Mathematical Foundations of Knowledge Graph Foundation Models

Bruno Ribeiro
Department of Computer Science
Purdue University
**work partially performed during sabbatical at Stanford*

Joint Mathematics Meetings, 2025
AMS Special Session on Knowledge Graphs

A Mathematical Framework for Graph Foundation Models

Bruno Ribeiro[1*]

[1*]Department of Computer Science, Purdue University,
405 Stadium Mall Drive, West Lafayette, 47907, Indiana, USA.

Corresponding author(s). E-mail(s): ribeirob@purdue.edu;

Abstract

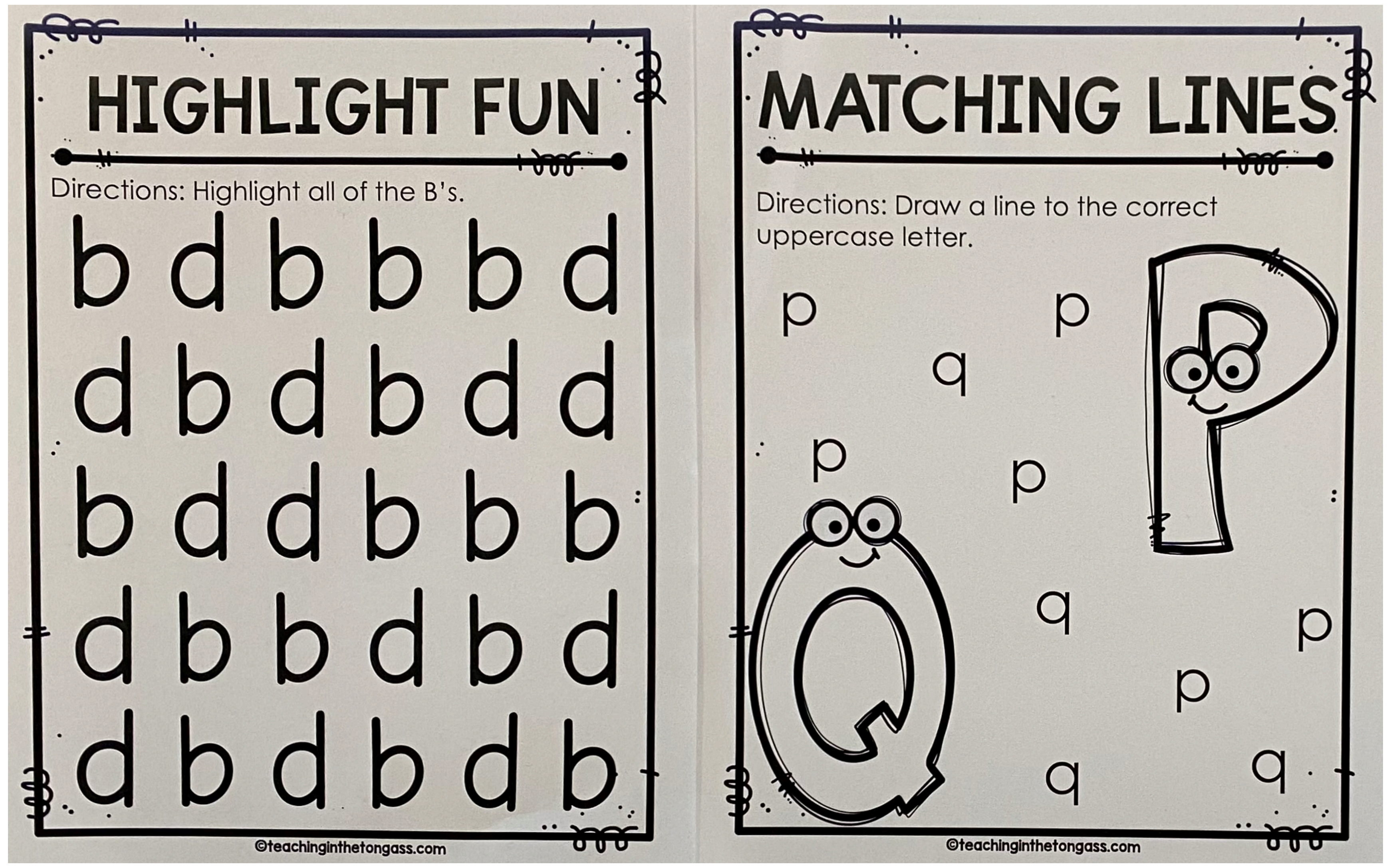
The deluge of sequential text data has been a boon for artificial intelligence, but it may be dwarfed by a largely untapped reservoir of human knowledge: graph-structured data, which underlies the Web's topology and the relational databases that govern our digital lives. Yet, despite its ubiquity, learning universal AI models for graph data remains a stubborn challenge. Here, we take up the task of establishing fundamental mathematical frameworks to facilitate the development of AI models that can effectively learn useful patterns from diverse graph-structured data. We will explore the theoretical and practical hurdles for tackling these unique challenges.

In preparation

or

Can Graph Neural Networks Learn to Generalize Beyond their Training Domains through extra Architectural Symmetries?

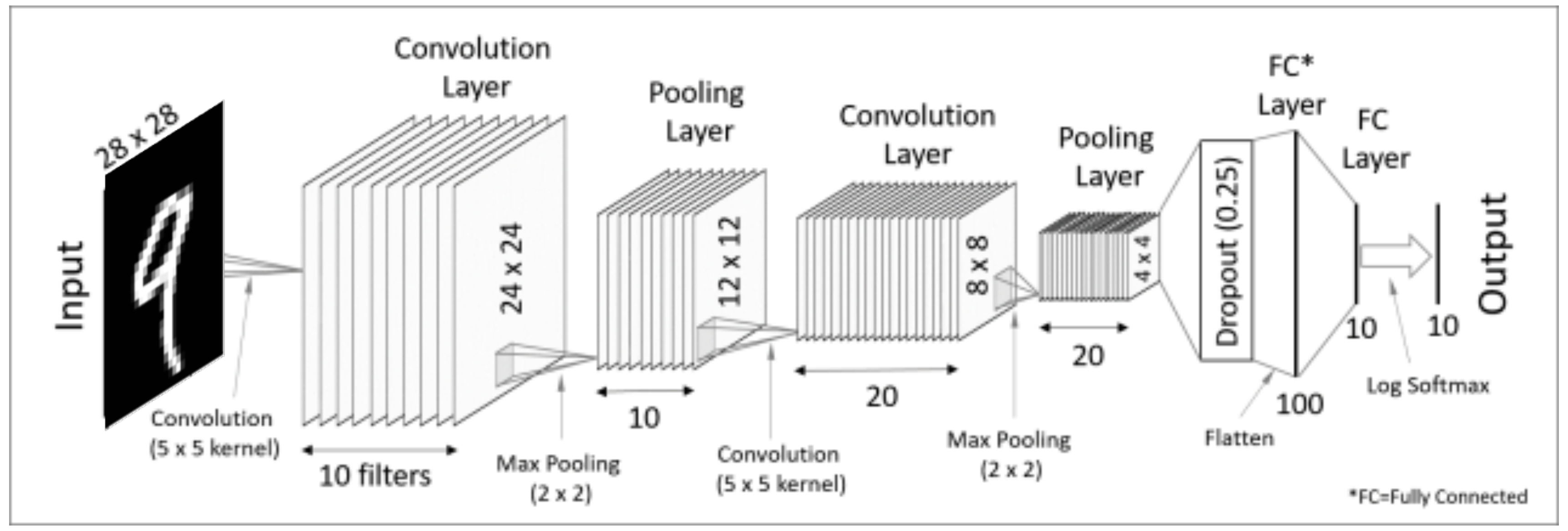
Symmetries in Human Learning



My son's kindergarten homework

[Sidman and Tailby, 1982]
[Sidman et al., 1982, "A search for symmetry in the conditional discriminations of rhesus monkeys, baboons, and children"]

VS



Any simple CNN can distinguishing 6 and 9 even before training starts

Talk Overview

- Knowledge Graphs
- Graph Foundation Models Desiderata
- The Symmetries of Graphs and Graph Tasks
- A Solution to the Diversity in Graph Task Symmetries
- A Solution to the Diversity of Graph Attributes
- Parting Thoughts

Knowledge Graph Adjacency Tensor

- Nodes = entities
- Edge type = type of relation

$\mathbf{A} \in \mathbb{Z}^{n \times n \times p}$

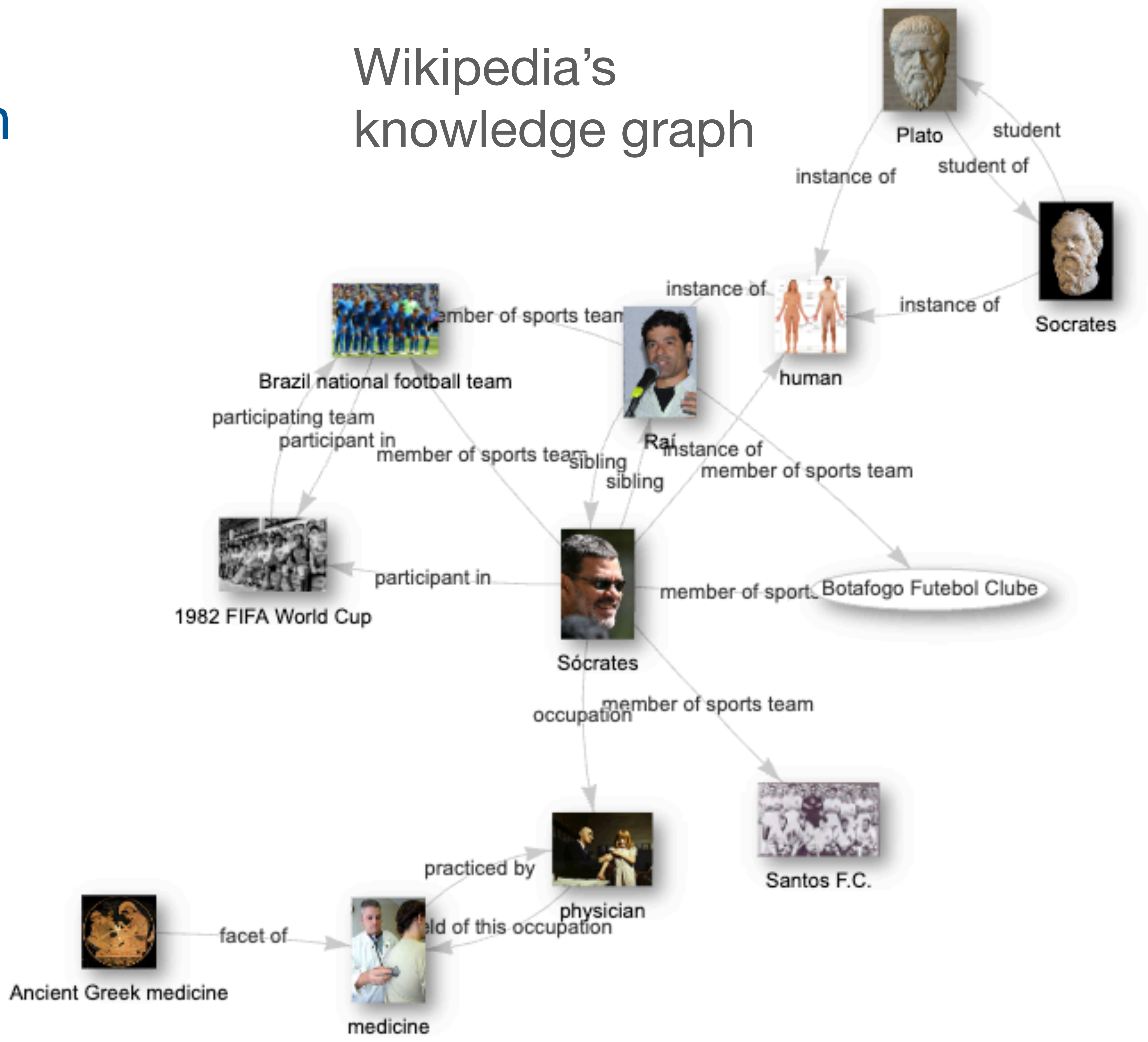
Adjacency tensor:

