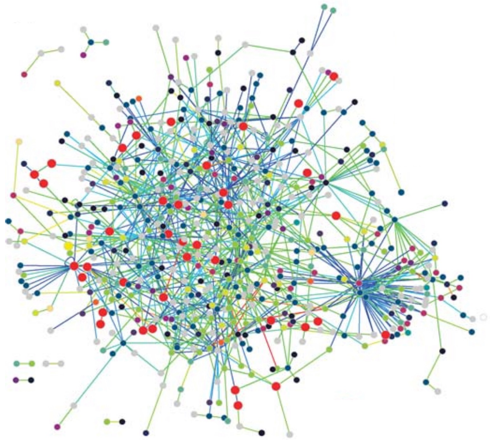
Unearthing the Relationship Between Graph Neural Networks and Matrix Factorization

Bruno Ribeiro
Assistant Professor
Department of Computer Science
Purdue University

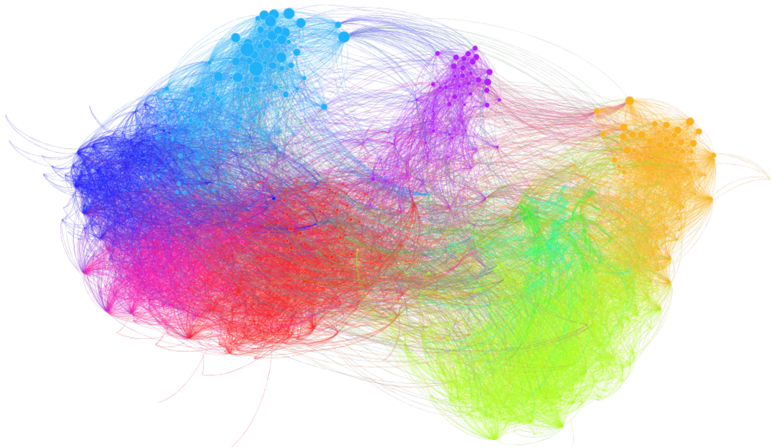
Twitter,
September 2nd , 2021

Work w/ Bala Srinivasan, Beatrice Bevilacqua, Jianfei Gao, Yangze Zhou, S Chandra Mouli

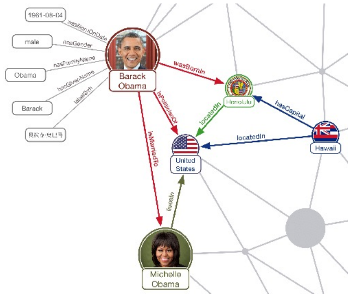
Traditional Graph Tasks



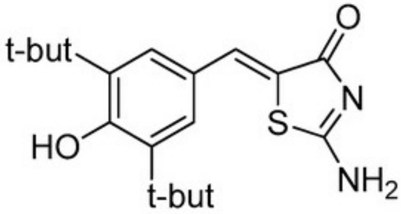
Biological Graphs



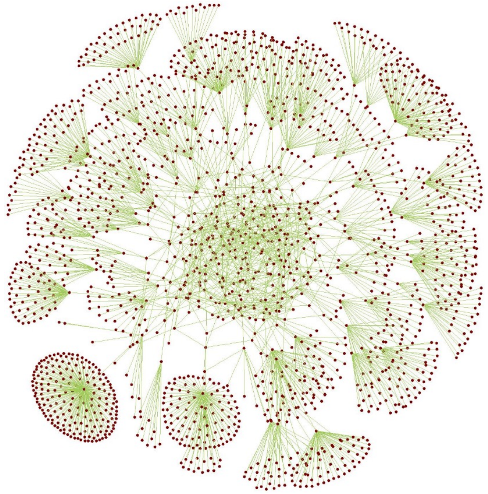
Social Graphs



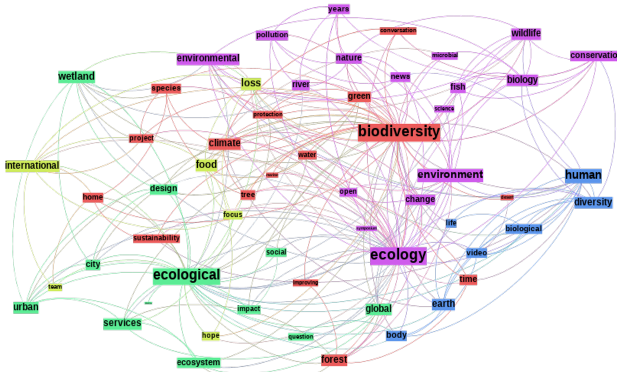
Knowledge graph reasoning



Predict molecular properties



The Web



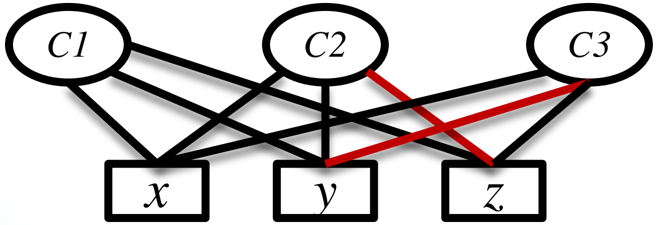
Ecological Graphs

Recent years: Dramatic expansion of graph tasks

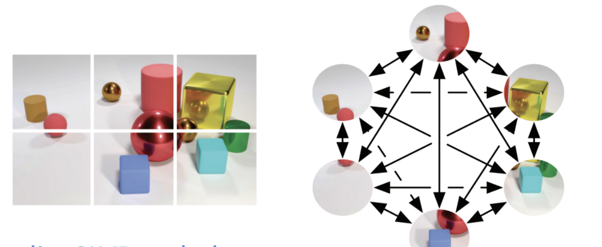
New Graph Tasks in Machine Learning

- Logical reasoning

$$(x \lor y \lor z) \land (x \lor y \lor \neg z) \land (x \lor \neg y \lor z)$$

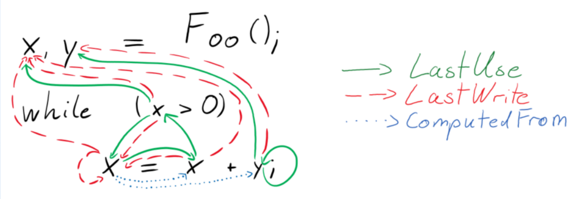


- Scene understanding, world understanding (RL)



Credit: CLVR task dataset

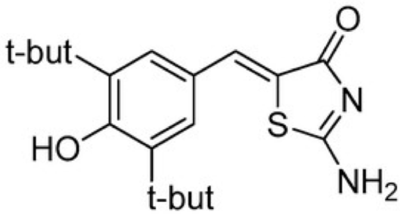
- Program synthesis



Source: IntelliCode

e.g.: Learning to Represent Programs with Graphs ICLR'18
Miltiadis Allamanis, Marc Brockschmidt, Mahmoud Khademi

- Graph generation (drug discovery)

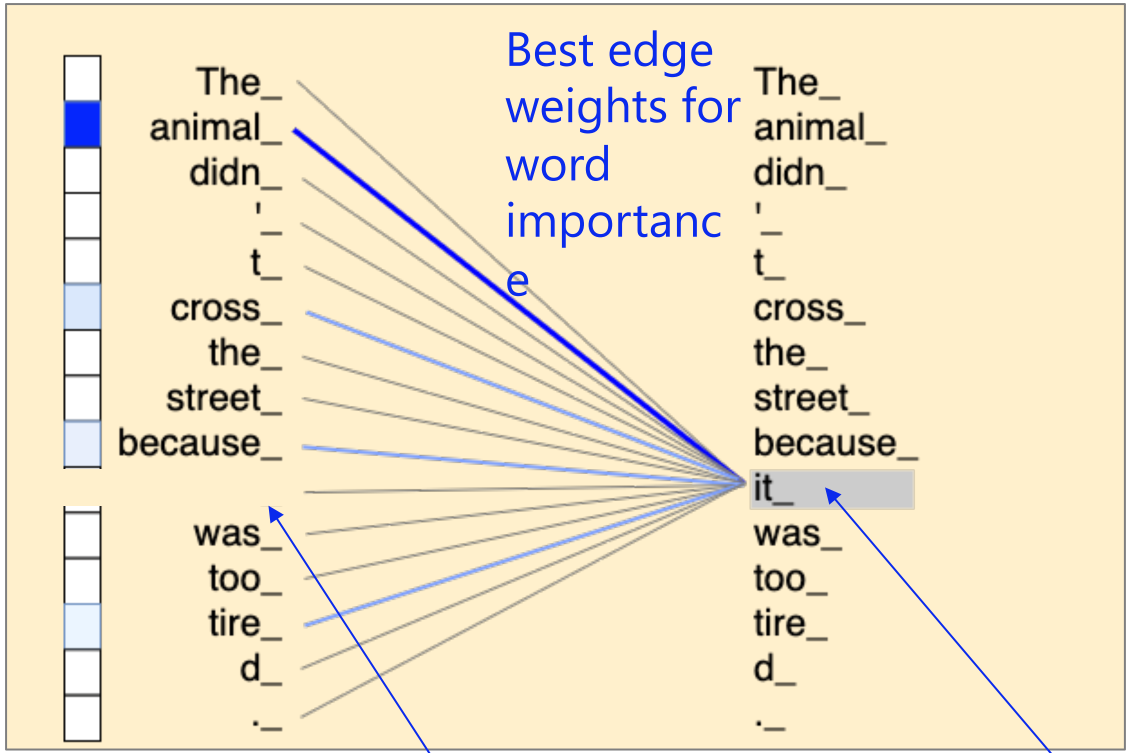


- Natural language processing (Transformers)

Bruno Ribeiro

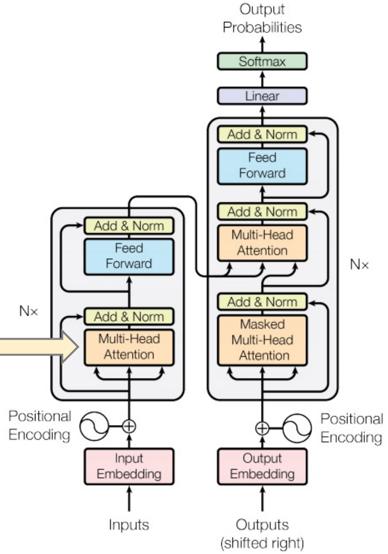
Example of **Graph** Application in Language: Text Generation / Translation / Comprehension

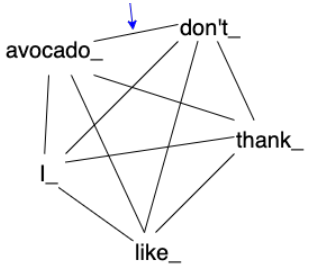
- Self-attention is a type of graph neural network



Masked word

Able to predict the masked word



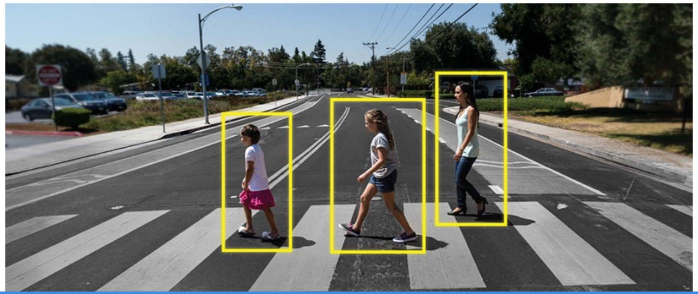


Graph with edges weights representing important words in "I don't like avocado"

What is special about neural networks for graphs?

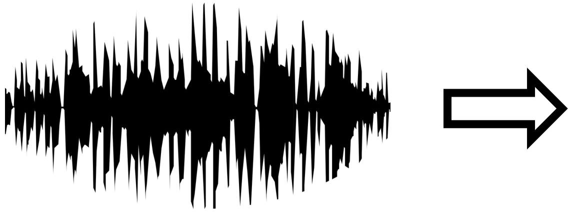
Tasks Standard Neural Networks **Excel**

- Image/Video Tasks



What do these tasks have in common?

- Text Recognition / Prediction
  - The quick brown fox jumps over the lazy dog
- Speech Recognition

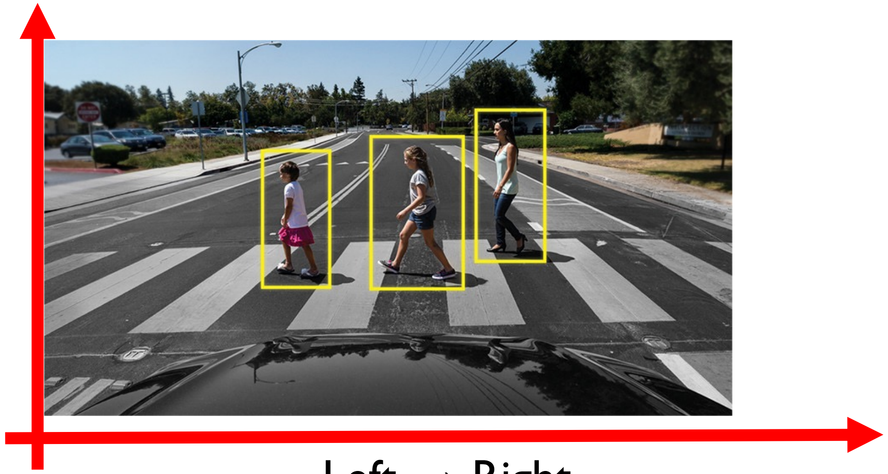


quick brown fox

Standard Neural Networks Need a **Canonical Orientation**

- All training examples have the same orientation
- Input can be represented as a **vector** (ordered set)

Down → Up

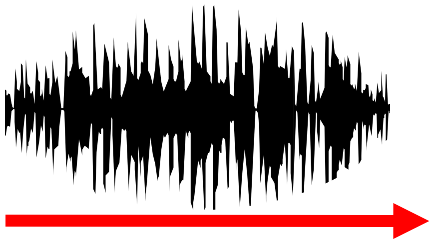


Left → Right

The quick brown fox jumps over the lazy dog

Left → Right

Sound wave:



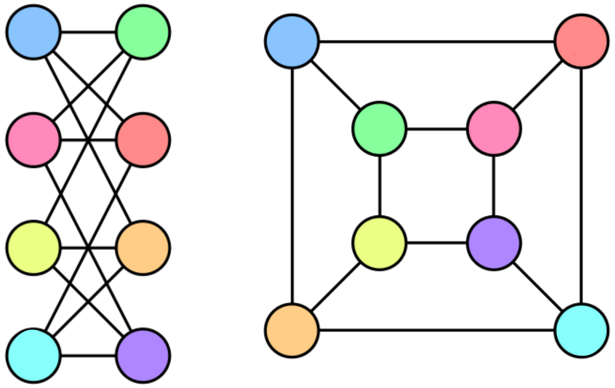
Past → Future

Graph Input Has no Clear Canonical Orientation

- Example:

Consider two graphs

- Are these the same graph?

