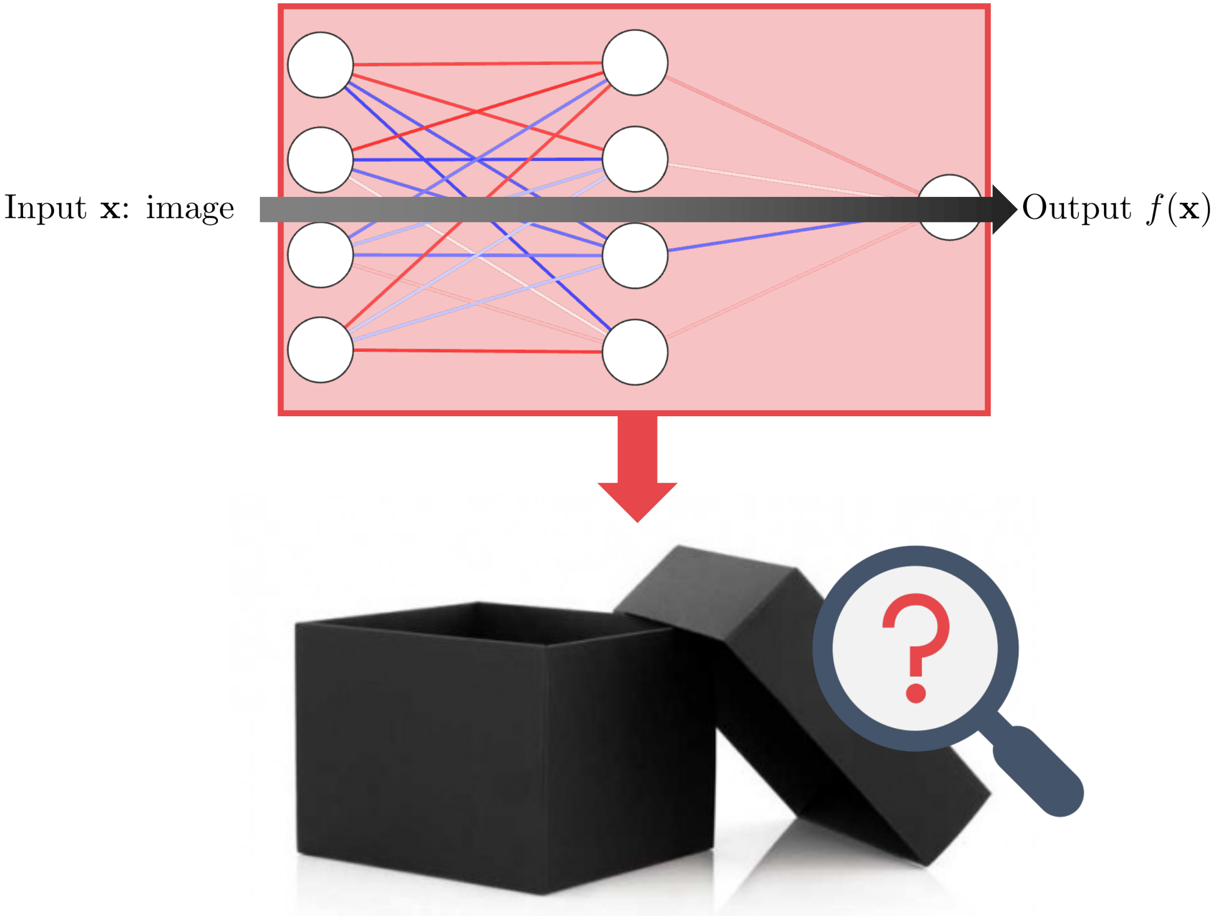
- You *at minimum* need to define data.edge_index
- Node features are usually represented as data.x
- Don't forget to include both directions for undirected graphs
- Most graph processing/manipulation tools are in torch_geometric.utils. Or just transform into a networkx object!

Part 2: Towards explainable graph learning with attention

Understanding the basic concepts of explainable AI

Why explainable AI?

Neural networks have complex structure with a lot of parameters.