- If $p = 1$ relation have ids, special value means no relation

- If p = <number of relations> relations are one-hot encoded

More generally, we will encode other node features and edge features in dimension p

Wikipedia's knowledge graph



What is a “Graph Foundation Model”?

What is a "~~Graph~~ Foundation Model"?

A Definition of a *Foundation Model*

- **Massive Heterogeneous Training Data**: Foundation models are trained on enormous **heterogeneous** datasets, which can include text, code, and time series.

- **Transfer Learning**: They have the ability to transfer knowledge gained from **one task/domain** to **other tasks/domains**, often through pre-training on general-purpose objectives

- **Self-Supervised Learning**: Foundation models often leverage self-supervised learning techniques, allowing them to learn meaningful representations from unlabeled data.

- **Broad Task Applicability**: Due to their flexible nature, foundation models can be applied to a wide range of tasks/domains

A Definition of a *Graph Foundation Model*

- **Massive Heterogeneous Training Data**: Graph foundation models are trained on enormous **heterogeneous** graph datasets, which can include text, code, and time series.

- **Transfer Learning**: They have the ability to transfer knowledge gained from **one** graph **task/domain** to **other** graph **tasks/domains**, often through pre-training on general-purpose objectives

- **Self-Supervised Learning**: Graph foundation models often leverage self-supervised learning techniques, allowing them to learn meaningful representations from unlabeled data.

- **Broad Task Applicability**: Due to their flexible nature, graph foundation models can be applied to a wide range of graph tasks/domains

Desiderata for Graph Foundation Models

Desiderata for Graph Foundation Models (GFMs)

Minimal Requirement:

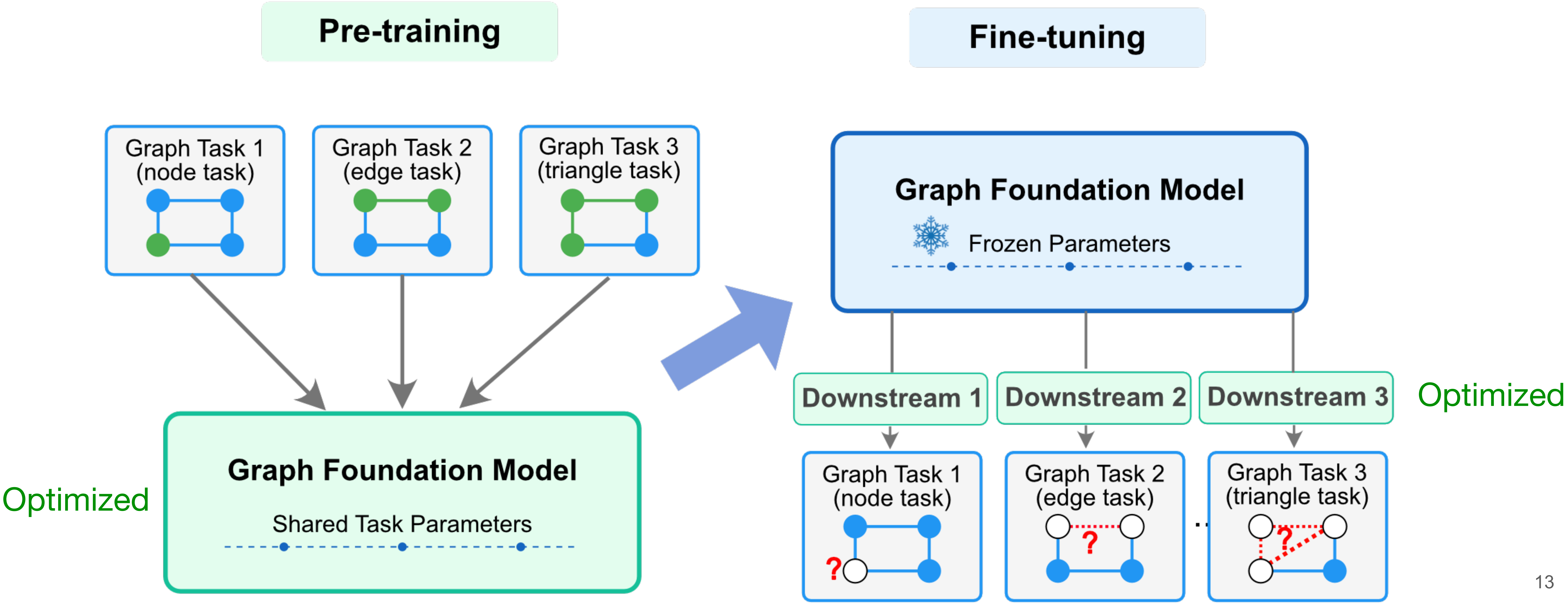
GFMs should train on diverse graph tasks/domains

Desiderata:

1. **Task transferability**
2. **Feature space universality**
3. **Spatio-temporal transferability**
4. **Interoperability with sequence models**

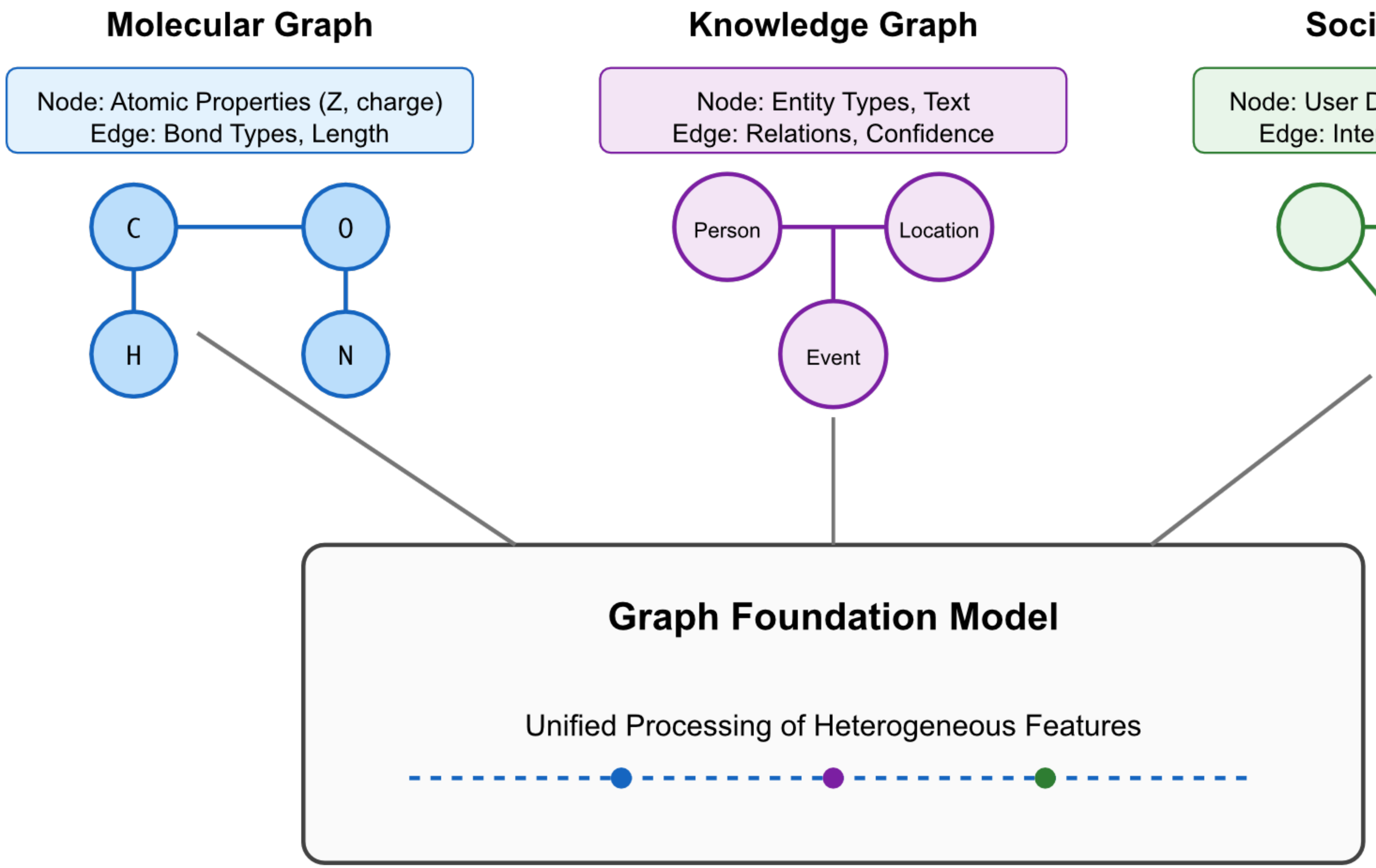
Desiderata for Graph Foundation Models

1. **Task transferability:** Graph foundation models should be capable pretraining on a diverse set of tasks **then** fine-tune on diverse downstream tasks