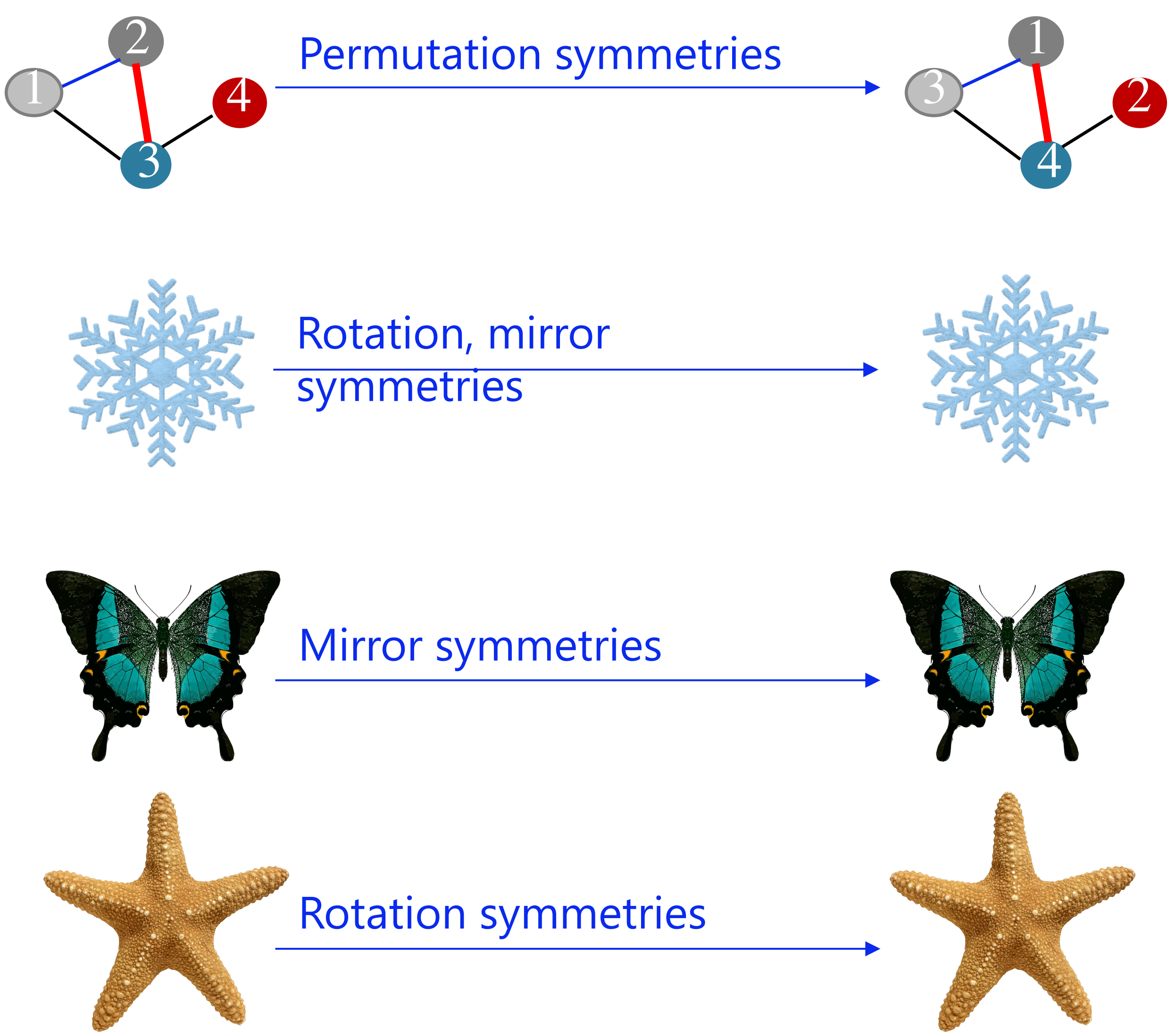


src: Wikipedia

isomorphic graphs = same input

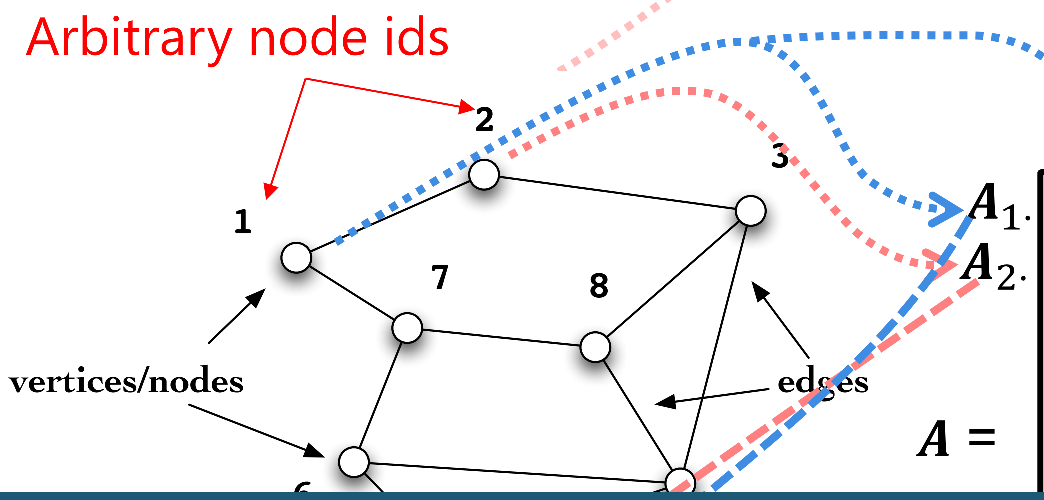
Machine Learning on Graphs is all about Symmetry

Symmetries in nature



Invariance/Equivariance in Graph Representation Learning

Step 1: Encode all graph + node & edge attributes as a matrix (Tensor)



$$
\begin{array}{cc}
 & \begin{array}{cccccccc} A_{\cdot 1} & A_{\cdot 2} & & & & & & \end{array} \\
\begin{array}{c} A_{1\cdot} \\ A_{2\cdot} \\ \\ \\ \\ \end{array} & A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}
\end{array}
$$

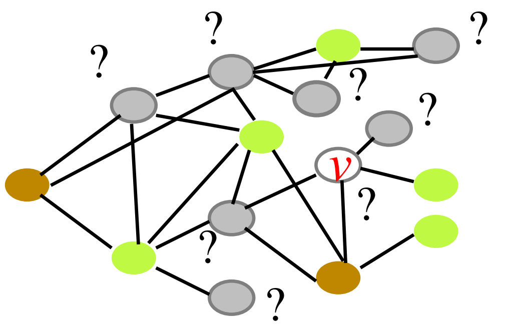
Elements of Graph matrices may be *m*-dimensional (for arbitrary *m*):

- To encode node and edge attributes
- To encode multiple edges (for multiplex networks)

$$
\boldsymbol{\pi} = (2,1,3,4,5,6,7,8) \Rightarrow \quad \rho_\pi(\boldsymbol{A}) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}
$$

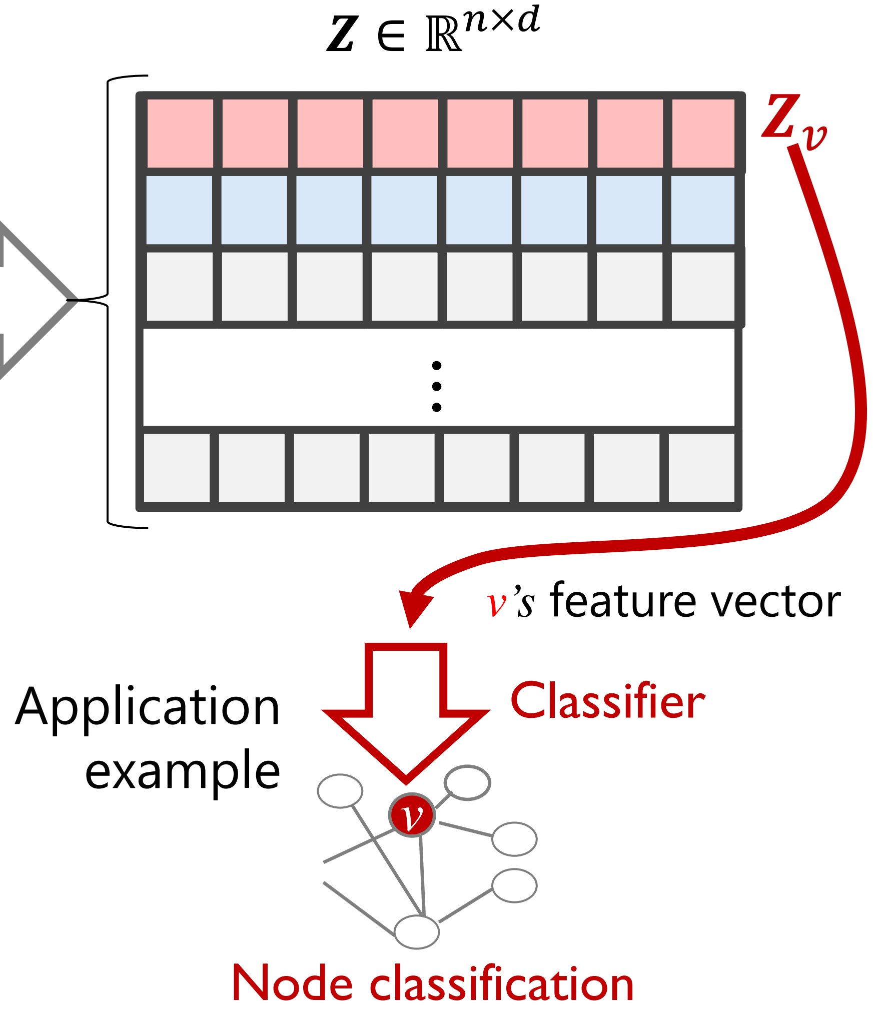
Step 2: Define Representation (e.g. Graph Neural Networks (GNNs))

- A GNN $f_W(\boldsymbol{A})$ is a neural network that learn graph $\boldsymbol{A}$ representations that are ***equivariant*** to node permutations



$$\boldsymbol{A} = \begin{array}{cc} & v \\ v & \left[\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right] \end{array}$$

$$f_W(\boldsymbol{A}) = \boldsymbol{Z}$$



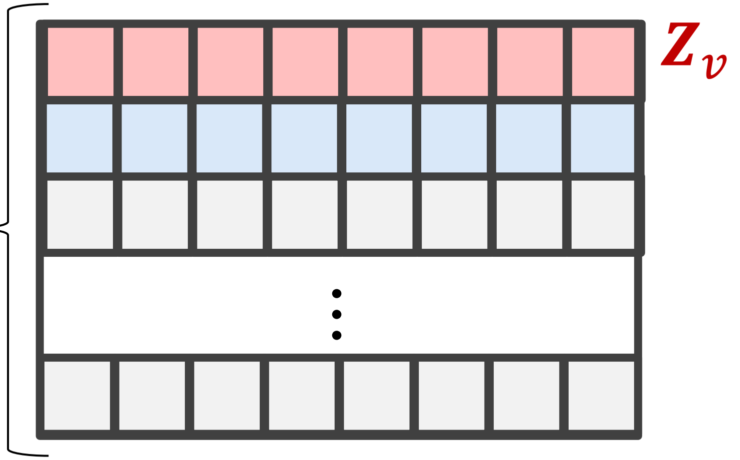
Democrat (DEM)
Republican (GOP)

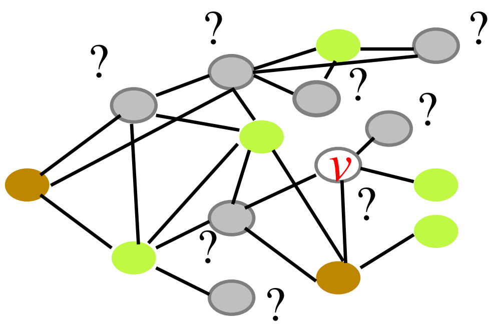
Step 2: Define Representation (e.g. Graph Neural Networks (GNNs))

- A GNN $f_W(\boldsymbol{A})$ is a neural network that learn graph $\boldsymbol{A}$ representations that are ***equivariant*** to node permutations

$$\begin{array}{cc} & \color{red}{v} \\ \color{red}{v} & \\ \boldsymbol{A} = & \begin{bmatrix} 0&1&1&0&0&0&0&0 \\ 1&0&0&0&0&0&1&0 \\ 1&0&0&1&0&0&0&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&1&1&0&1&0 \\ 0&0&0&0&0&1&0&1 \\ 0&1&1&1&0&0&1&0 \end{bmatrix} \end{array} \qquad f_W(\boldsymbol{A}) = \boldsymbol{Z}$$

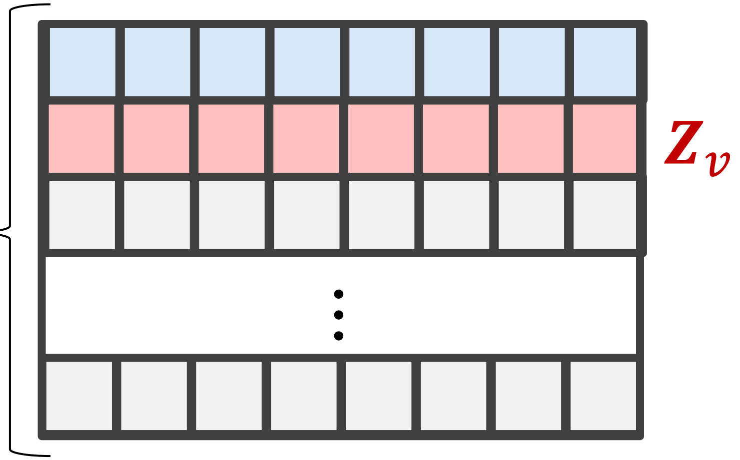
$\boldsymbol{Z} \in \mathbb{R}^{n \times d}$