- Large number of parameters
- High-nonlinearity
- Complex inner structure

has made them very ***hard to interpret and understand,** making it a **black box**.

Main question of Explainable AI:
Why has a neural network model made its prediction?.

Black box image source: https://www.kdnuggets.com/2015/04/model-interpretability-neural-networks-deep-learning.html
*Note: Linear models / decision trees are generally considered to be interpretable

Attribution maps: The most popular type of explanation

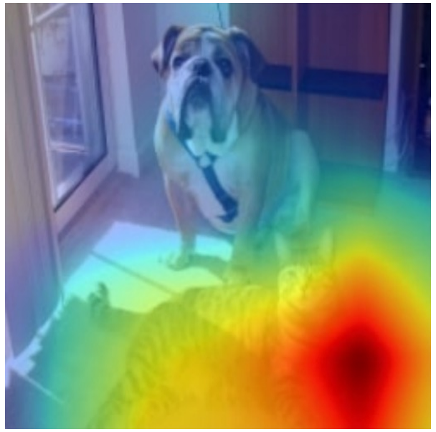
Attribution maps are one of the most popular ways, especially in CV and NLP.

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

Input image to ResNet

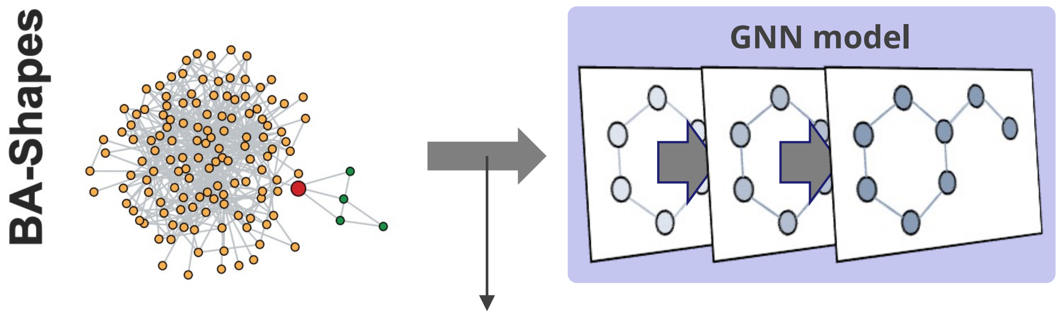
Output: "Cat"

Result of GradCAM

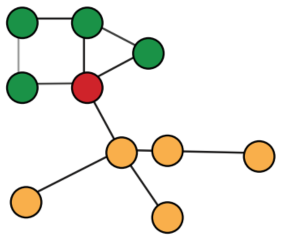


Highlights relevant pixels

Computation graph



Highlights relevant subgraph



Example: DTD [1], LRP [2], LIME [3], GradCAM [4], ...

Example: GNNExplainer [5], PGExplainer [6], ...

[1] Montavon et al., "Explaining nonlinear classification decisions with deep Taylor decomposition", Pattern Recognit. 65: 211-22 (2017)
[2] Bach et al., "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation", PLOS ONE 10(7): e0130140.
[3] Riberiro et al, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier", KDD 2016
[4] Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017
[5] Ying et al., "GNNExplainer: Generating Explanations for Graph Neural Networks", NeurIPS 2019
[6] Luo et al., "Parameterized explainer for graph neural network", NeurIPS 2020

Part 2: Towards explainable graph learning with attention

Can we understand graph attention networks using attention?

What is attention?