Desiderata for Graph Foundation Models

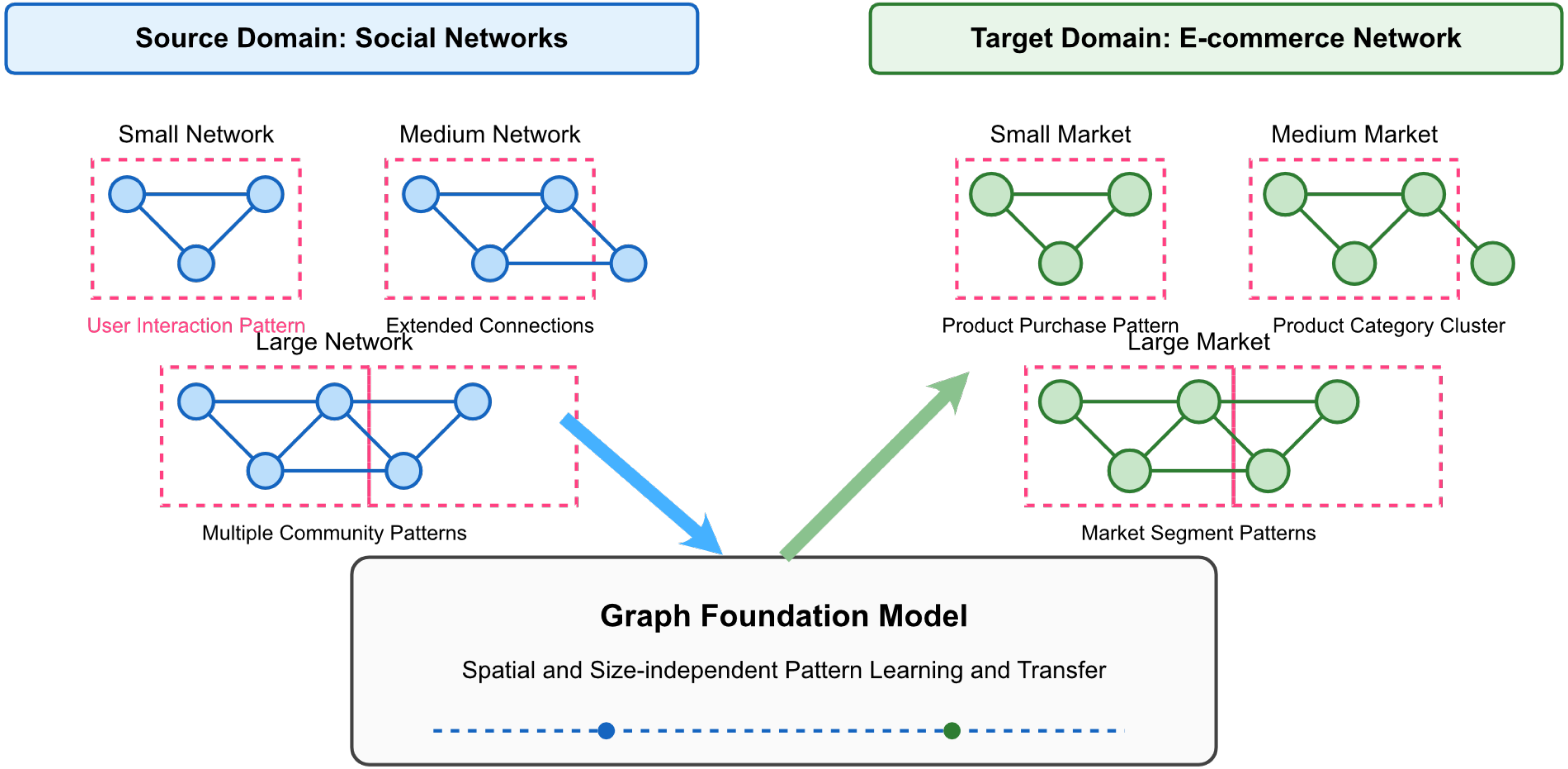
2. **Feature space universality:** Graph foundation models should be able to handle graphs with **heterogeneous node and edge feature spaces** (both categorical, discrete and continuous), allowing for seamless integration of diverse data types and sources



Desiderata for Graph Foundation Models

3. **Spatio-temporal transferability:** Graph foundation models should learn patterns that are transferable across graphs of varying sizes, across time, and across graph locations

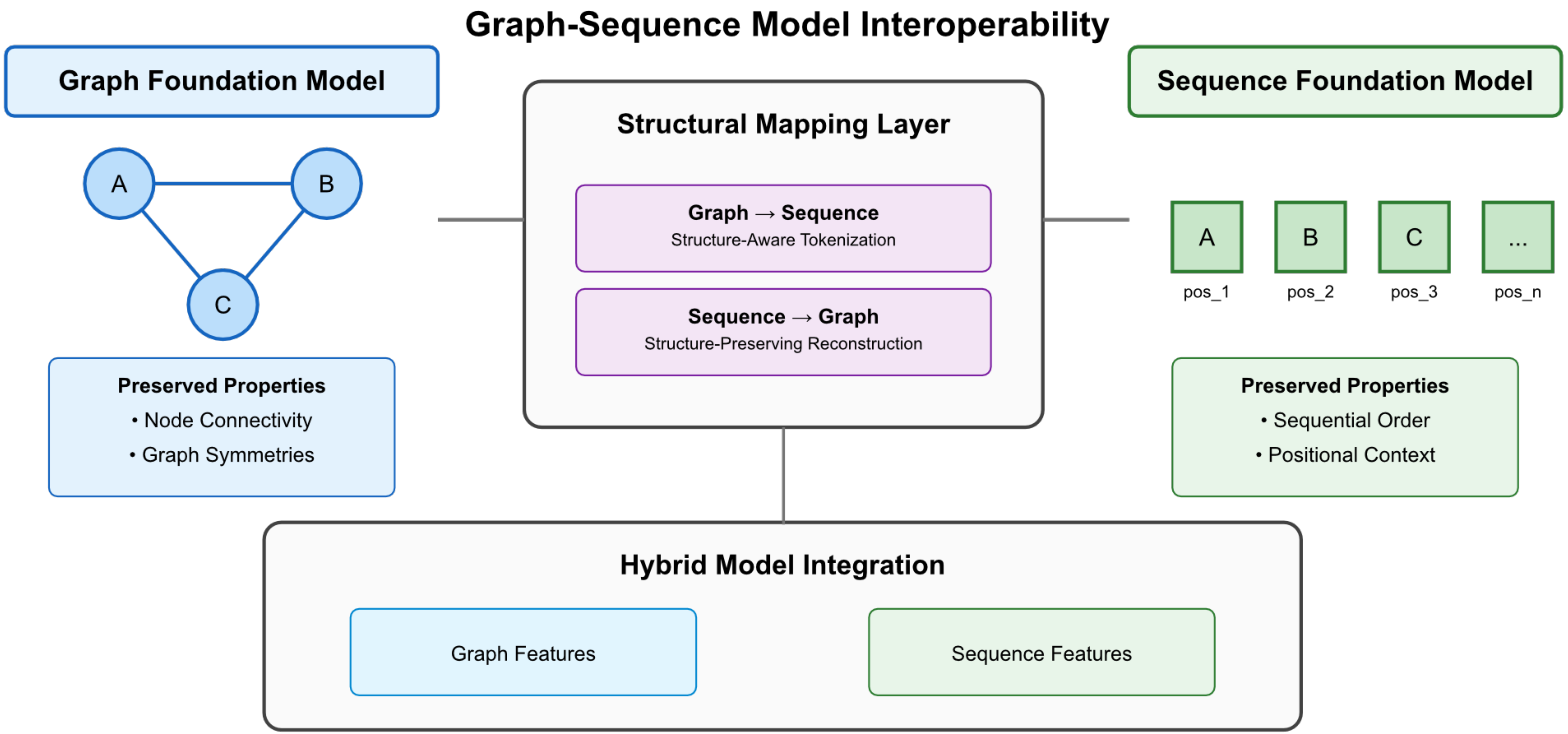
Not on this talk



Desiderata for Graph Foundation Models

4. **Interoperability with sequence models:** Graph foundation models should be able to **communicate effectively with sequence foundation models**

Not on this talk



1. Task Transferability

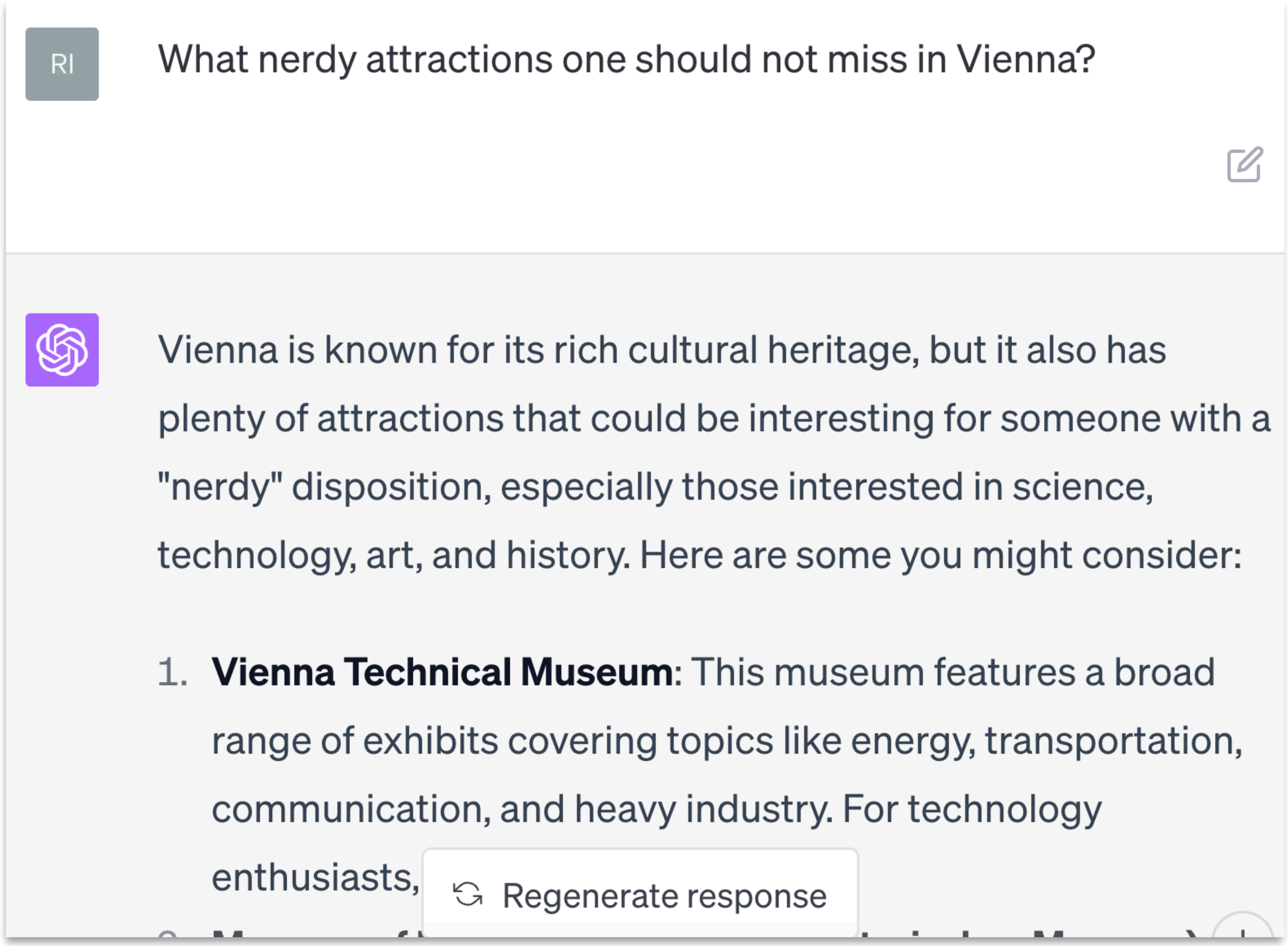
Reconciling Task-Specific Symmetries in Graph Representation Learning