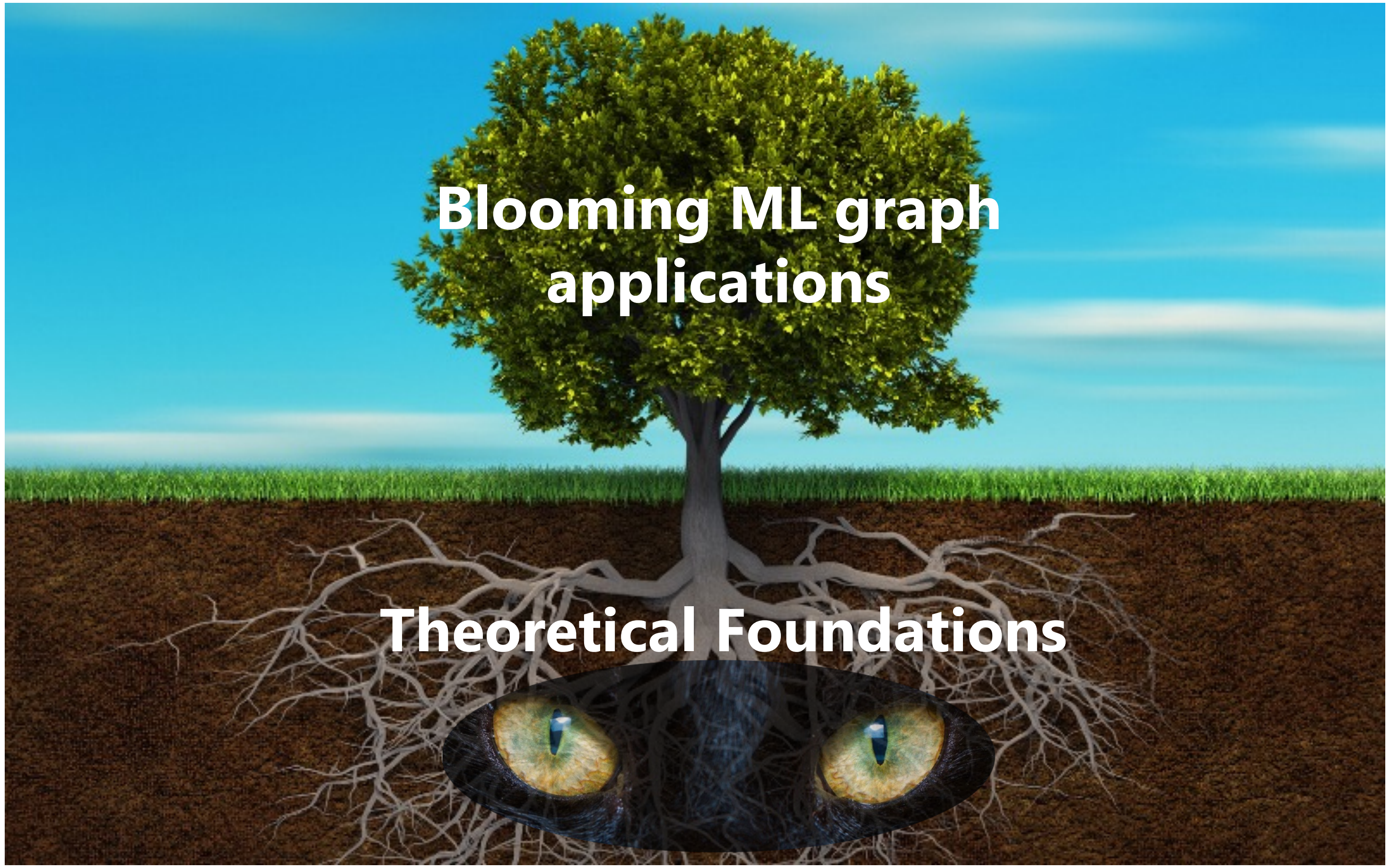


Democrat (DEM)
Republican (GOP)

$$\rho_\pi(\boldsymbol{A}) = \begin{bmatrix} 1&0&0&0&0&0&1&0 \\ 0&1&1&0&0&0&0&0 \\ 0&1&0&1&0&0&0&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&1&1&0&1&0 \\ 0&0&0&0&0&1&0&1 \\ 1&0&1&1&0&0&1&0 \end{bmatrix} \qquad f_W(\rho_\pi(\boldsymbol{A})) = \rho'_\pi(\boldsymbol{Z})$$



Blooming graph applications

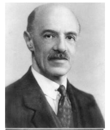


This talk: What is lurking in the foundations?

Equivariance to node permutations is the **defining characteristic** of node embeddings in graph representation learning

What about matrix factorization?

Matrix Factorization

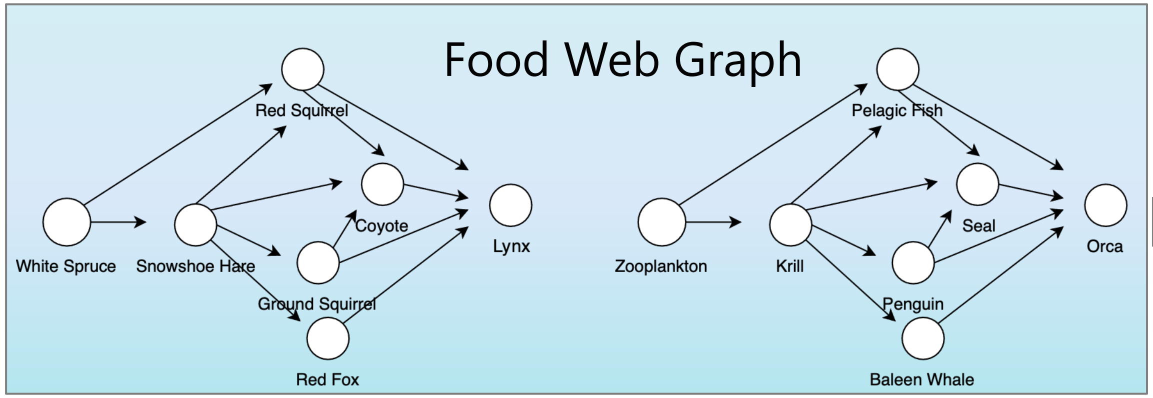


- Node embeddings arguably first defined by the common **factors** of Spearman (1904) via Singular Value Decomposition (**SVD**)
- Hotelling (1933) and others defined Principal Component Analysis (**PCA**)
- 116+ years later, we essentially use the same methods
  - They are very useful!

The node order matters on algorithm output

But SVD node embeddings are NOT equivariant

$$\boldsymbol{A} = \begin{bmatrix} A_{\text{land}} & \\ & A_{\text{sea}} \end{bmatrix}$$

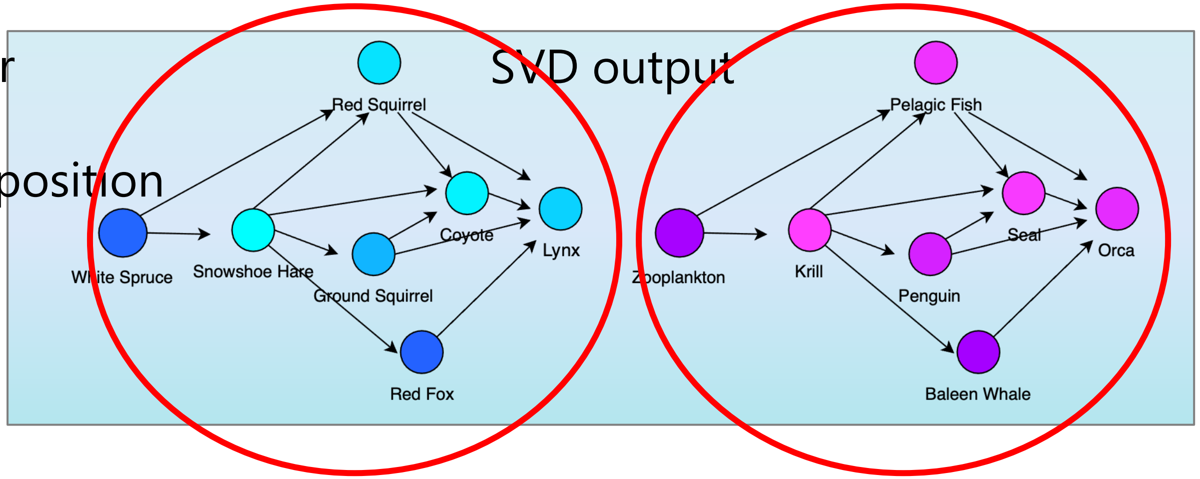


Singular **V**alue **D**ecomposition

SVD

Cluster 1

Cluster 2



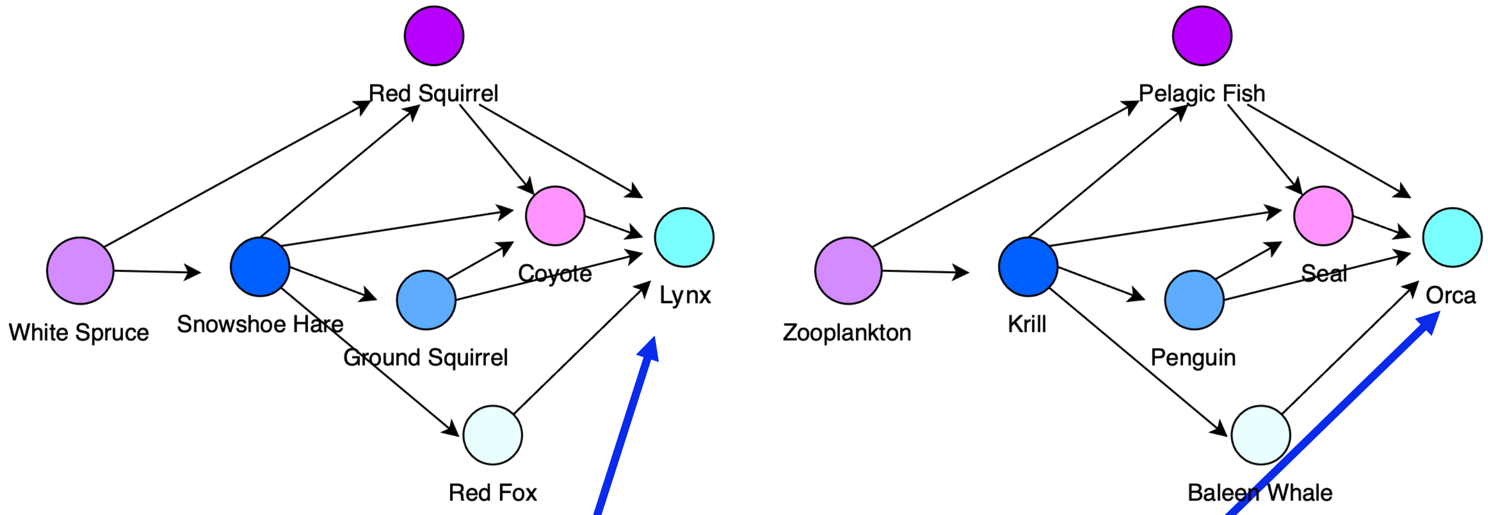
Colors = Left singular values

Equivalence to node permutations is the **defining characteristic** of node embeddings in graph representation learning

What defines graph representation learning then?

A Property of Equivariant Node Representations

Food Web Graph Example
(where land and sea have disconnected components)



(Srinivasan & R., ICLR 2020)

$$f_{\text{GNN}}(\boldsymbol{A})_{\text{lynx}} = f_{\text{GNN}}(\boldsymbol{A})_{\text{orca}}$$

- **Equivariant node representations are based on graph structure alone** (isomorphic nodes → same representations)

- **True for GNNs and true for all equivariant node representations**

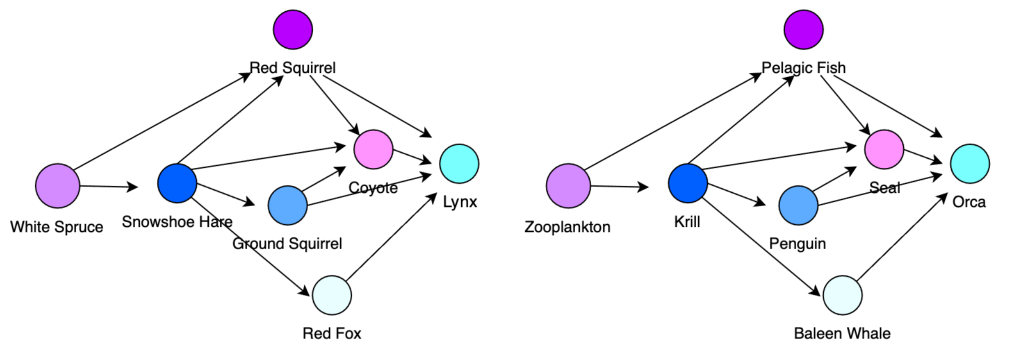
Types of Node Representations

What is the relationship between

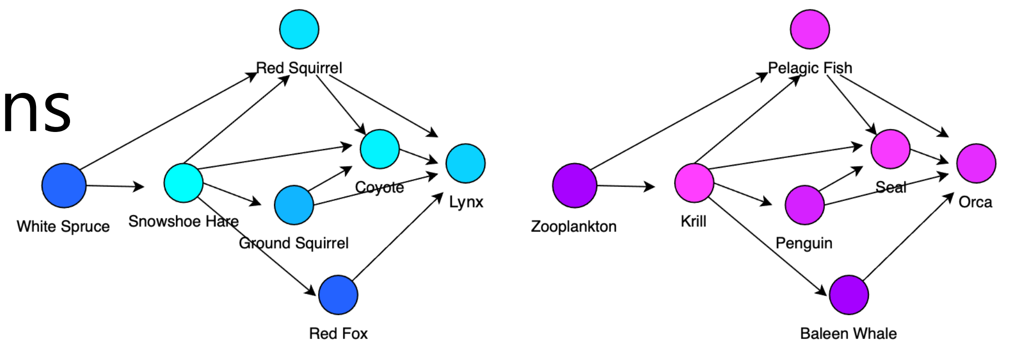
- Equivariant node representations (e.g. GNNs)

and

- Permutation-sensitive node representations (e.g., SVD’s left singular vectors)