A weighted sum operation where the weights are determined by the model

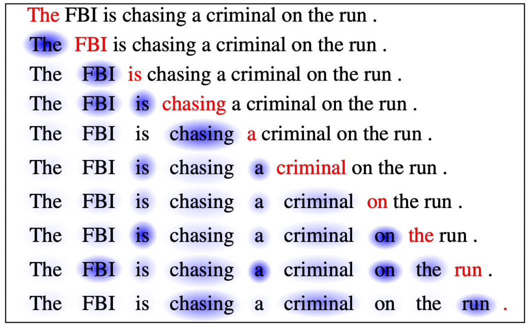
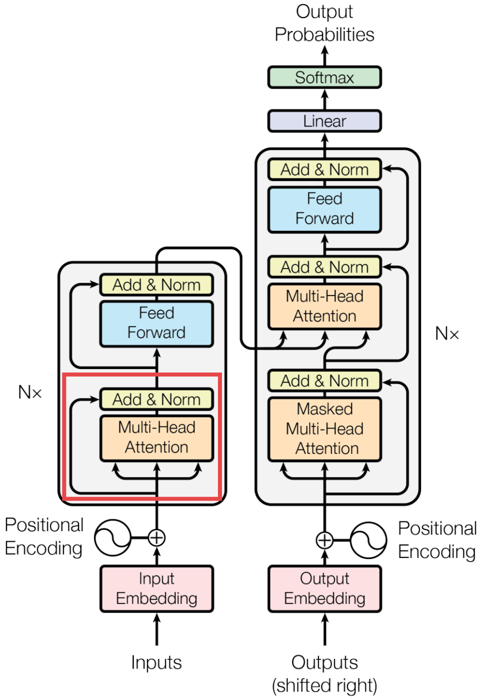
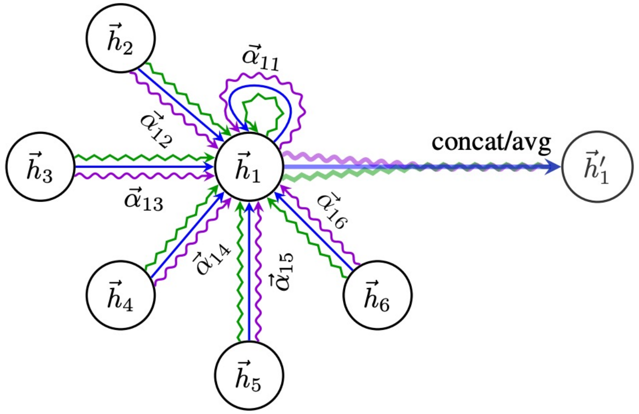


Figure 1: Illustration of our model while reading the sentence *The FBI is chasing a criminal on the run.* Color *red* represents the current word being fixated, *blue* represents memories. Shading indicates the degree of memory activation.

Long Short-Term Memory Networks (LSTMN)
(Cheng et al., EMNLP 2016)



Transformer networks
(Vaswani et al., NeurIPS 2017)