Part 1.1:
The Symmetries of Graph Neural Networks (Recap)

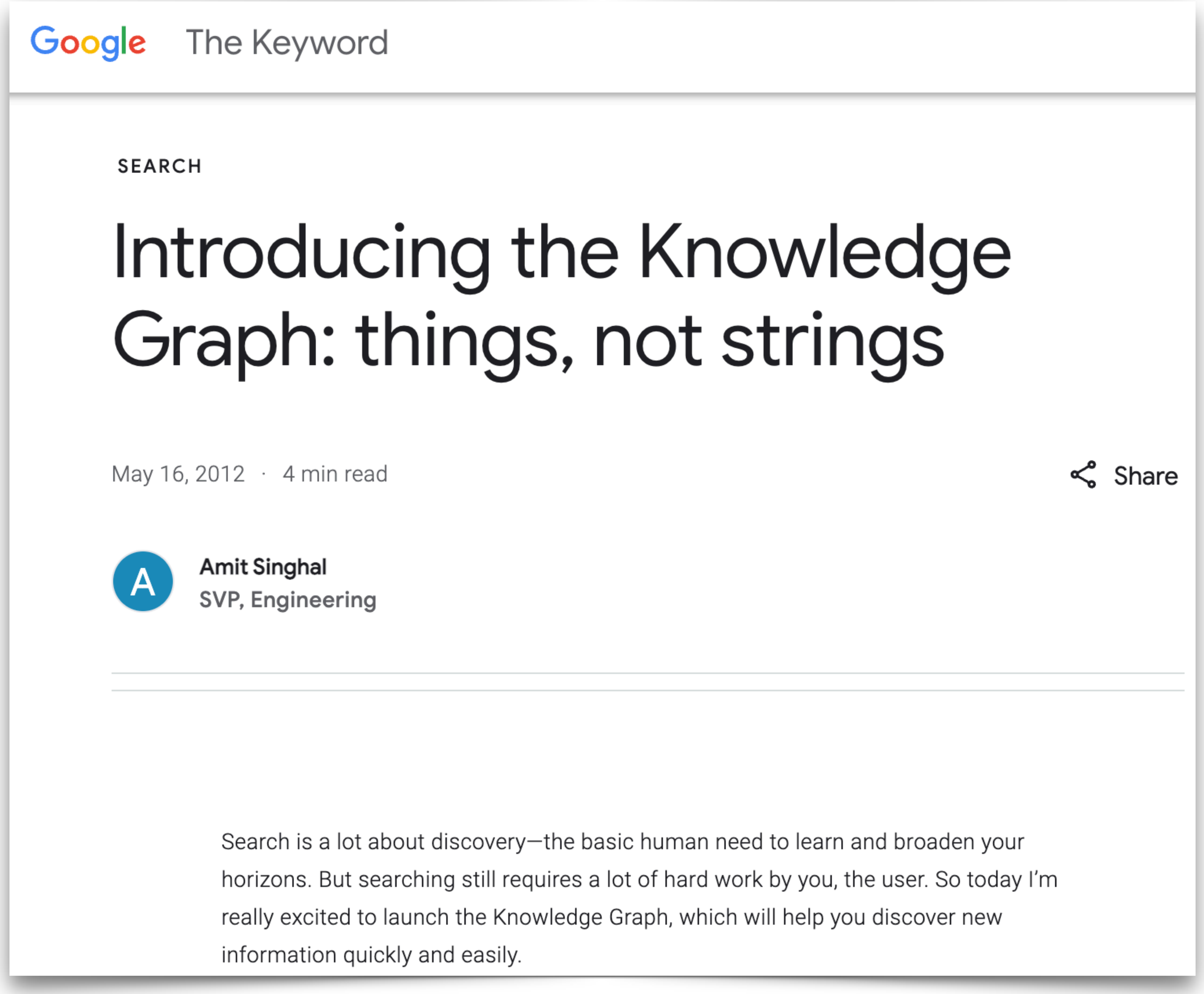
OpenAI vs Google

Knowledge search: **Strings** (text) or **things** (graphs)?

In **2022** OpenAI demoed ChatGPT, **“strings-only”** method.



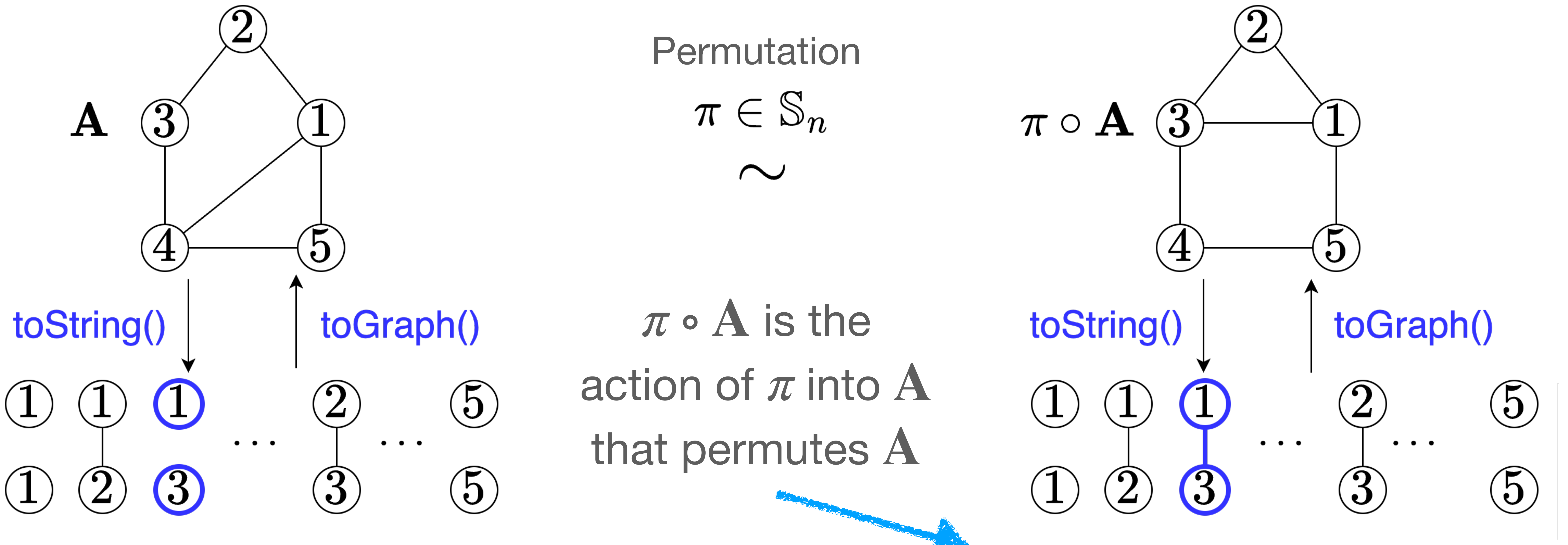
In **2012** Google declared *web search* as **“things, not strings”**.



Graphs are “strings” + symmetries

- **Graphs *are* sequences of edges** with associated (permutation) **symmetries** since **node ids are arbitrary** [Murphy et al., 2019, Xu et al., 2019, Morris et al., 2019].
- In statistics this assumption is called *exchangeability*

Graph sequence isomorphism: Graphs with distinct sequences can be the **same graph.**



$\text{vec}(\mathbf{A}) = (A_{11}, A_{12}, A_{13}, \ldots, A_{23}, \ldots, A_{55}) \sim \text{vec}(\pi \circ \mathbf{A}) = (A_{11}, A_{12}, A_{13}, \ldots, A_{23}, \ldots, A_{55})$

Why are symmetries relevant in relational learning?

ChatGPT used to fail at multi-hop reasoning [Dziri et al., 2023]
(now it fails only on larger graphs).

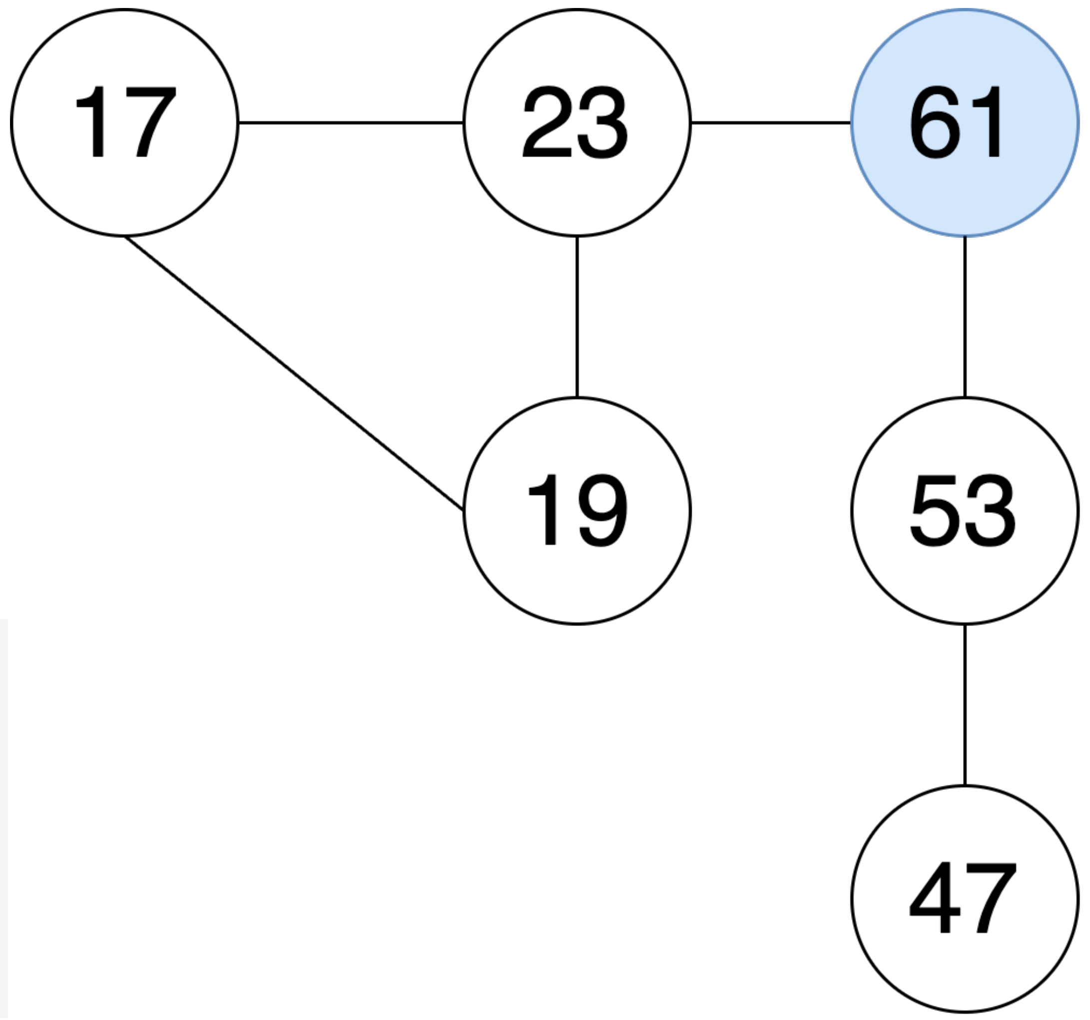
- Order-sensitive models can struggle with tasks that require symmetries