Structural node representations



Positional node representations

Definition:

- **Positional node representation:** Any permutation-**sensitive** node representation (e.g. matrix factorization)

- **Structural node representation:** Any permutation-**insensitive** node representation
(e.g. GNN)

On the Equivalence between Positional Node Embeddings and Structural Graph Representations

Balasubramaniam Srinivasan
Department of Computer Science
Purdue University
bsriniv@purdue.edu

Bruno Ribeiro
Department of Computer Science
Purdue University
ribeiro@cs.purdue.edu

(Srinivasan & R., ICLR 2020)

See (Srinivasan & R., ICLR 2020) for details & notation

Theorem 1. *Let* $\mathcal{S} \subseteq \mathcal{P}^\star(V)$ *be a set of non-empty subsets of* V. *Let* $\boldsymbol{Y}(\mathcal{S}, \mathbf{A}, \boldsymbol{X}) = (Y(\vec{S}, \mathbf{A}, \boldsymbol{X}))_{S\in\mathcal{S}}$ *be a sequence of random variables defined over the sets* $S \in \mathcal{S}$ *of a graph* $G = (\mathbf{A}, \boldsymbol{X})$, *such that* $Y(\vec{S_1}, \mathbf{A}, \boldsymbol{X}) \overset{d}{=} Y(\vec{S_2}, \mathbf{A}, \boldsymbol{X})$ *for any two jointly isomorphic subsets* $S_1, S_2 \in \mathcal{S}$ *(Definition 7), where* $\overset{d}{=}$ *means equality in their marginal distributions. Then, there exists a measurable function* φ *such that,* $\boldsymbol{Y}(\mathcal{S}, \mathbf{A}, \boldsymbol{X}) \overset{a.s.}{=} (\varphi(\Gamma^\star(\vec{S}, \mathbf{A}, \boldsymbol{X}), \epsilon_S))_{S\in\mathcal{S}}$, *where* ϵ_S *is a pure source of random noise from a joint distribution* $p((\epsilon_{S'})_{\forall S'\in\mathcal{S}})$ *independent of* $\mathbf{A}$ *and* $\boldsymbol{X}$.

*Structural causal model (e.g. noise transfer theorem (Kallenberg 2006))

See (Srinivasan & R., ICLR 2020) for details & notation

Theorem 2 (The statistical equivalence between node embeddings and structural representations). *Let* $\boldsymbol{Y}(\mathcal{S}, \mathbf{A}, \boldsymbol{X}) = (Y(\vec{S}, \mathbf{A}, \boldsymbol{X}))_{S \in \mathcal{S}}$ *be as in Theorem 1. Consider a graph* $G = (\mathbf{A}, \boldsymbol{X}) \in \Sigma$. *Let* $\Gamma^\star(\vec{S}, \mathbf{A}, \boldsymbol{X})$ *be a most-expressive structural representation of nodes* $S \in \mathcal{P}^\star(V)$ *in* G. *Then,*

$$Y(\vec{S}, \mathbf{A}, \boldsymbol{X}) \perp\!\!\!\perp_{\Gamma^\star(\vec{S}, \mathbf{A}, \boldsymbol{X})} \boldsymbol{Z} | \mathbf{A}, \boldsymbol{X}, \quad \forall S \in \mathcal{S},$$

for any node embedding matrix $\boldsymbol{Z}$ *that satisfies Definition 12, where* $A \perp\!\!\!\perp_B C$ *means* A *is independent of* C *given* B. *The above is a consequence of the fact that, for any embedding distribution* $p(\boldsymbol{Z}|\mathbf{A}, \boldsymbol{X})$, *there exists a* $\mathcal{G}$*-equivariant function* ϕ *such that* $p(\boldsymbol{Z}|\mathbf{A}, \boldsymbol{X}) = \int_\epsilon \phi((\Gamma^\star(v, \mathbf{A}, \boldsymbol{X}))_{v \in V}, \epsilon) dP(\epsilon)$, *where* ϵ *is a pure noise source and* P *its associated Lebesgue measure. Finally,* $\forall (\mathbf{A}, \boldsymbol{X}) \in \Sigma$, *there exists a most-expressive node embedding* $\boldsymbol{Z}^\star | \mathbf{A}, \boldsymbol{X}$ *such that,*

$$\Gamma^\star(\vec{S}, \mathbf{A}, \boldsymbol{X}) = \mathbb{E}_{\boldsymbol{Z}^\star}[f^{(|S|)}((\boldsymbol{Z}^\star_v)_{v \in S}) | \mathbf{A}, \boldsymbol{X}], \quad \forall S \in \mathcal{S},$$

for some appropriate collection of functions $\{f^{(k)}(\cdot)\}_{k=1,\ldots,n}$.

Structural (node-equivariant) representations

$\Downarrow$

Positional (permutation-sensitive) node representations

Structural representations ⇒ Positional node representations

Consequences of Theorems 1 & 2 of (Srinivasan & R., ICLR 2020)

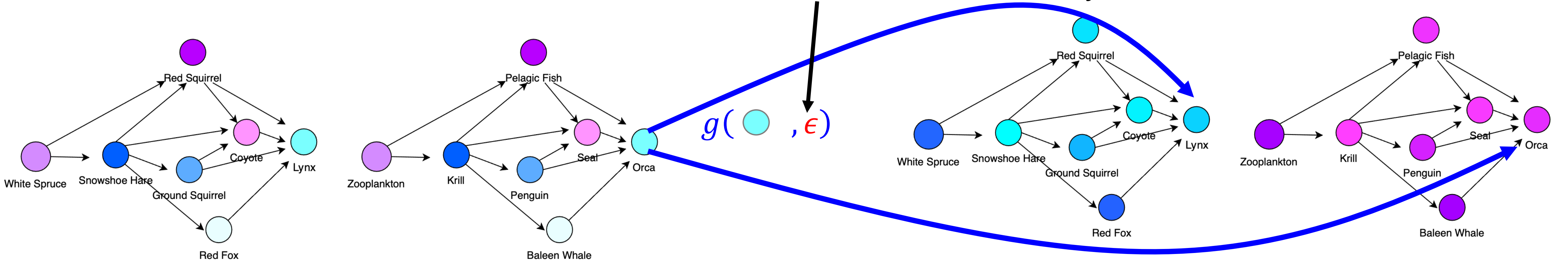
- If equivariant representation $f_{\text{struc}}(\boldsymbol{A})_v$ is most-expressive, then:
  - For any r.v. Y_v, $\exists g$ s.t.

$$Y_v | \boldsymbol{A}, v = g(f_{\text{struc}}(\boldsymbol{A})_v, \epsilon),^* \qquad \epsilon \sim \text{Uniform}(0,1)$$

- Example:
  - For the SVD's **left singular vector** Z_v for node v in adjacency matrix $\boldsymbol{A}$, $\exists g$ s.t.

$$Z_v | \boldsymbol{A}, v = g(f_{\text{struc}}(\boldsymbol{A})_v, \epsilon)$$

Permutation sensitivity comes from randomness



Structural node representations

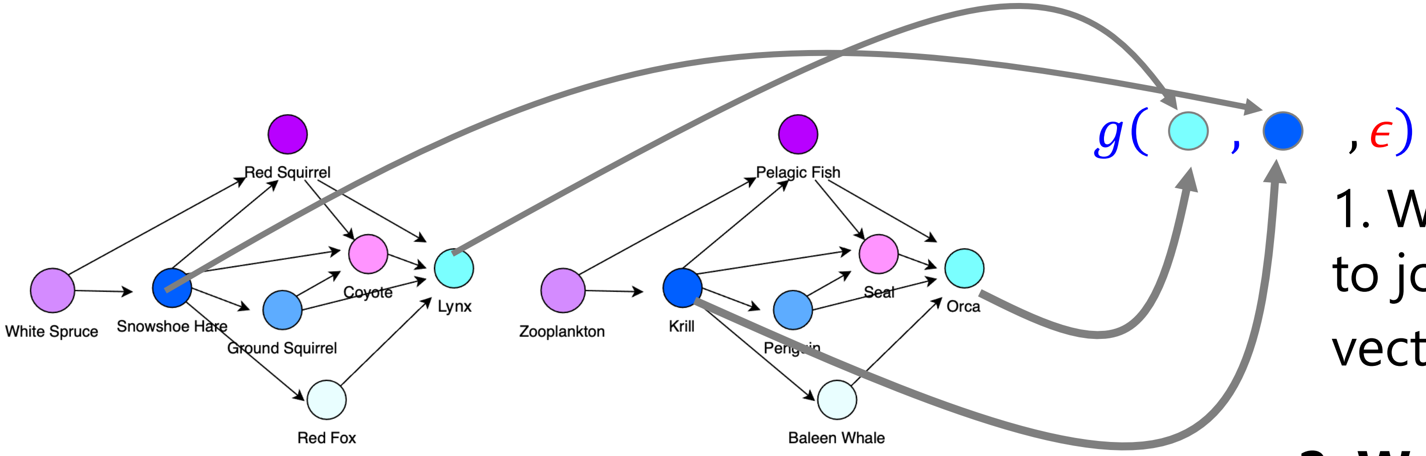
Positional node representations

Structural representations ⇒ Positional node representations

More consequences of Theorems 1 & 2 of (Srinivasan & R., ICLR 2020)

- Note $Z_v | \boldsymbol{A}, v = g(f_{\text{struc}}(\boldsymbol{A})_v, \epsilon)$ **is a marginal distribution over the nodes**
- The joint distribution Z_v, Z_u is harder to get:

$$Z_v, Z_u | \boldsymbol{A}, v, u \neq g(f_{\text{struc}}(\boldsymbol{A})_v, f_{\text{struc}}(\boldsymbol{A})_u, \epsilon)$$



Structural node representations

1. We will not be able to jointly obtain singular vectors from g

2. We will not be able to accurately predict links

Structural representations ⇒ Positional node representations

PURDUE UNIVERSITY

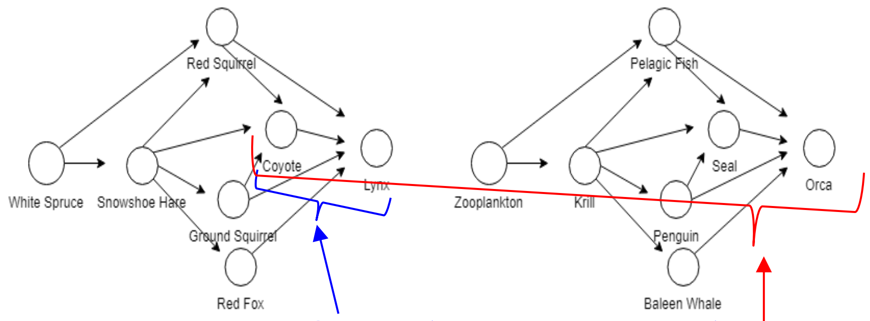
More consequences of Theorems 1 & 2 of (Srinivasan & R., ICLR 2020)

- Predictions over multiple nodes require a joint structural representation of those nodes

- **Consequence:**
  - To predict a hidden edge Y_{vu} we need a **JOINT** representation
    - Pairwise node representation **equivariant over edges, not nodes**

$$Y_{vu} | \boldsymbol{A}, v, u = g_2(f_{\text{struc}}(\boldsymbol{A}, v, u), \epsilon), \qquad \epsilon \sim \text{Uniform}(0,1)$$

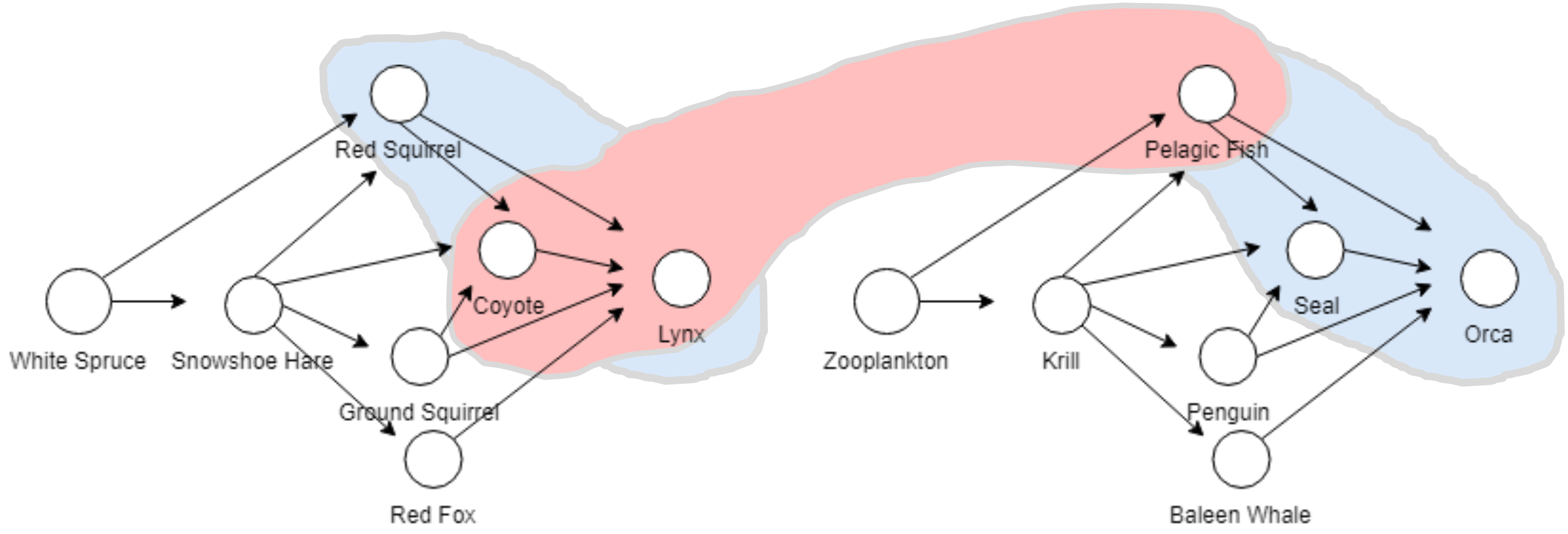
Joint k-node representations to predict properties of k nodes