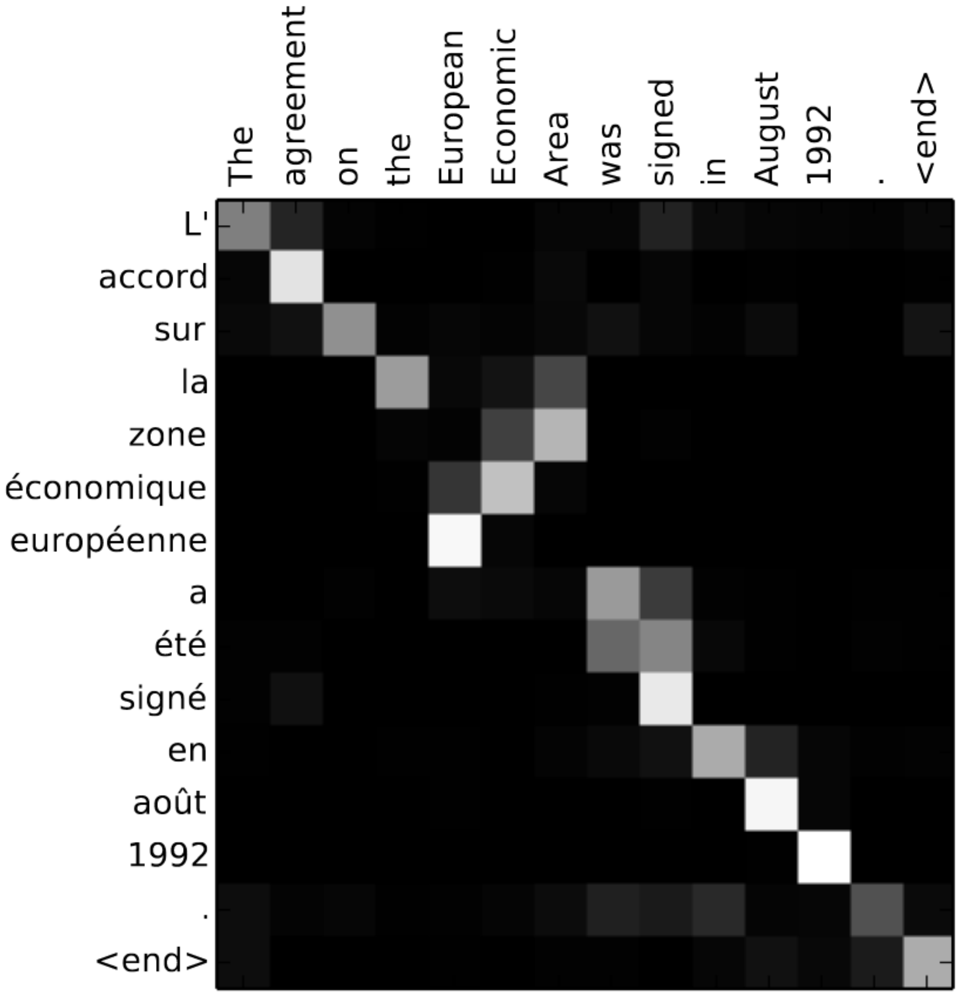


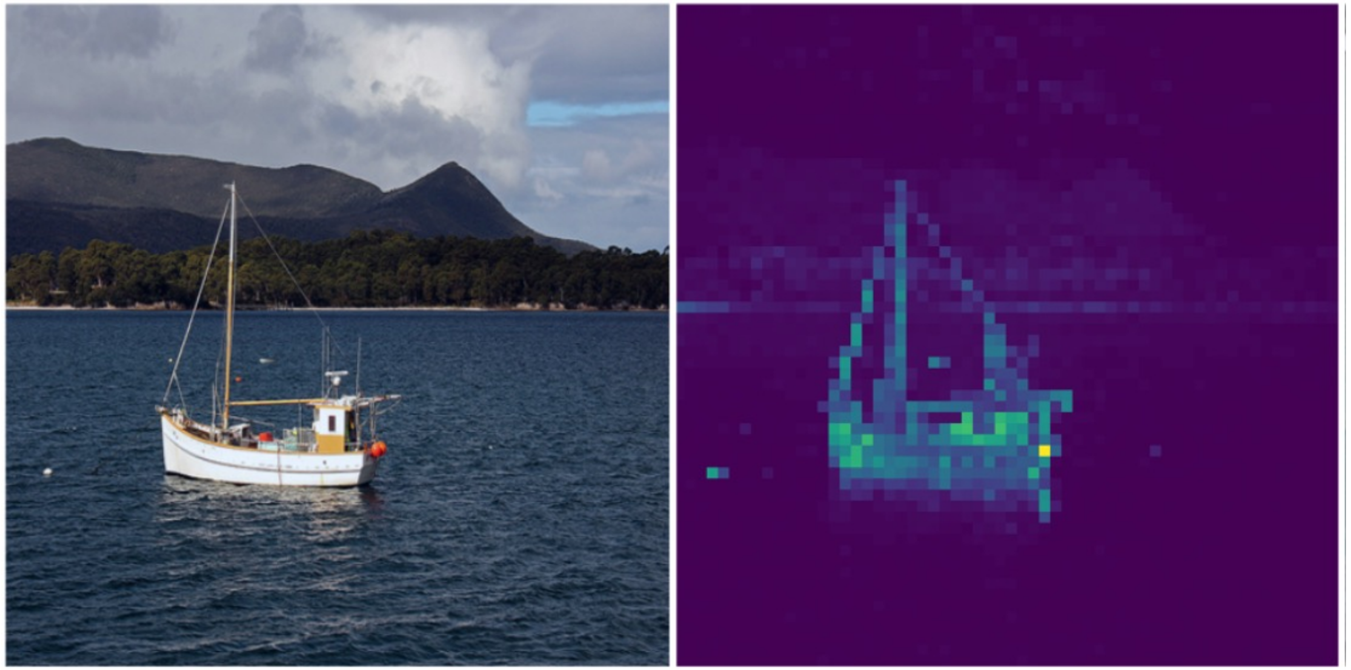
Graph attention networks
(Velickovic et al., ICLR 2018)

Cheng et al., Long Short-Term Memory-Networks for Machine Reading, EMNLP 2016
Vaswani et al., Attention Is All You Need, NeurIPS 2017
Velickovic et al., Graph attention networks, ICLR 2018

Can we interpret attention = attribution?