Q: Give number of nodes reachable from 61 in exactly two hops

In the given undirected graph, there are two nodes that are exactly 2-hops away from node 61. These nodes are 17 and 19.

But if we reorder the edges in the prompt the answer changes.

In the given graph, there are three nodes that are exactly 2-hops away from node 61. These nodes are 17, 19, and 47.



- **ChatGPT's answers are sensitive to edge order**
- **Models respecting symmetries must treat all paths identically**

Defining symmetries through groups

A group $\mathcal{G}$ is a set together with a binary operation $\star$ such that:

- Closure holds i.e., $\forall a, b \in \mathcal{G}, \; a \star b \in \mathcal{G}$
- Associativity holds $(a \star b) \star c = a \star (b \star c) \;\; \forall a, b, c \in \mathcal{G}$
- Identity element exists i.e., $\exists e \in \mathcal{G}$ s.t. $a \star e = e \star a = a \;\; \forall a \in \mathcal{G}$
- Inverse exists for every element and $a \star a^{-1} = a^{-1} \star a = e \;\; \forall a \in \mathcal{G}$

Credit: Bala Srinivasan

(Left) Group actions

For a group $\mathcal{G}$, binary operation $\star$, and with identity e, and a set X, a (left) group action is a function $\circ : \mathcal{G} \times X \to X$, such that

- $e \circ x = x, \ \ \forall x \in X$
- $\pi \circ (h \circ x) = (\pi \star h) \circ x, \ \ \forall \pi, h \in \mathcal{G}, \ \ \forall x \in X$

A function f is $\mathcal{G}$-**invariant** if $f(x) = f(\pi \circ x), \forall \pi \in \mathcal{G}, \ \ \forall x \in X$

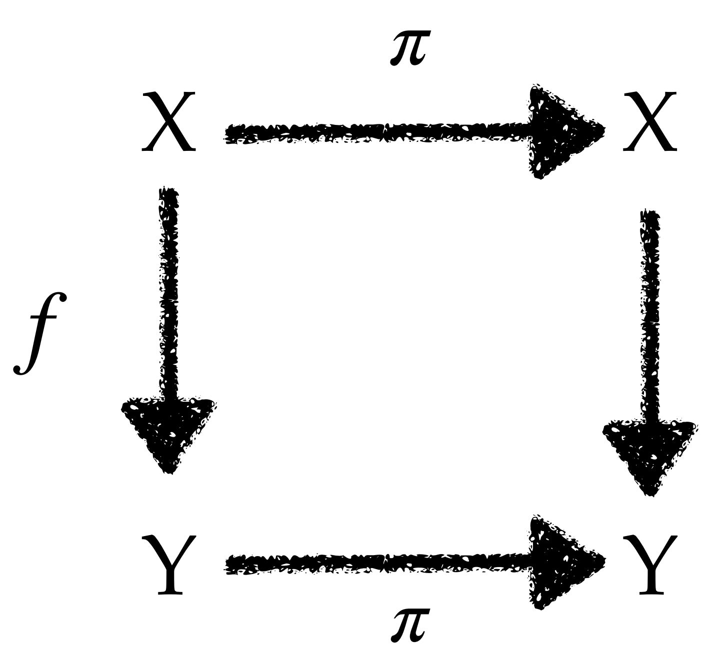
A function f is $\mathcal{G}$-**equivariant** if $\pi \circ f(x) = f(\pi \circ x), \forall \pi \in \mathcal{G}, \ \ \forall x \in X$

Credit: Bala Srinivasan

Group Equivariant and Invariant Neural Networks

Let $\mathrm{X} \subseteq \mathbb{R}^d$ and $\mathrm{Y} \subseteq \mathbb{R}^k$ be two vector spaces
Let $f : \mathrm{X} \rightarrow \mathrm{Y}$ be a neural network function

$\mathscr{G}$-equivariant function



$$f(\pi \circ x) = \pi \circ f(x)$$

$\forall \pi \in \mathscr{G}$

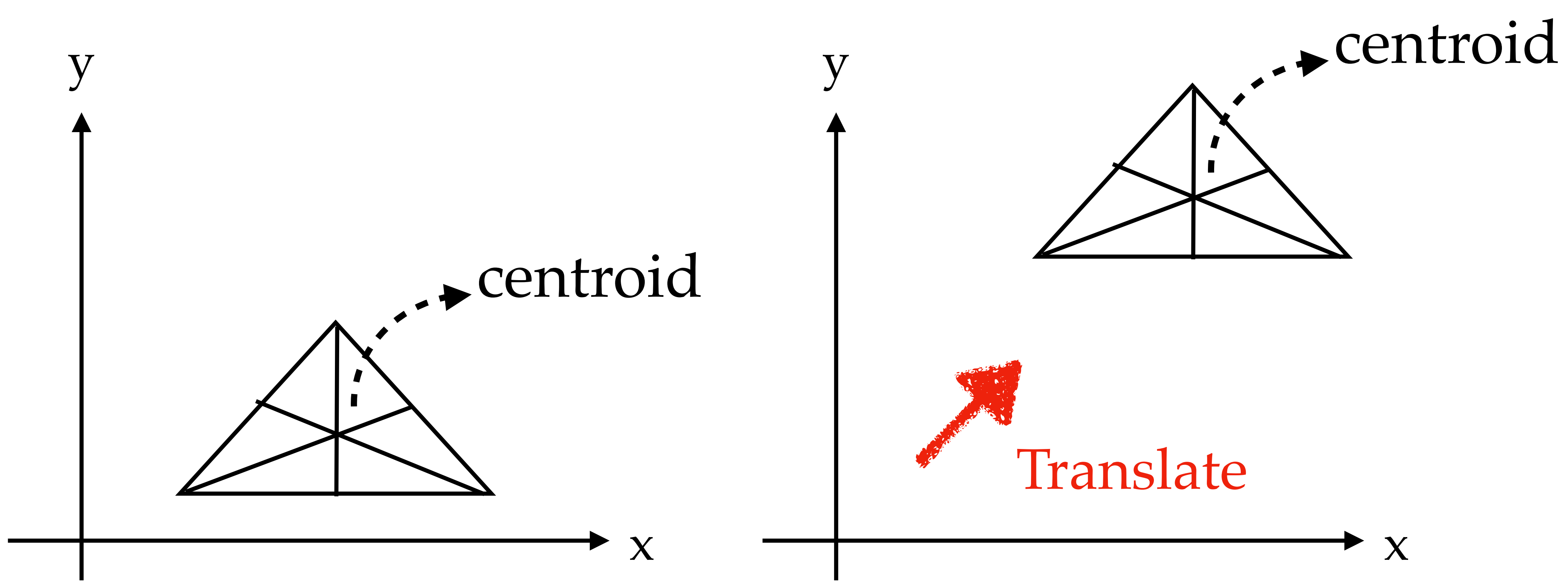
$\mathscr{G}$-invariant function

$$f(\pi \circ x) = f(x)$$

$\forall \pi \in \mathscr{G}$

Group representations of appropriate dimensions

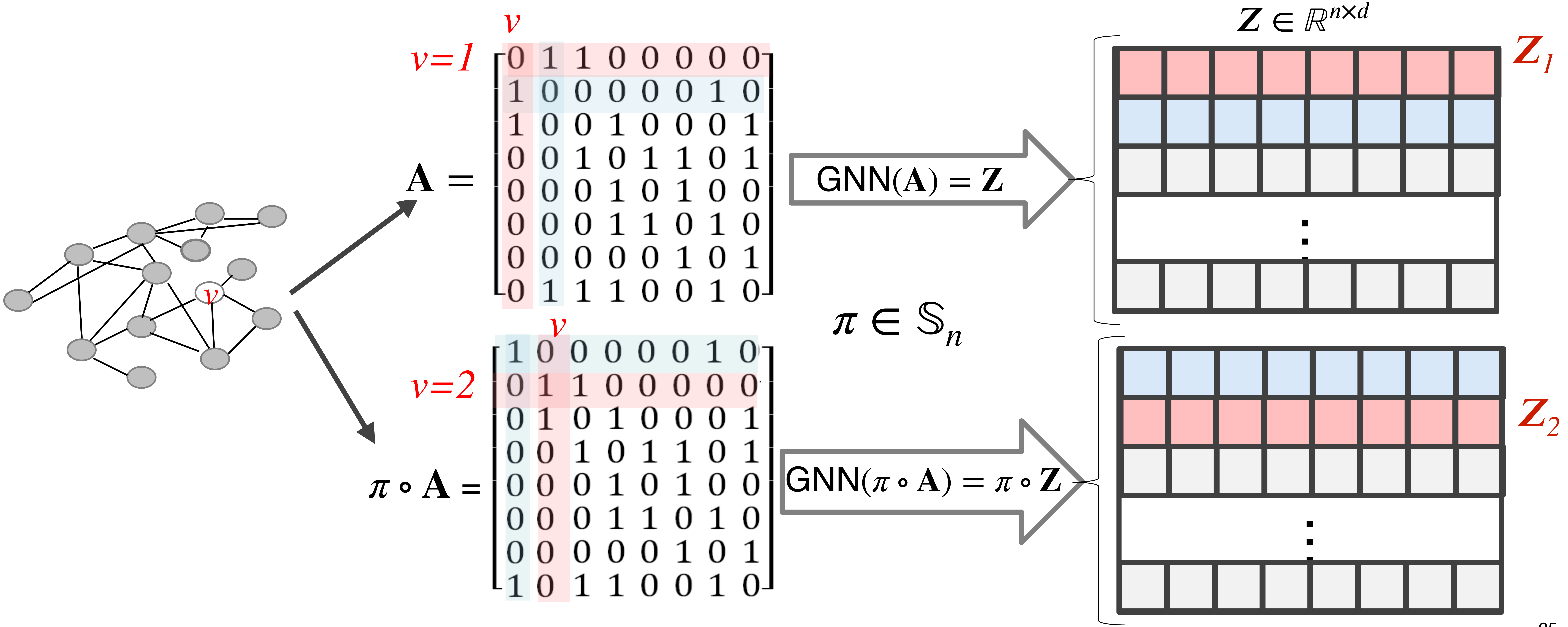
Symmetries of a triangle (2D):
- **Area** of the triangle is **invariant** to translations
- **Centroid** of the triangle is **equivariant** to translations