$f_{\text{struc}}(\boldsymbol{A}, \text{lynx,coyote})$
$\neq f_{\text{struc}}(\boldsymbol{A}, \text{orca,coyote})$

Example: Joint structural k-node representations

- Example: Structural representations of 3 nodes



$$f^{(3)}_{\text{struct}}(A)_{\text{squirrel,coyote,lynx}} = f^{(3)}_{\text{struct}}(A)_{\text{fish,seal,orca}} \neq f^{(3)}_{\text{struct}}(A)_{\text{fish,coyote,lynx}}$$

Positional node representations $\Rightarrow$ Structural representations

Averaging: Positional node representations ⇒ Structural representations

- **Positional representations of k nodes** are to most-expressive **k-node structural representations**

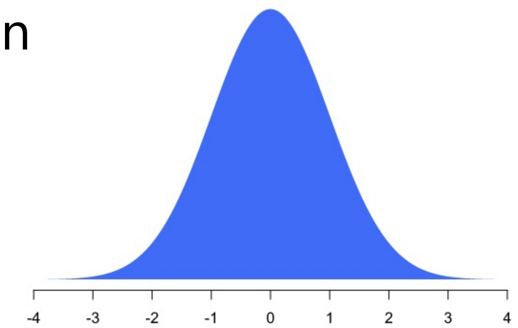
  as

  Samples of a distribution are to sufficient statistics of the distribution

Statistical implications (an analogy)

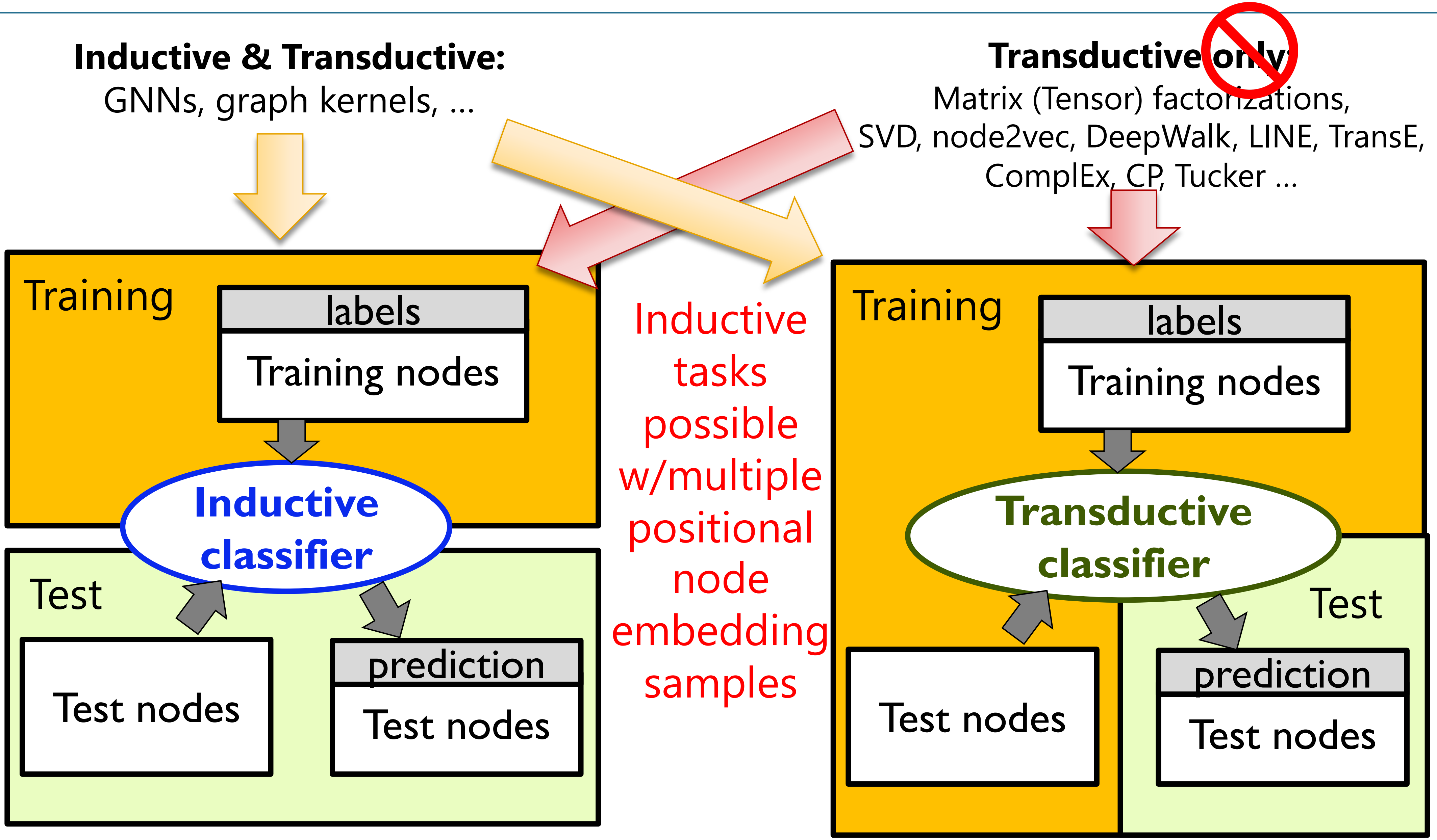
- Suppose Z_1, Z_2, … ~ Normal(μ, σ), sampled independently
- Assume $f(Z_1, Z_2, \dots)$ is a most-expressive permutation-invariant representation

  Then, $\hat{\mu}, \hat{\sigma} \Leftrightarrow f(Z_1, Z_2, \dots)$, where $\hat{\mu}, \hat{\sigma}$ are sufficient statistic of $Z_1, Z_2, \dots$
- Using a single sample Z_1 we cannot estimate $\hat{\sigma}$



Same happens with positional node representations:
Single sample not good enough for some tasks

Example: Inductive vs transductive classifiers on graphs



This theoretical framework explains results in the literature

- Use of **unique node IDs as node features** in **GNNs** breaks equivariance
  - Make representations depend on node ID permutation (i.e., positional)
  - With unique IDs, GNN node representation is positional and predicts links well

Variational Graph Auto-Encoders

Thomas N. Kipf, University of Amsterdam, T.N.Kipf@uva.nl
Max Welling, University of Amsterdam, Canadian Institute for Advanced Research (CIFAR), M.Welling@uva.nl

Position-aware Graph Neural Networks

Jiaxuan You[1] Rex Ying[1] Jure Leskovec[1]

On Positional and Structural Node Features for Graph Neural Networks on Non-attributed Graphs

Hejie Cui[1,*], Zijie Lu[2,*], Pan Li[3], and Carl Yang[1,†]

Rethinking Graph Transformers with Spectral Attention

Devin Kreuzer*, McGill University, Mila, Montreal, Canada, devin.kreuzer@mail.mcgill.ca
Dominique Beaini*, Valence Discovery, Montreal, Canada, dominique@valencediscovery.com
William L. Hamilton, McGill University, Mila, Montreal, Canada, wlh@cs.mcgill.ca
Vincent Létourneau, University of Ottawa, Ottawa, Canada, vletour2@uottawa.ca
Prudencio Tossou, Valence Discovery, Montreal, Canada, prudencio@valencediscovery.c

A Generalization of Transformer Networks to Graphs

Vijay Prakash Dwivedi,[¶] Xavier Bresson[¶]
[¶] School of Computer Science and Engineering, Nanyang Technological University, Singapore

Benchmarking Graph Neural Networks

Vijay Prakash Dwivedi[1*] vijaypra001@e.ntu.edu.sg
Chaitanya K. Joshi[1*] chaitanya.joshi@ntu.edu.sg
Thomas Laurent[2] tlaurent@lmu.edu
Yoshua Bengio[3,4] yoshua.bengio@mila.quebec
Xavier Bresson[1] xbresson@ntu.edu.sg

Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction

Zhaocheng Zhu[1,2], Zuobai Zhang[1,2], Louis-Pascal Xhonneux[1,2], Jian Tang[1,3,4]

  - Can also build more expressive equivariant representations
    - Through positional ⇒ structural representations (via averaging positional representations)

Relational Pooling for Graph Representations 2019

Ryan L. Murphy[1] Balasubramaniam Srinivasan[2] Vinayak Rao[1] Bruno Ribeiro[2]

Building powerful and equivariant graph neural networks with structural message-passing 2020

Clément Vignac, Andreas Loukas, and Pascal Frossard
EPFL

A Collective Learning Framework to Boost GNN Expressiveness for Node Classification 2021

Mengyue Hang[1] Jennifer Neville[1] Bruno Ribeiro[1]

How hard is to distinguish graphs with graph neural networks? 2020

Andreas Loukas
École Polytechnique Fédérale Lausanne

Can Graph Neural Networks Count Substructures? 2021

Zhengdao Chen, New York University, zc1216@nyu.edu
Lei Chen, New York University, lc3909@nyu.edu
Soledad Villar, Johns Hopkins University, soledad.villar@jhu.edu
Joan Bruna, New York University, bruna@cims.nyu.edu

And many more

Quest to build joint structural representations

Link Prediction Based on Graph Neural Networks

Muhan Zhang
Department of CSE
Washington University in St. Louis

Yixin Chen
Department of CSE
Washington University in St. Louis

Subgraph Pattern Neural Networks for High-Order Graph Evolution Prediction

Changping Meng, S Chandra Mouli, Bruno Ribeiro, Jennifer Neville

Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks

Cristian Bodnar[*1] Fabrizio Frasca[*2 3] Yu Guang Wang[*4 5 6]
Nina Otter[7] Guido Montúfar[*4 7] Pietro Liò[1] Michael M. Bronstein[2 3]

Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings

Christopher Morris[*] Gaurav Rattan[†] Petra Mutzel[‡]

Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks

Christopher Morris,[1] Martin Ritzert,[2] Matthias Fey,[1] William L. Hamilton,[3]
Jan Eric Lenssen,[1] Gaurav Rattan,[2] Martin Grohe[2]

Unsupervised Joint k-node Graph Representations with Compositional Energy-Based Models

Leonardo Cotta[*]
Purdue University
cotta@purdue.edu

Carlos H. C. Teixeira
Universidade Federal de Minas Gerais, Brazil
carlos@dcc.ufmg.br

Ananthram Swami
United States Army Research Laboratory
ananthram.swami.civ@mail.mil

Bruno Ribeiro
Purdue University
ribeiro@cs.purdue.edu

On the Universality of Invariant Networks

Haggai Maron[1] Ethan Fetaya[2 3] Nimrod Segol[1] Yaron Lipman[1]

Neural Higher-order Pattern (Motif) Prediction in Temporal Networks

Yunyu Liu
Purdue University
LIU3154@PURDUE.EDU

Jianzhu Ma
Peking University
MAJIANZHU@PKU.EDU.CN

Pan Li
Purdue University
PANLI@PURDUE.EDU

Classifying Graph Neural Networks

- **Positional or Structural?**

  - **Positional or Structural is the most important characteristic of a graph representation learning model**

    - Positional $\Longrightarrow$ Sensitive to node ID permutations
      - Can predict properties of any subset of nodes but prediction (actually, intermediate representations) should be averaged over multiple samples

    - Structural $\Longrightarrow$ Equivariant to node ID permutations
      - Can only predict properties of a **node** or the **entire graph**, but nothing* in-between

  - Everything else is less important to know:
    - Model expressiveness is not as important as type of node embedding
    - If GNN is message-passing or not is not as important as type of node embedding

* In certain tasks, with large subset sizes, the reconstruction conjecture kicks in and it may be possible using smaller k-node representations

Now that we understand node representation uses and limitations…

… are there other fundamental limitations in graph representation learning?

Predicting the future of a temporal graph can be just observational

Gao & R., On the Equivalence Between Temporal and Static Equivariant Graph Representations for Observational Predictions, arXiv:2103.07016, 2021.

Temporal Graph Representation Learning is **Observational**

PURDUE UNIVERSITY

(Gao & R., 2021) describes the theory of temporal graph representation learning

- Modeling time evolution unrelated to modeling cause and effect

- Classifies temporal graph representation learning:
  1. Time-and-graph methods
  2. Time-then-graph methods

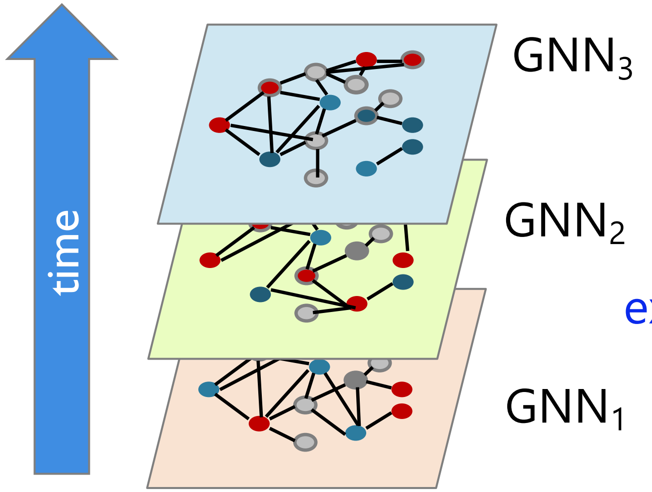
  - In general, **time-and-graph** and **time-then-graph** are equally expressive