When we think of the role of attention, we can naturally interpret as '<u>where the model looks</u>'
...which is essentially **attribution maps!** (at least intuitively)





(Left) Bahdanau et al., Neural machine translation by jointly learning to align and translate, ICLR 2015
(Right) Caron et al., Emerging properties in self-supervised vision transformers, CVPR 2021

Can we just say attention = attribution?

Attention is heavily studied as an important candidate for model explanation

Is attention explanation?

Attention is not Explanation

Sarthak Jain
Northeastern University
jain.sar@husky.neu.edu

Byron C. Wallace

Attention is not not Explanation

Sarah Wiegreffe*

Yuval Pinter*
School of Interactive Computing
Georgia Institute of Technology
uvp@gatech.edu

Is Attention Explanation? An Introduction to the Debate

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François* and **Patrick Watrin***
CENTAL, IL&C, University of Louvain, Belgium
{adrien.bibal,remi.card
thomas.francois,pat

**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

How to generate better attention heatmaps in transformers?

Quantifying Attention Flow in Transformers

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

Willem Zuidema
ILLC, University of Amsterdam

Transformer Interpretability Beyond Attention Visualization

Hila Chefer[1] Shir Gur[1] Lior Wolf[1,2]
[1]The School of Computer Science, Tel Aviv University
[2]Facebook AI Research (FAIR)

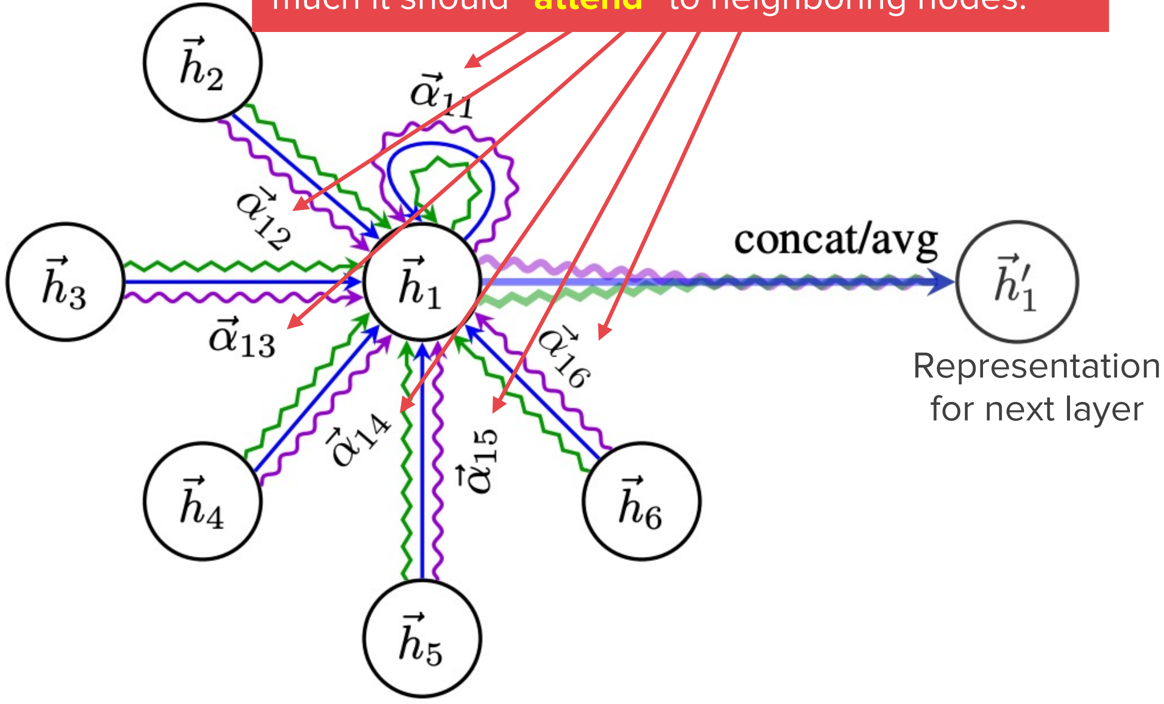
Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers

Hila Chefer[1] Shir Gur[1] Lior Wolf[1,2]
[1]The School of Computer Science, Tel Aviv University
[2]Facebook AI Research (FAIR)

So graph attention networks (GATs) are explainable?

Well, the majority of the literature seems to overlook GATs as a valid candidate of ‘inherently explainable model’

Just like Transformers, **the model decides** how much it should ”**attend**” to neighboring nodes.



One layer of GAT

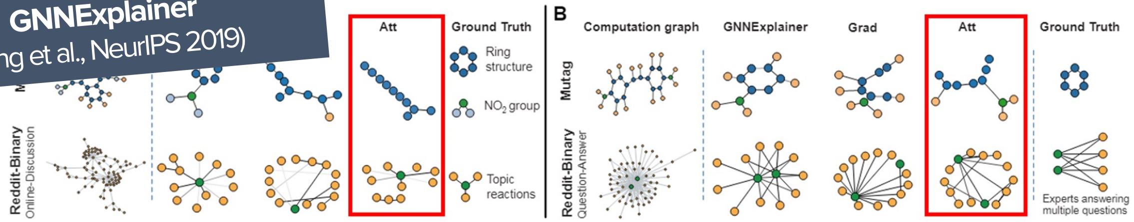
GNN-XAI evaluation
(Sanchez-Lengeling et al., NeurIPS 2020)

Node-level tasks

					BA-Community				Tree-Grid			
			…phNets	GAT	GCN	MPNN	GraphNets	GAT	GCN	MPNN	GraphNets	GAT
…aseline	0.27	0.27	0.27	0.27	0.38	0.38	0.38	0.38	0.62	0.62	0.62	0.62
GradInput	0.58	0.64	0.39	0.72	0.52	0.51	0.5	0.5	0.65	0.71	0.66	0.67
SmoothGrad(GI)	0.58	0.64	0.39	0.72	0.52	0.51	0.51	0.49	0.65	0.71	0.66	0.67
GradCAM-last	0.79	0.84	0.86	0.8	0.7	0.67	0.68	0.61	0.7	0.77	0.81	0.7
GradCAM-all	0.67	0.78	0.65	0.76	0.67	0.71	0.73	0.57	0.68	0.7	0.67	0.68
IG	--	--	--	--	0.81	0.75	0.72	0.62	--	--	--	--
Attention Weights	--	--	--	0.5	--	--	--	0.5	--	--	--	0.49

“…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)