Credit: Bala Srinivasan

Symmetries in relational learning

- **Permutation equivariance**: A **Graph Neural Network**, GNN($\mathbf{A}$), is a neural network that learns node embeddings from adjacency matrix $\mathbf{A} \in R^{n \times n \times (p+1)}$.
  GNN node embeddings are ***equivariant*** to $\pi \circ \mathbf{A}$, where $\pi \in \mathbb{S}_n$ and $\mathbb{S}_n$ is the permutation group



G-equivariances in Graph Neural Networks (GNNs):

- Kondor, R., & Trivedi, S., On the generalization of equivariance and convolution in neural networks to the action of compact groups. ICML 2018.

- Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019, July). Weisfeiler and leman go neural: Higher-order graph neural networks. AAAI 2019.

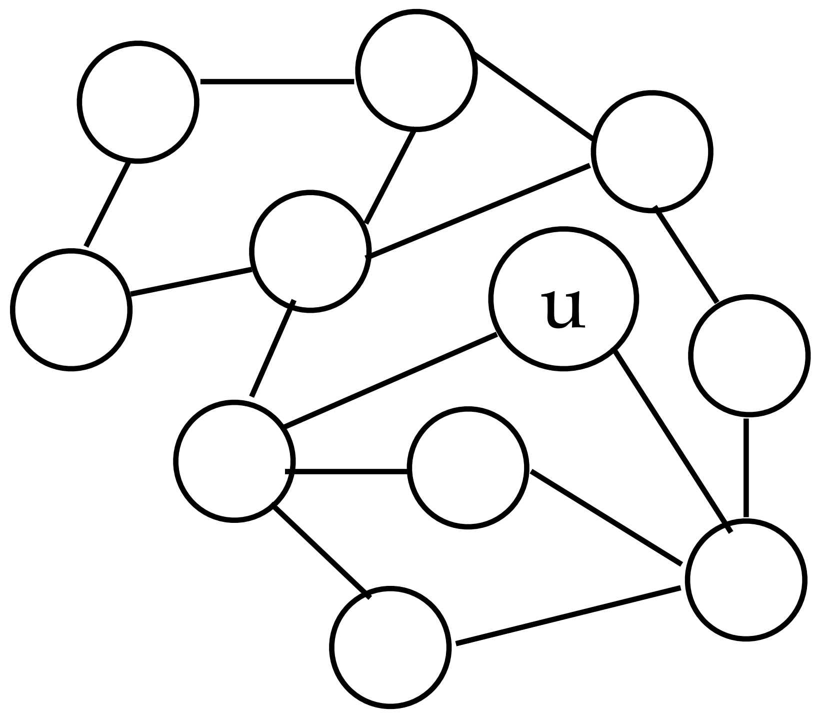
- Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. ICLR 2019.

Figure from: On the Equivalence between Positional Node Embeddings and Structural Graph Representations (Srinivasan & R., ICLR 2020)

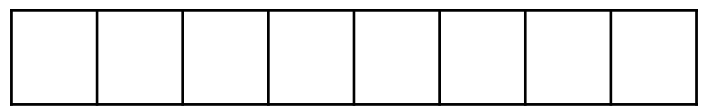
Downstream Task — Node Classification

f — (Abstractly) A function that outputs node representations $f : \mathbb{R}^{n \times n \times p} \times \mathbb{N} \to \mathbb{R}^{n \times d}, \;\; d > 0$

G-invariant embedding $f(\mathbf{A}, u) = f(\pi \circ \mathbf{A}, \pi \circ u) \in \mathbb{R}^d \;\; , \pi \in \mathbb{S}_n$



$f(\mathbf{A}, u) \in \mathbb{R}^d$

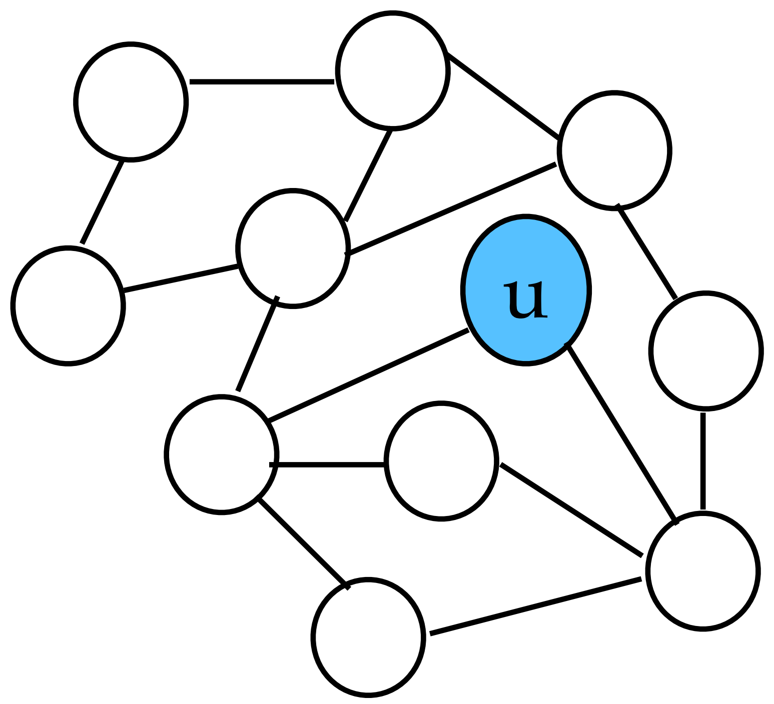


node u's embedding

Classifier

$g_\theta : \mathbb{R}^d \to \{1, \ldots, n_{\text{classes}}\}$

$\mathbf{A} \in \mathbb{R}^{n \times n \times (1+k+p)}$ simplified tensor notation of the graph



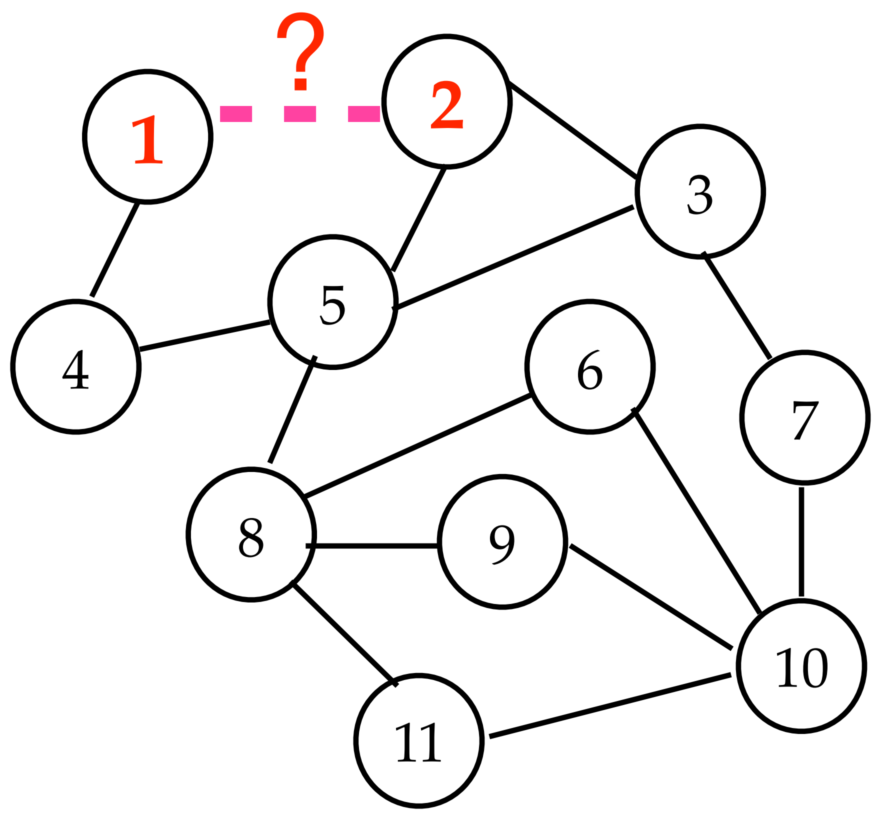
Node Classification (Downstream Task)

Example:
Given a social network $\mathbf{A}$, predict the types of ads to serve user **u**

Downstream Task — Link (edge) Prediction

f — (Abstractly) A function that outputs node representations $f : \mathbb{R}^{n \times n \times p} \times \mathbb{N}^2 \to \mathbb{R}^{n \times d}, \; d > 0$

G-equivariant embedding $\pi \circ f(\mathbf{A}) = f(\pi \circ \mathbf{A})_{\pi \circ u} \in \mathbb{R}^{n \times d} \;, \pi \in \mathbb{S}_n$



$f(\mathbf{A},1) \in \mathbb{R}^d$ Node 1's embedding

$f(\mathbf{A},2) \in \mathbb{R}^d$ Node 2's embedding

Classifier

$g_\theta : \mathbb{R}^{2 \times d} \to \{1, \ldots, n_{\text{classes}}\}$

$y_{1,2}$

Link (edge) Classification (Downstream Task)

$\mathbf{A} \in \mathbb{R}^{n \times n \times p}$ simplified tensor notation of the graph