  - Using standard GNNs, time-then-graph are more expressive than time-and-graph

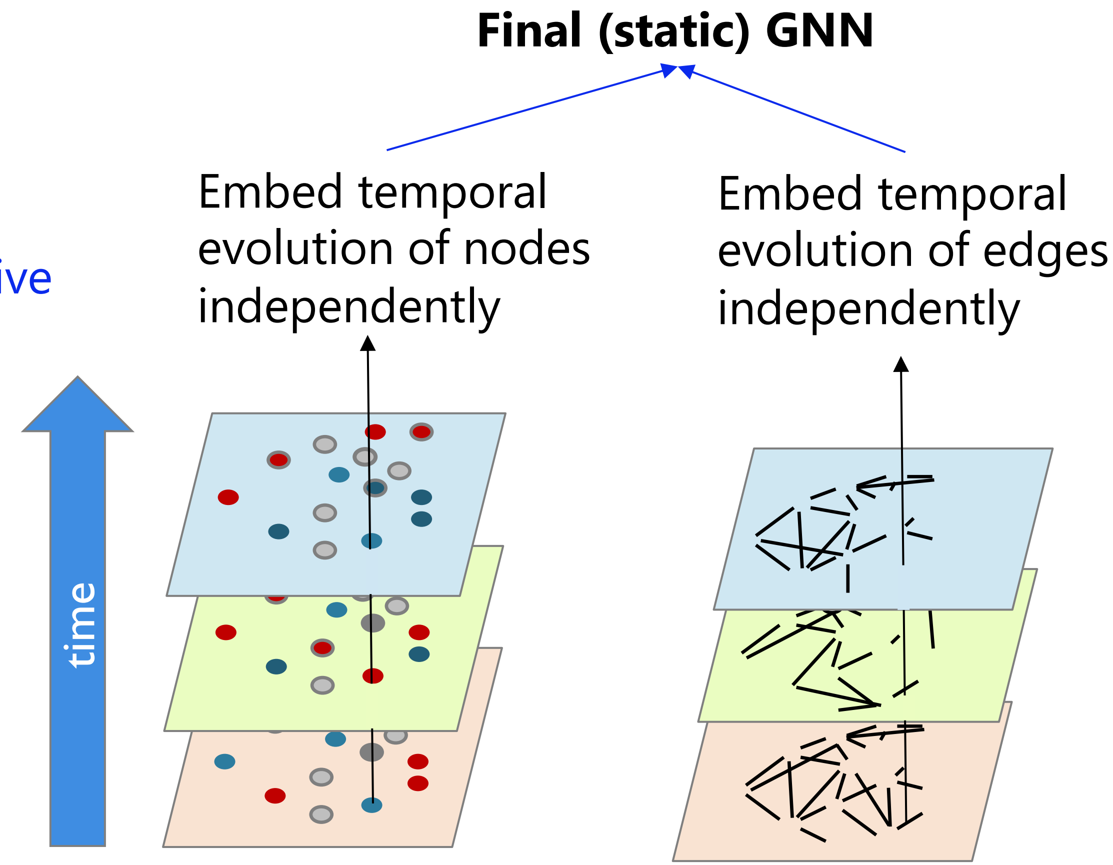
Time-then-graph more expressive than Time-and-graph (when using GNNs)

- **Time-and-graph**
  - Encodes how node embeddings evolve over time
  - Majority of existing works

- **Time-then-graph**
  - Embedding encodes time evolution of nodes and edges independently
  - Impose permutation-equivariance via final static graph

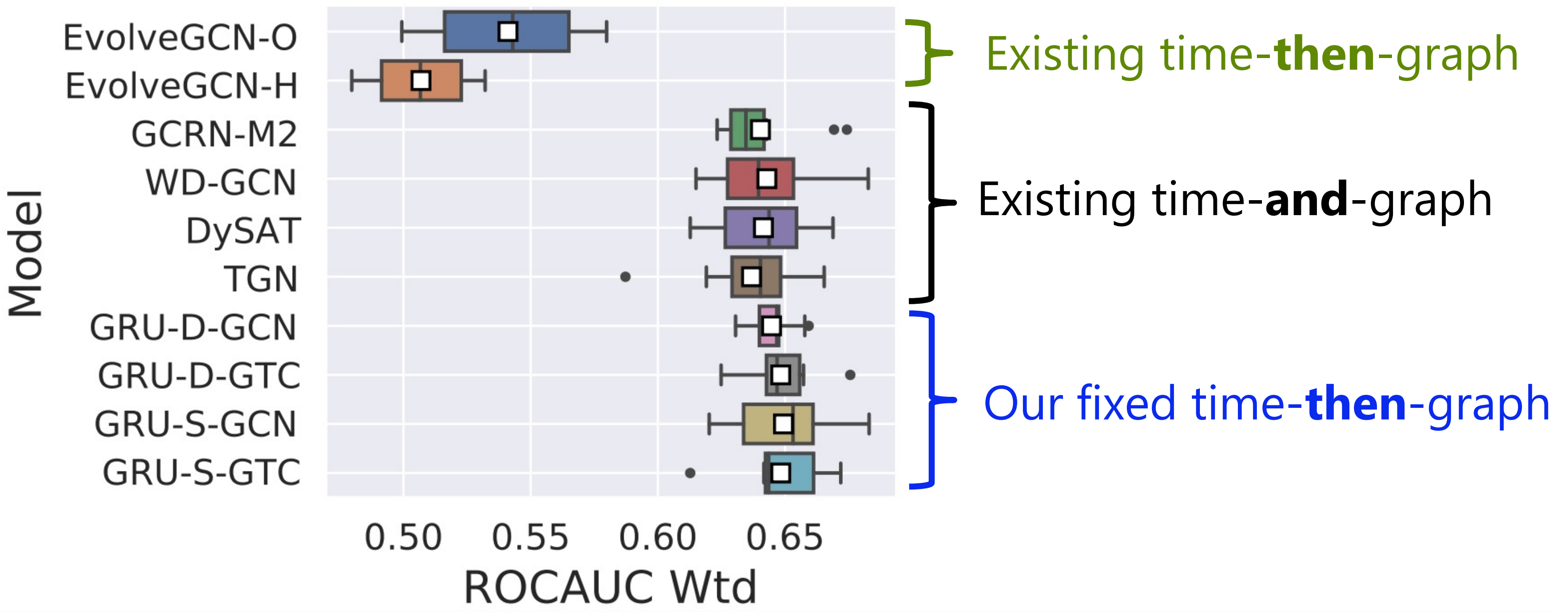


Less expressive < More expressive



Example: COVID-19 Observational Predictions

Task: Predict if a node will get infected
- **Input:** Temporal graph and epidemic evolution (discretized in time)
- **Output:** Probability a node gets infected in next step



Shows Temporal-GNNs predictions can be purely observational: Predicts infections without modeling how virus spreads over the graph

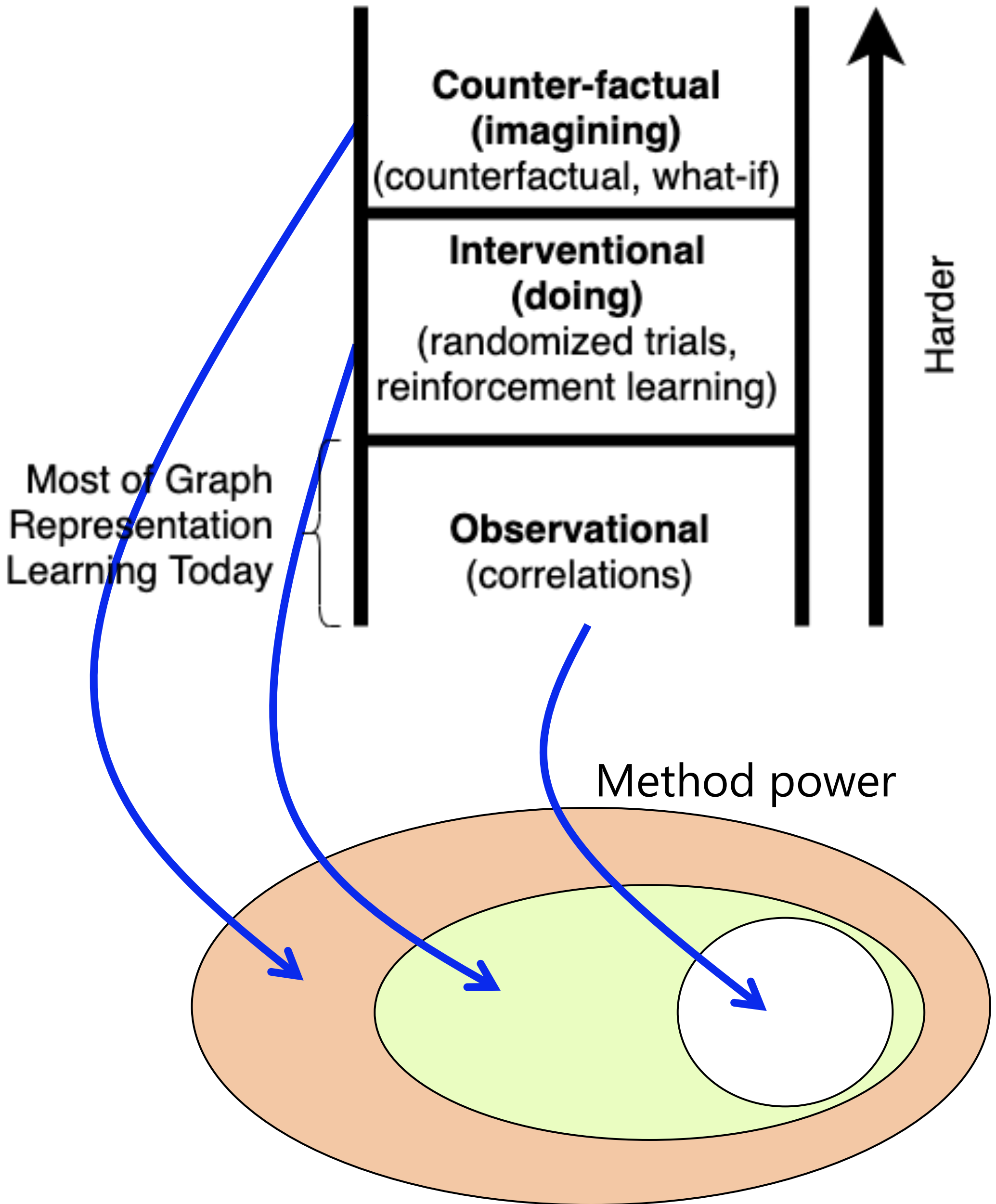
Take-home: Temporal GNNs are not designed to learn causal models (even if just Granger causality)

The Ladder of Causality

- Graph Representation Learning today is designed for **observational tasks**

- **The hierarchy of causality:** Lower-levels methods incapable of higher-level tasks
  - **Counterfactua**l methods can perform **all** tasks
  - **Interventional** methods can **also** perform **observational** tasks
  - **Observational** methods **can only** perform observational tasks

- **Rest of the talk:**
  Counterfactual Graph Representation Learning



Counterfactual Graph Representation Learning: Two Findings

1. Out-of-distribution generalization from single training environment is possible via G-invariances
   - But counterfactual G-invariances are stronger (more invariant) than G-invariance

NEURAL NETWORKS FOR LEARNING COUNTERFACTUAL G-INVARIANCES FROM SINGLE ENVIRONMENTS

S Chandra Mouli
Department of Computer Science
Purdue University

Bruno Ribeiro
Department of Computer Science
Purdue University

2. Out-of-distribution generalization w.r.t. graph sizes without test examples
   - Possible if GNN is coupled with a stable property as graphs grow

Size-Invariant Graph Representations for Graph Classification Extrapolations

Beatrice Bevilacqua [*1] **Yangze Zhou** [*2] **Bruno Ribeiro** [1]

Counterfactual Graph Representation Learning: Two Findings

1. Out-of-distribution generalization from single training environment is possible via G-invariances
    - But counterfactual G-invariances are stronger (more invariant) than G-invariance

NEURAL NETWORKS FOR LEARNING COUNTERFACTUAL G-INVARIANCES FROM SINGLE ENVIRONMENTS

S Chandra Mouli
Department of Computer Science
Purdue University

Bruno Ribeiro
Department of Computer Science
Purdue University

2. Out-of-distribution generalization w.r.t. graph sizes without test examples
    - Possible if GNN is coupled with a stable property as graphs grow

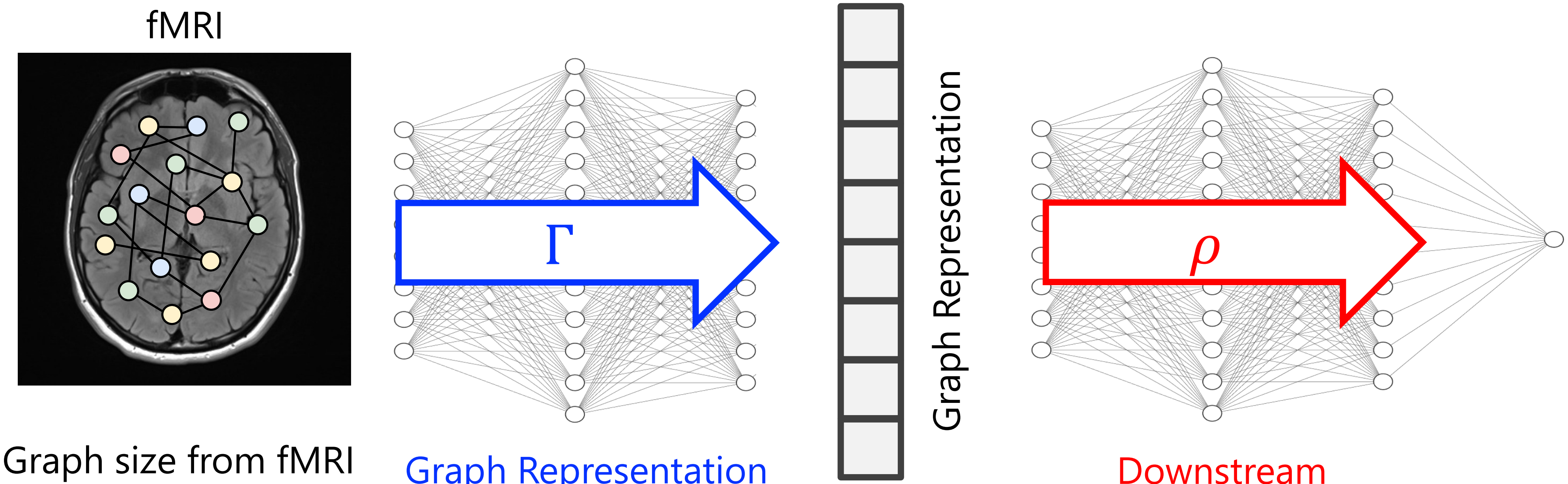
Size-Invariant Graph Representations for Graph Classification Extrapolations

Beatrice Bevilacqua [*1] Yangze Zhou [*2] Bruno Ribeiro [1]

Out-of-distribution Generalization w.r.t. Graph Sizes

Setting: Graph Classification Task

fMRI



Predicted property: Schizophrenic person?