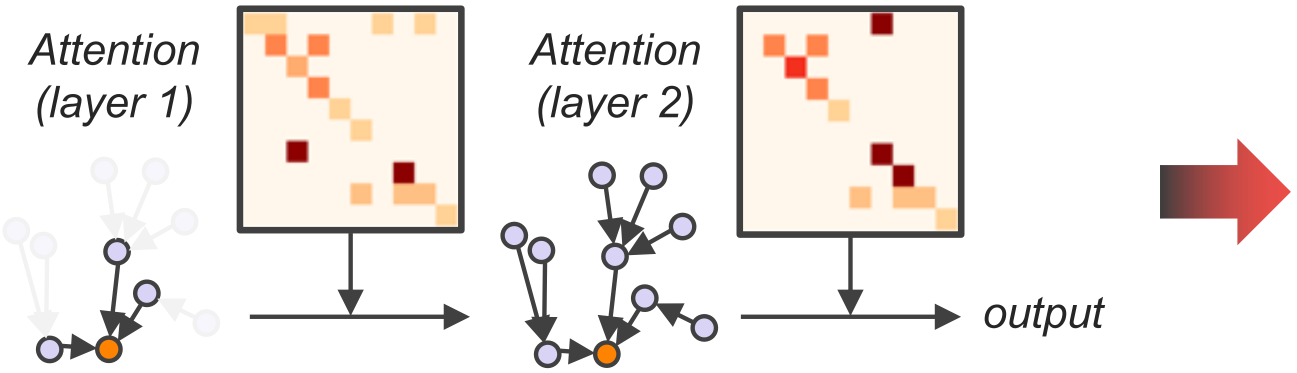
			Explanation AUC			
		0.750	0.905	0.612	0.717	0.783
		0.739	0.824	0.667	0.674	0.765
		-	-	-	0.773	0.653
…xplainer	0.925	0.836	0.948	0.875	0.742	0.727
PGExplainer	**0.963**±0.011	**0.945**±0.019	**0.987**±0.007	**0.907**±0.014	**0.926**±0.021	**0.873**±0.013
Improve	4.1%	13.0%	4.1%	3.7%	24.7%	11.5%
			Inference Time (ms)			
GNNExplainer	650.60	696.61	690.13	713.40	934.72	409.98
PGExplainer	10.92	24.07	6.36	6.72	80.13	9.68
Speed-up	59x	29x	108x	106x	12x	42x

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

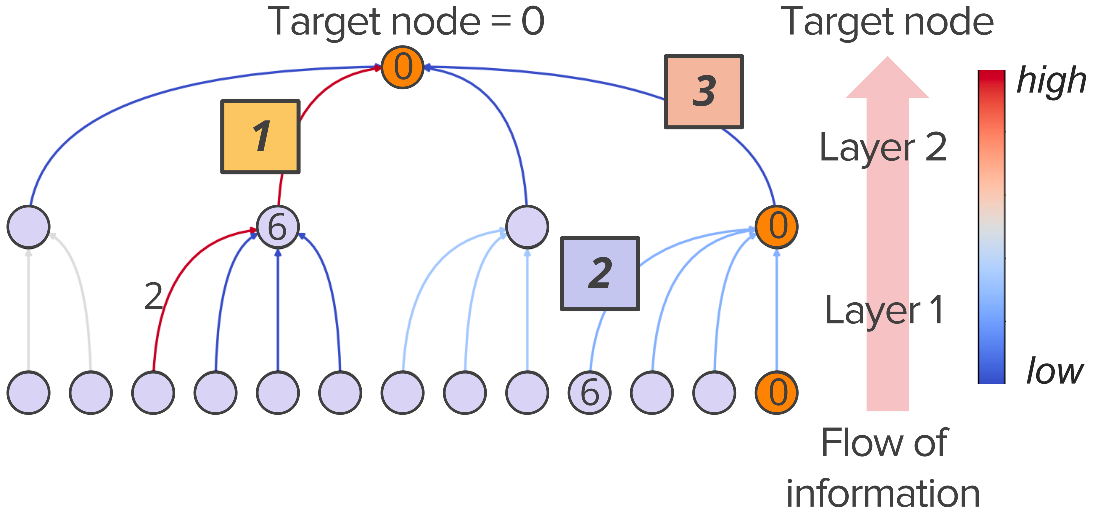
(Left) Velickovic et al., Graph attention networks, ICLR 2018
Sanchez-Lengeling et al., Evaluating attribution for graph neural networks, NeurIPS 2020
Ying et al., GNNExplainer: Generating explanations for graph neural networks, NeurIPS 2019
Luo et al., Parameterized Explainer for Graph Neural Network, NeurIPS 2020

GATs are explainable… with a little bit of extra effort

We just need to consider the ‘flow’ of information better within the GAT model



“Computation tree”



Proposed calculation (Shin et al., AAAI 2025)

$$\phi^{0}_{6,0} = \boxed{1} + \boxed{3} \times \boxed{2}$$

“Importance of edge (6, 0) when the target node is 0”