Example:
Predict a relation between Taylor Swift and Terence Tao

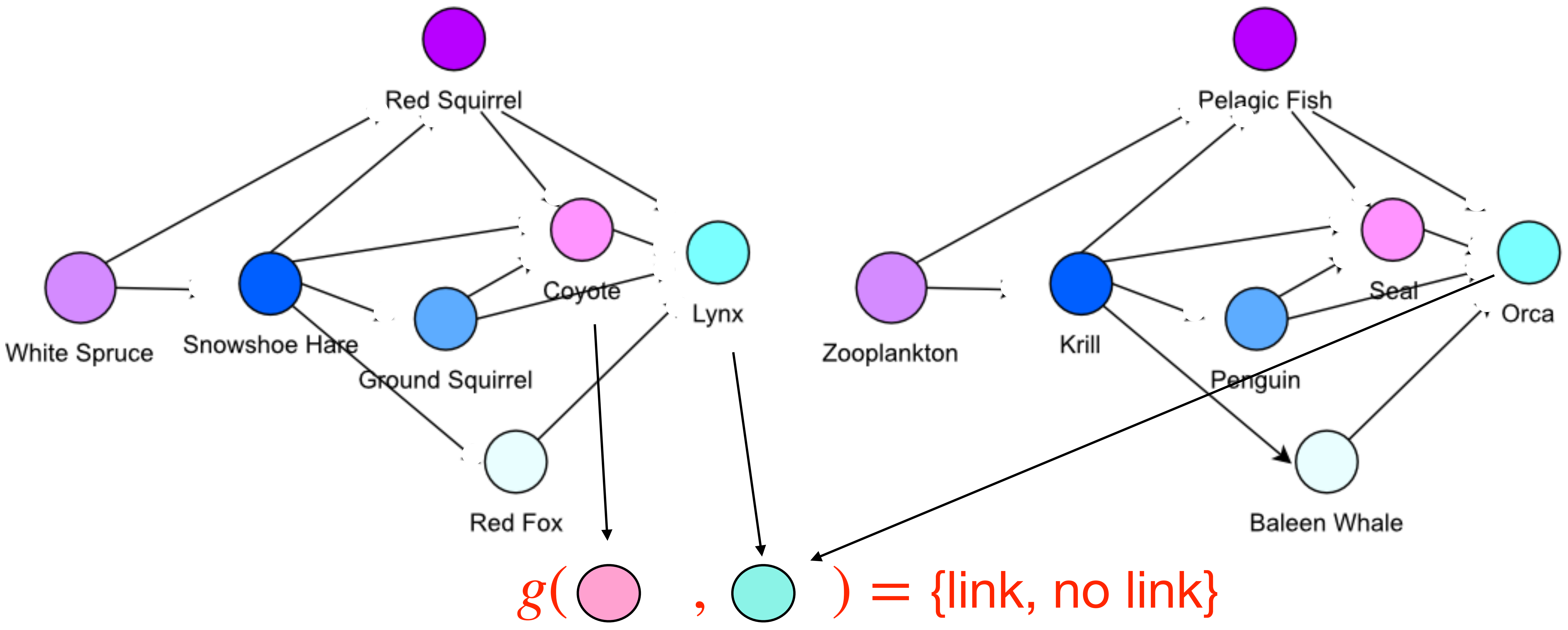
There is something wrong with this approach…

Part 1.2:
The Symmetries of Graph Tasks

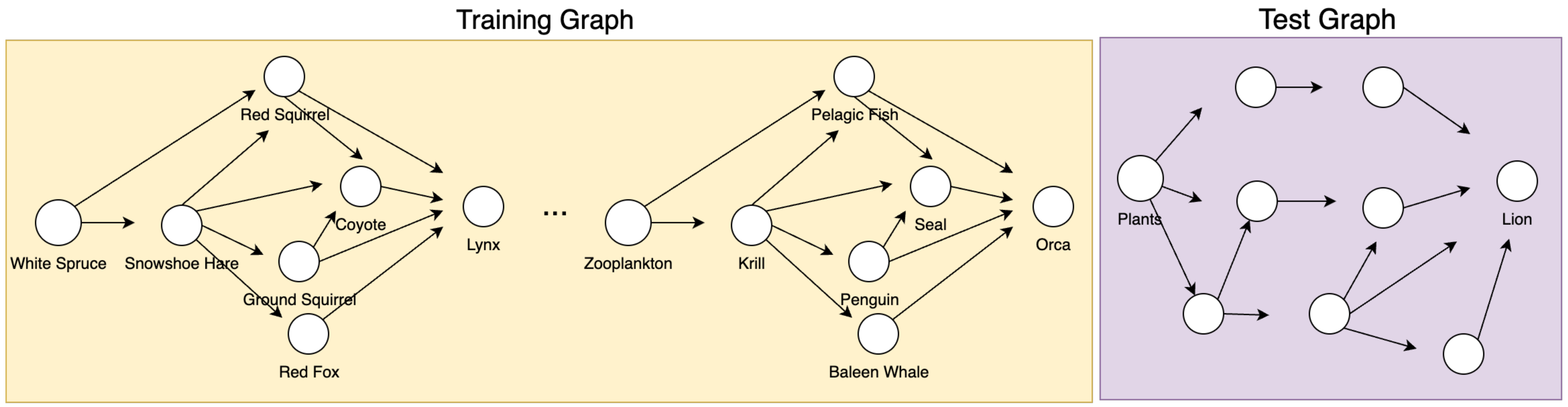
Equivariant node representations **generalize poorly in** link prediction

- *Edge-based tasks have symmetries that are incompatible with the symmetries required for node tasks*

- If $g : \mathbb{R}^d \times \mathbb{R}^d \to \{\text{link, no link}\}$, then any g that learns to predict edge (Lynx, Coyote) must also predict edge (Orca, Coyote)