Graph size from fMRI is a hyperparameter

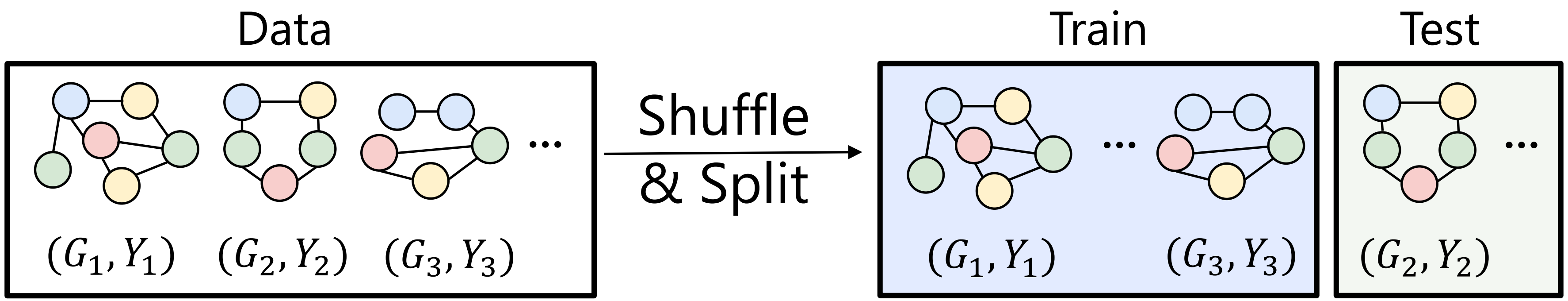
Graph Representation Learning

Downstream classifier

Current Graph Classification Approach

Graph Representation Learning generally assumes:

Train distribution = Test distribution

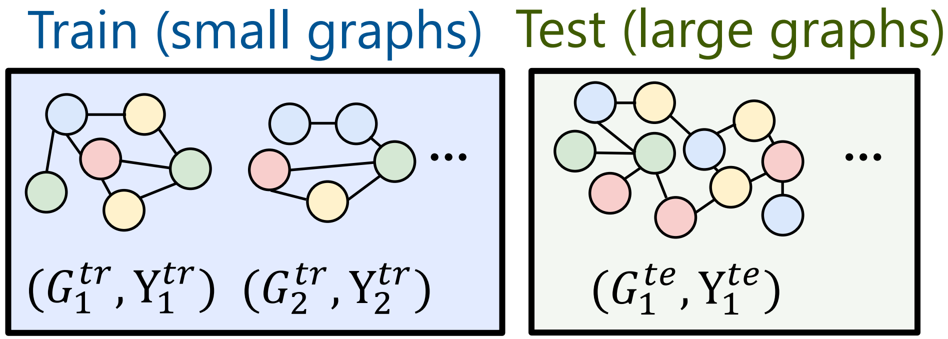


What if test data were out of distribution (OOD)?

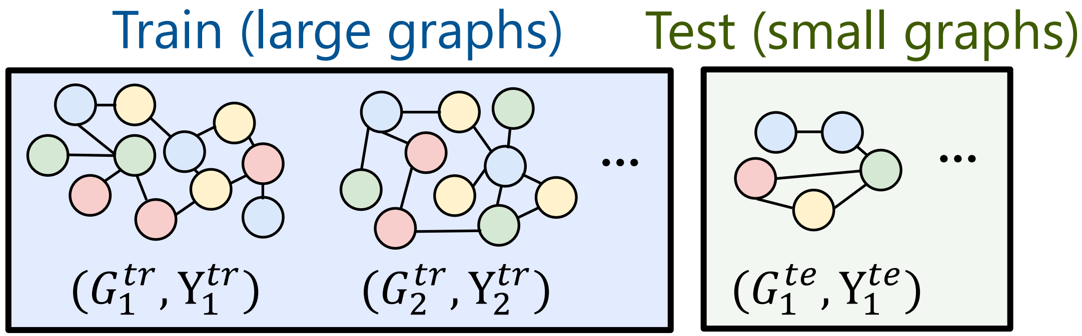
Bevilacqua, Zhou, R., ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

Extrapolation to Different Graph Sizes

What if train has **small** graphs but test has **large** graphs?



What if train has **large** graphs but test has **small** graphs?



Bevilacqua, Zhou, R., ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

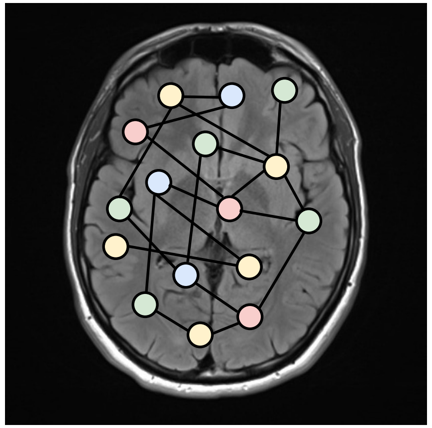
Size Extrapolation with GNNs?

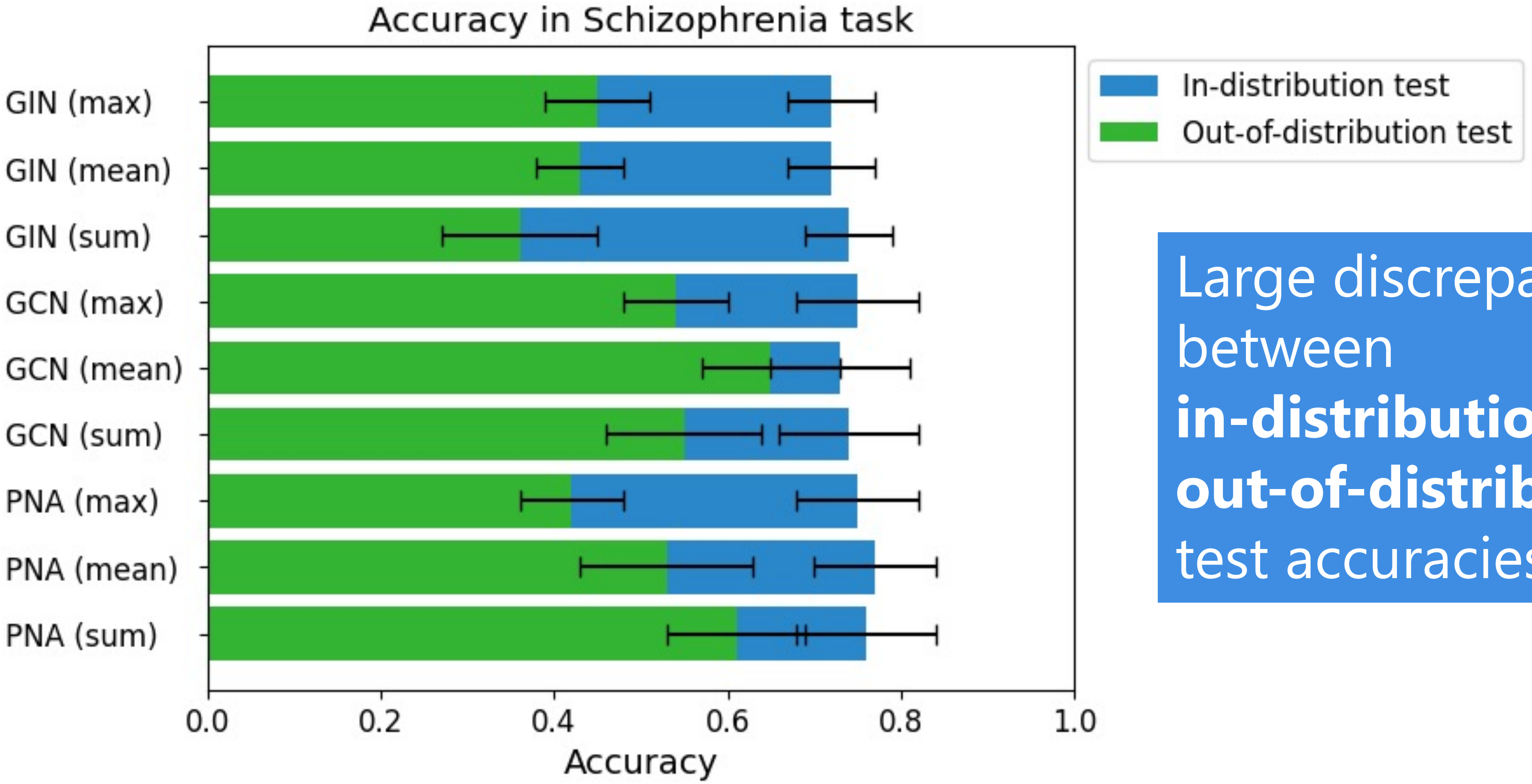
Do Graph Neural Networks (GNNs) extrapolate?

⇒ GNNs can be applied to graphs of any size

⇒ But may not extrapolate between **small (train)** and **large (test)** graphs:

Schizophrenia task





Large discrepancy between **in-distribution** and **out-of-distribution** test accuracies

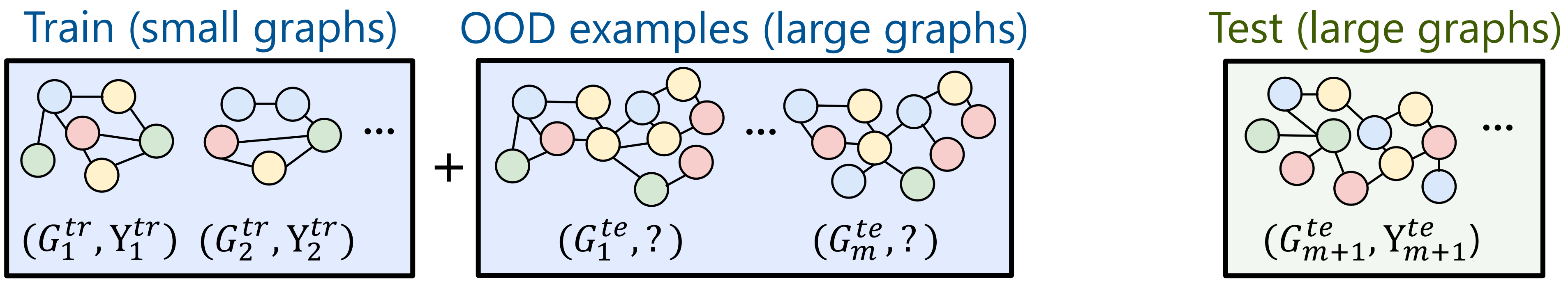
Bevilacqua, Zhou, R., ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

How to Extrapolate in Graph Classification Tasks?

How do we extrapolate beyond the training distribution?

If OOD examples available, **data-driven methods** work:

- Domain Adaptation
- Covariate Shift Adaptation
- Few-shot Learning
- Data Augmentation
- Invariant Risk-Minimization (IRM)*



How to Extrapolate in Graph Classification Tasks?

Data-driven methods:

Pros	Cons
Can use existing GNN methods Don’t assume a mechanism for distribution shift	Must have OOD examples during training

What if no access to OOD data?

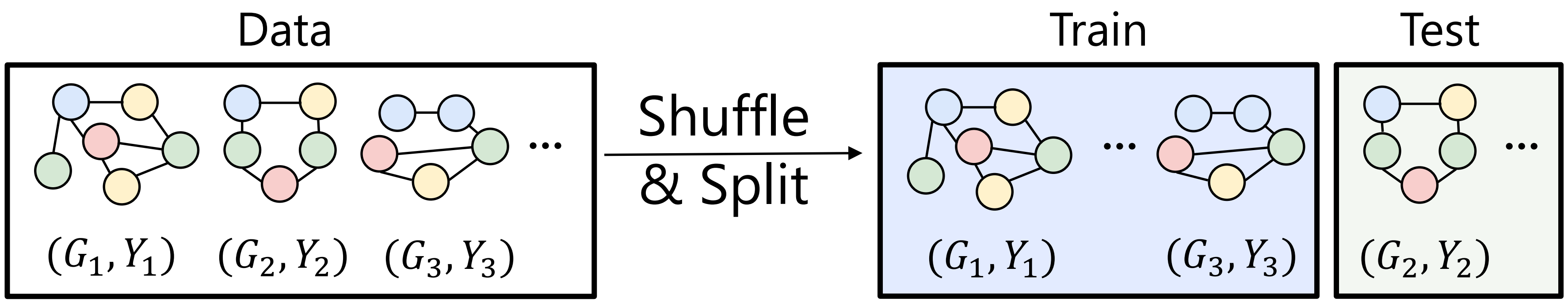
- Must define a causal mechanism

Next: Observational vs Causal (Counterfactual) modeling

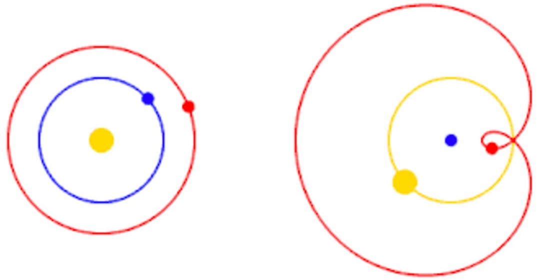
Bevilacqua, Zhou, R., ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

Differences between Observational and Counterfactual Tasks

Observational Task:
Predicting unseen examples of training distribution

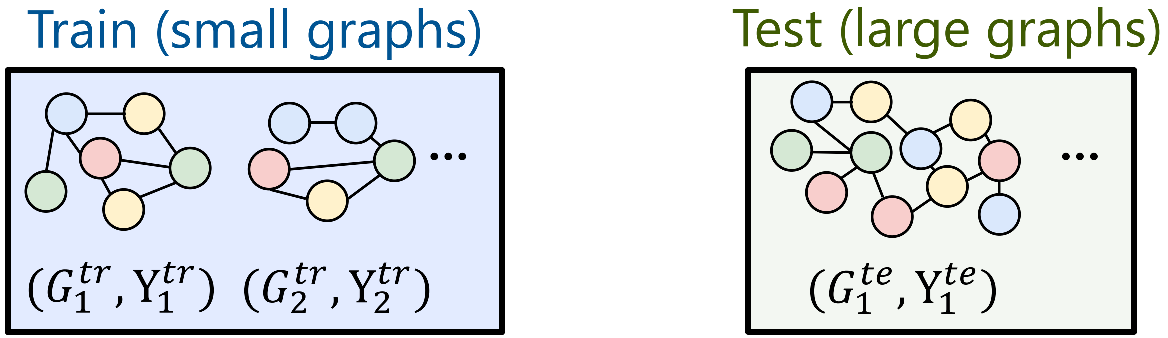


Planetary Motion Equivalent

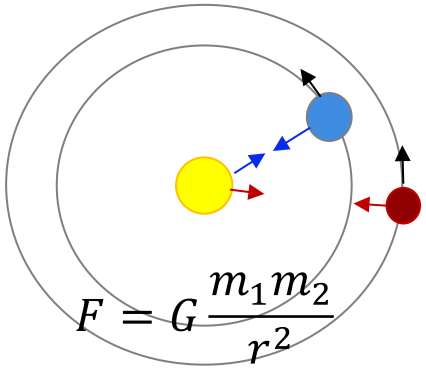


Kepler's law works
Epicycles work

Counterfactual Task (since we have no access to test data)**:**
What would be the label of a graph if it were larger?



or vice-versa



Newton's laws

Bevilacqua, Zhou, Ribeiro, ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

What would be the labels if the graphs were larger?