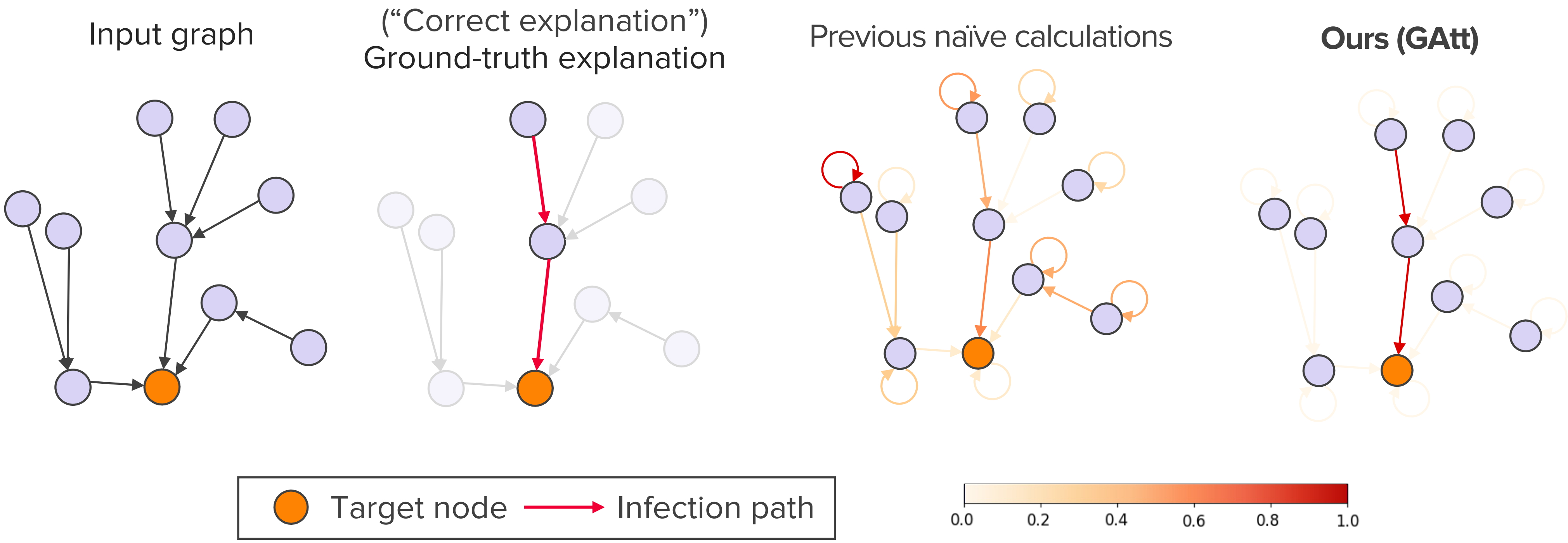
1. Add all occurances in the computation tree

2. Multiply other attention weights along the flow of information

Shin et al., “Faithful and Accurate Self-Attention Attribution for Message Passing Neural Networks via the Computation Tree Viewpoint”, AAAI 2025

And we can immediately get better attribution maps

Case study: Infection dataset (Faber et al., KDD 2021)



*Note: We do not change the GAT model. Remember, our contribution is how to calculate attribution maps after the training is complete

Faber et al., "When Comparing to Ground Truth is Wrong: On Evaluating GNN Explanation Methods", KDD 2021

Takeaways

Part 1: A practical introduction to graphs and graph neural networks

1. Understanding of **graphs** as a **general data type**
   - Nodes & Edges (“connections”)
   - A lot of things can be represented as a graph, including images!
2. Understanding of the general framework of **graph neural networks (GNNs)**
   - Message-passing = GNN (Unless it’s a graph transformer)
   - Aggregation + Transformation

Part 2: Towards explainable graph learning with attention

1. Understanding the basic concepts of **explainable AI**
   - Attribution maps = “Important parts of the input”
   - A lot of GNN explanations are also attribution maps
2. **Answer to the question: Can we understand graph attention networks using attention?**
   - (Shin et al., AAAI 2025) Conclusion: YES, but with a little bit of effort
   - BTW, this conclusion is applicable to other GNNs with self-attention

Thank you!

Please feel free to ask any questions :)

jordan7186.github.io

jordan3414@yonsei.ac.kr