Assuming a Graph Mechanism

Q: What would be the label if the graph were infinitely large?

$$N \to \infty$$

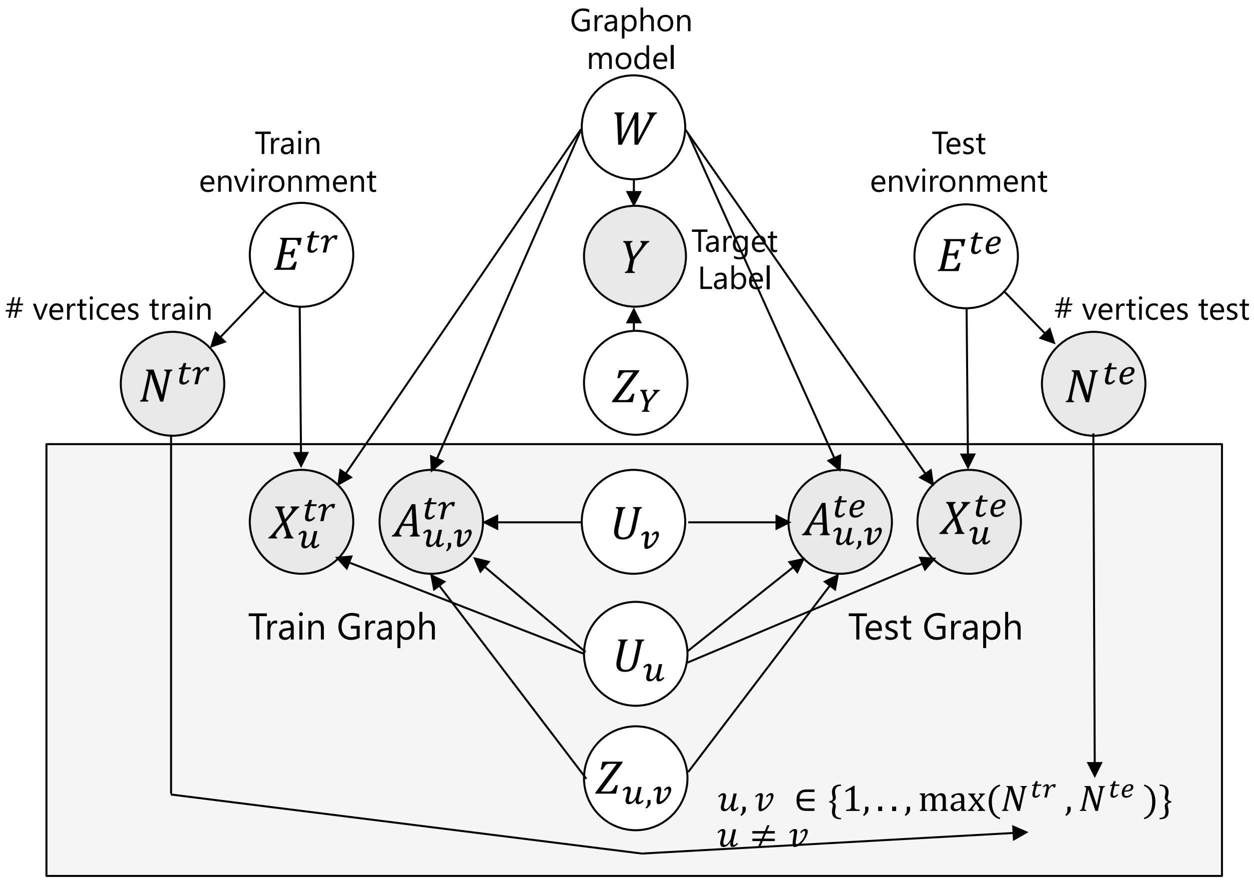
Lovász & Szegedy (2006) shows that for some graph families (graphons) there are features invariant to size

Build GNN based on these features

Paper: https://proceedings.mlr.press/v139/bevilacqua21a
Slides: https://www.cs.purdue.edu/homes/ribeirob/pdf/Bevilacqua_Zhou_ICML2021_slides.pdf

Assumes Causal Mechanism

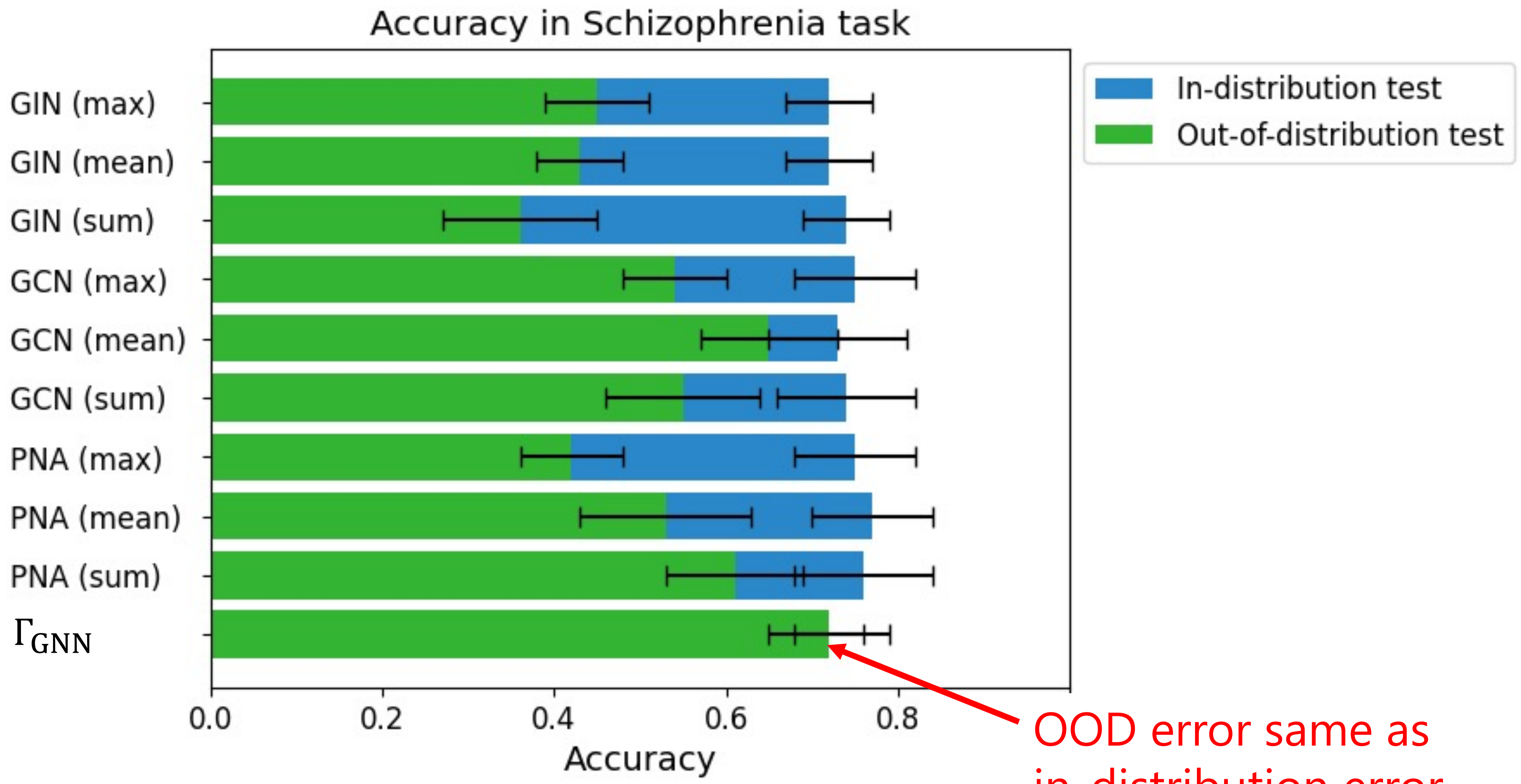
- *Structural Causal Model:*
  - Graph label Y is a function of the graph model W + some random noise
  - Graph size N^{tr} (N^{te}) is a function of "environment" E^{tr} (E^{te}) only
  - Train (test) graphs are generated by W and E^{tr} (E^{te}) with same random noises



Bevilacqua, Zhou, Ribeiro, ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

OOD Error in Schizophrenia Task

- Can Lovász & Szegedy's inspired GNN (Γ_{GNN}) extrapolate OOD?



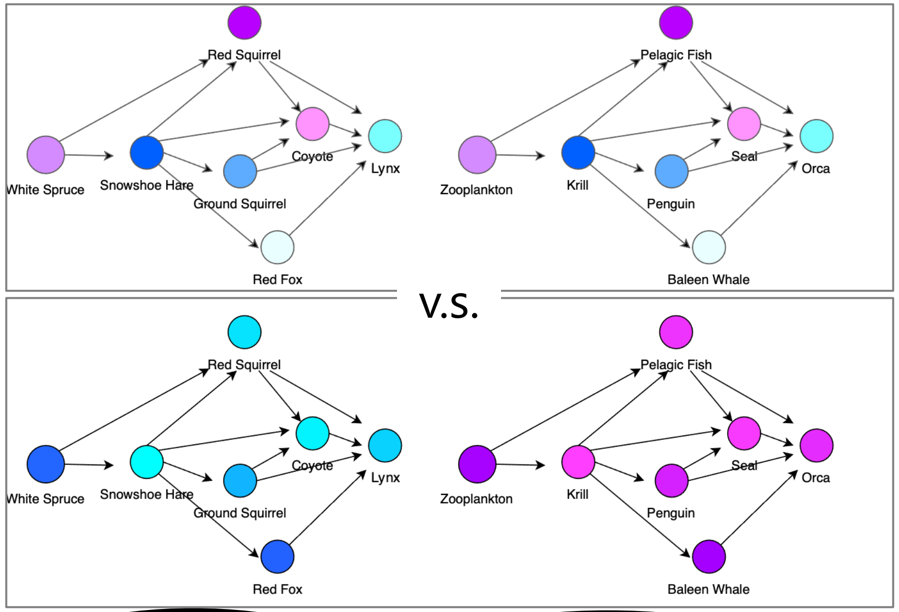
OOD error same as in-distribution error

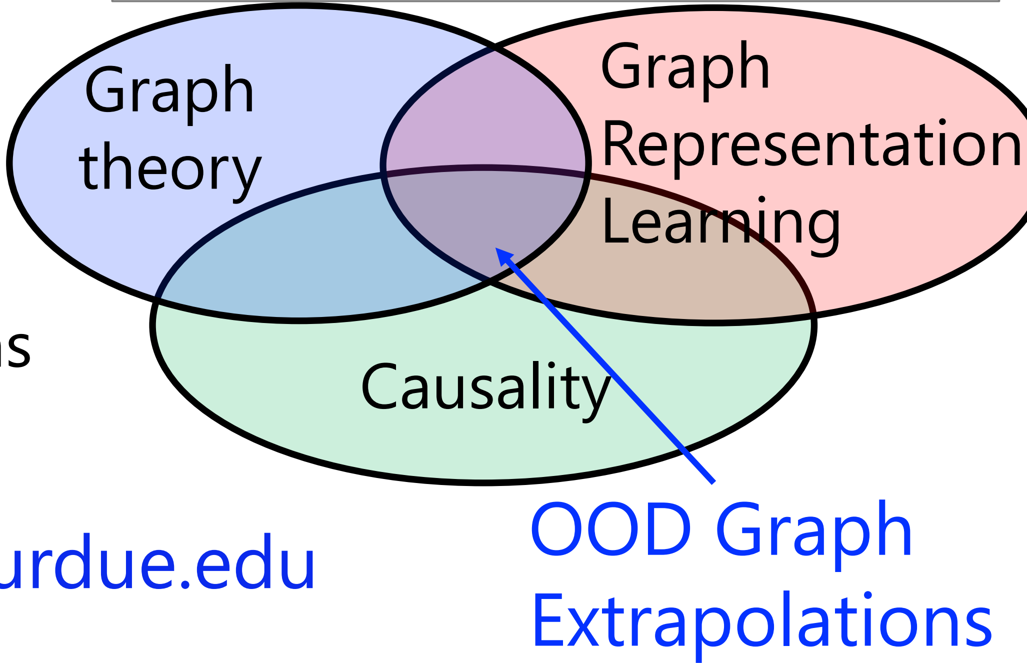
Bruno Ribeiro

Bevilacqua, Zhou, Ribeiro, ICML 2021, Size-Invariant Graph Representations for Graph Classification Extrapolations

Take-home

- Graph representation learning:
  - Fundamental difference:
    - Positional vs Structural node representation
  - Structural representations have limitations in joint predictions
  - Positional representations rely on Monte Carlo averaging for some tasks
- **Graph representation learning** methods are **observational** not **causal**
  - OOD without test data examples requires a counterfactual model
  - There are no universal OOD graph representations





Thank you!

@brunofmr
ribeiro@cs.purdue.edu