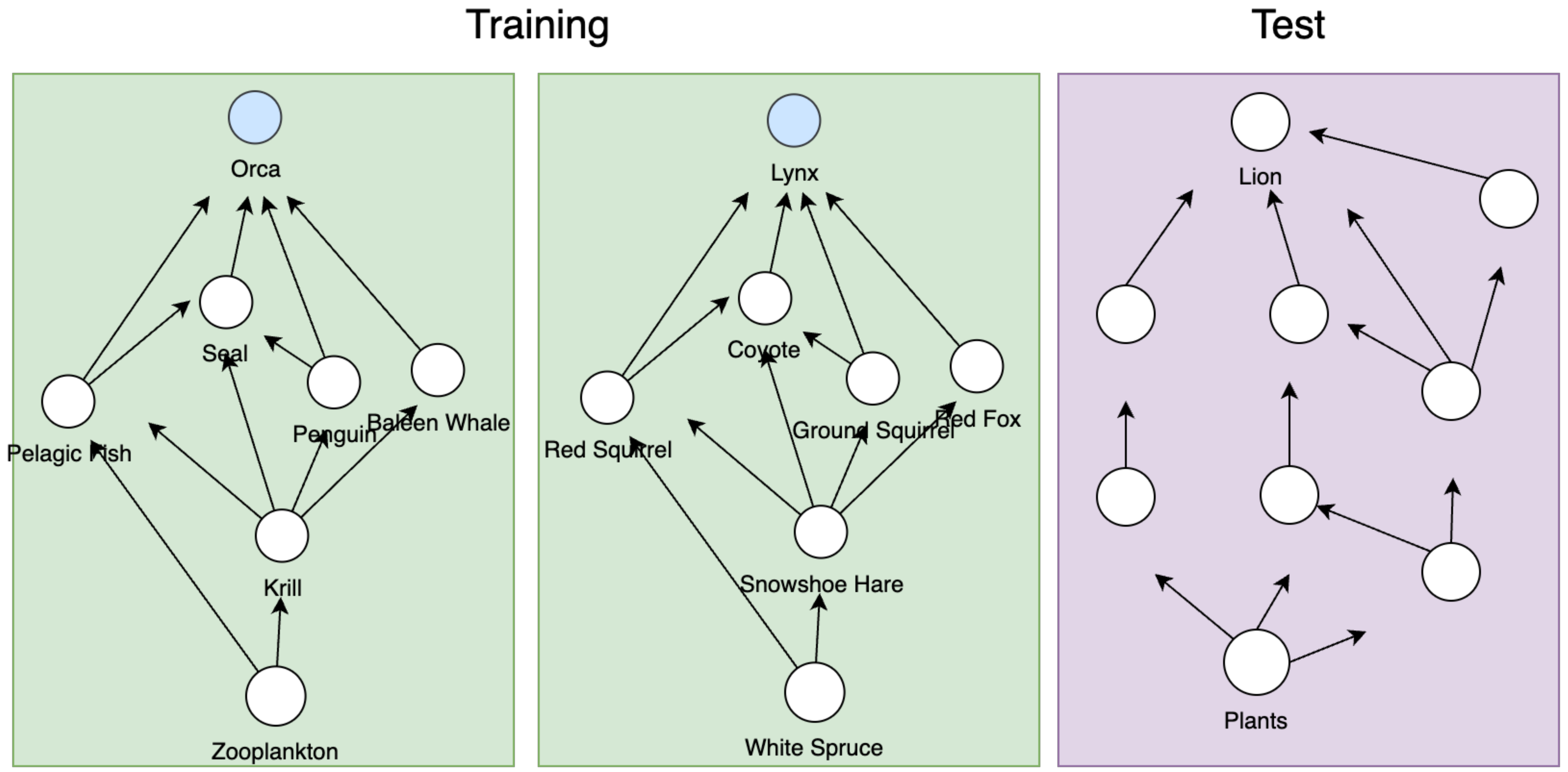
colors according to GNN embeddings



Graph Task Symmetries

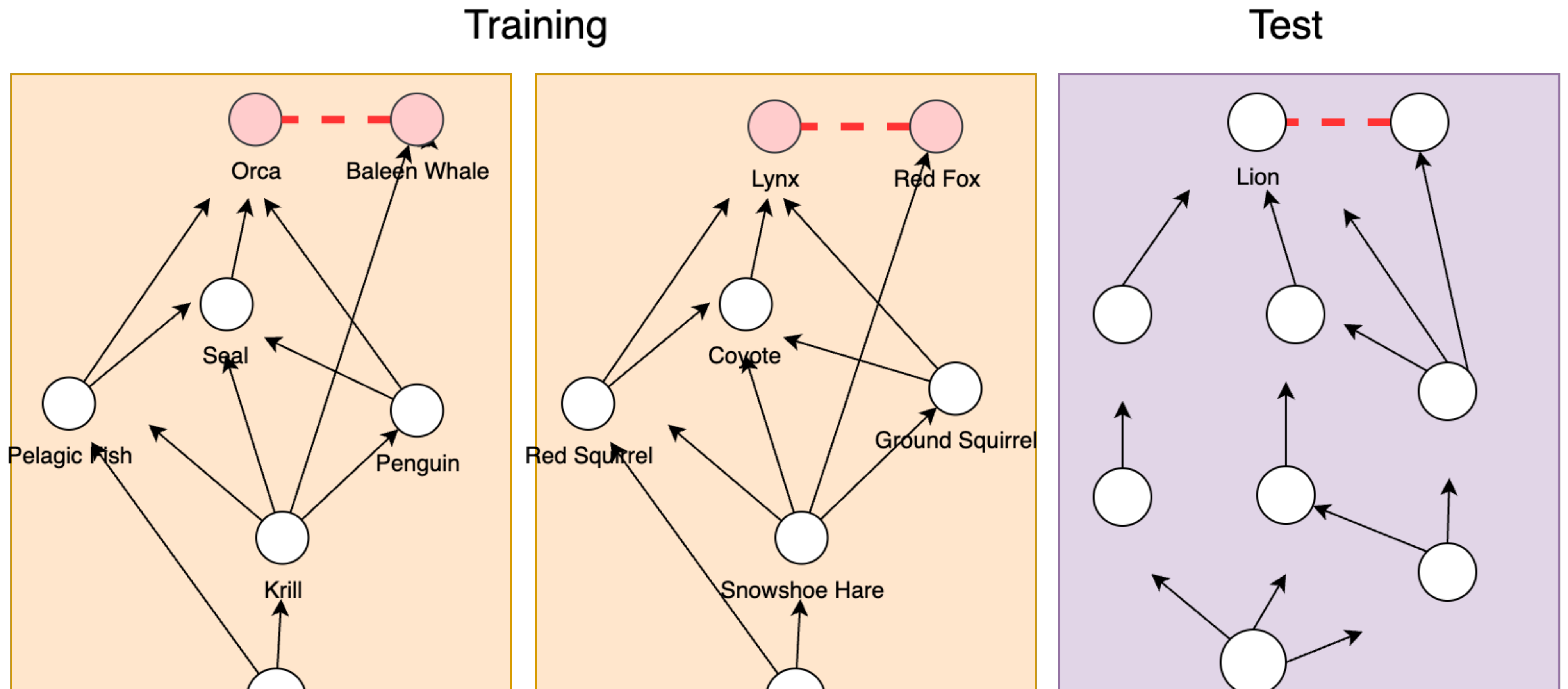
- Node classification
- Link prediction
- Triangle counting

Node Task



- These all require **different** neural network symmetries

Edge Task



On the Equivalence between Positional Node Embeddings and Structural Graph Representations (Srinivasan & R., ICLR 2020)

All k-node Tasks Require k-node Equivariances

- **Theorem (Task Equivariances) [summary].** The neural networks that can learn tasks over $k \in \{1,\ldots,n\}$ nodes should have distinct equivariances depending on k.

- **Theorem (Task Equivariances).** Let $G = (V, E, X)$ be an attributed graph, with $V = \{1,\ldots,n\}$ as nodes, E as edges, X as node and edge attributes. Let $S_k \subseteq V$ be a set of k nodes (w.l.o.g. $S_k = \{1,\ldots,k\}$). Consider a random variable Y_{S_k} encompassing the nodes in S_k that we wish to learn with a neural network f via supervised learning: $P(Y_{S_k} \mid S_k, G) = f(S_k, G)$. Then, f must be described by two permutation groups: the normal subgroup $\mathbb{S}_k$ that defines the equivariances of f related to the nodes S_k in the task and the normal subgroup $\mathbb{S}_n \backslash \mathbb{S}_k$ which describes the invariance of f to all remaining nodes in the graph.

R., "A Mathematical Framework for Graph Foundation Models", in preparation

“All k-node Tasks Require k-node Equivariances”

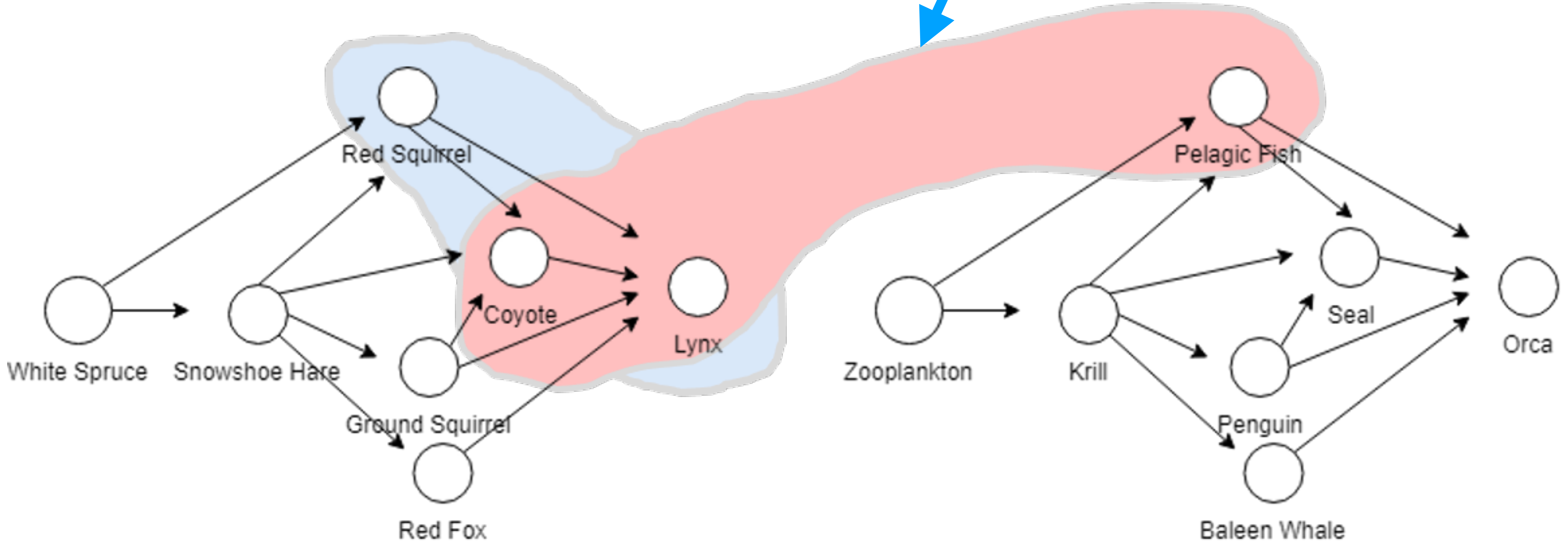
- **Example with k=3**

$\mathbb{S}_3 \subset \mathbb{S}_n$

G-equivariance on target nodes

$\mathbb{S}_n \backslash \mathbb{S}_3$

G-invariance
on remaining nodes

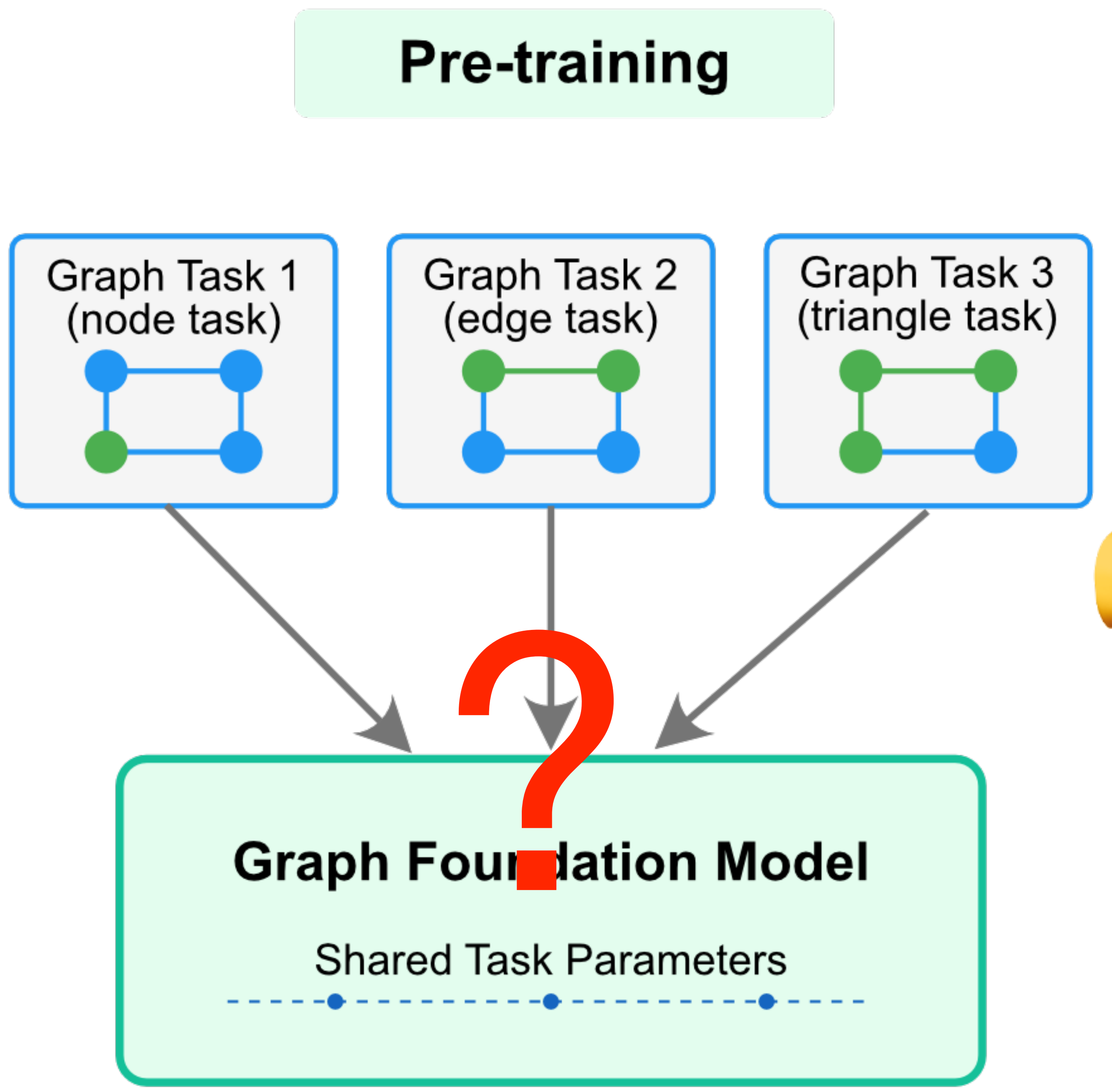


Example of 3-node task

On the Equivalence between Positional Node Embeddings and Structural Graph Representations (Srinivasan & R., ICLR 2020)

Impossibility Result

Unknown authors, "Holographic Node Representations: Pre-Training Task-Agnostic Node Embeddings" ICLR 2025 submission
https://openreview.net/forum?id=tGYFikNONB



Proposition 2.3 (Informal). *For any node embedding model f, there exists two tasks of different orders for which at least one is not solvable using f.*

Then, there are **no Graph Foundation Models** that can pretrain informative **node embedding vectors** $z_v \in \mathbb{R}^d, v \in V$ over multiple graph tasks

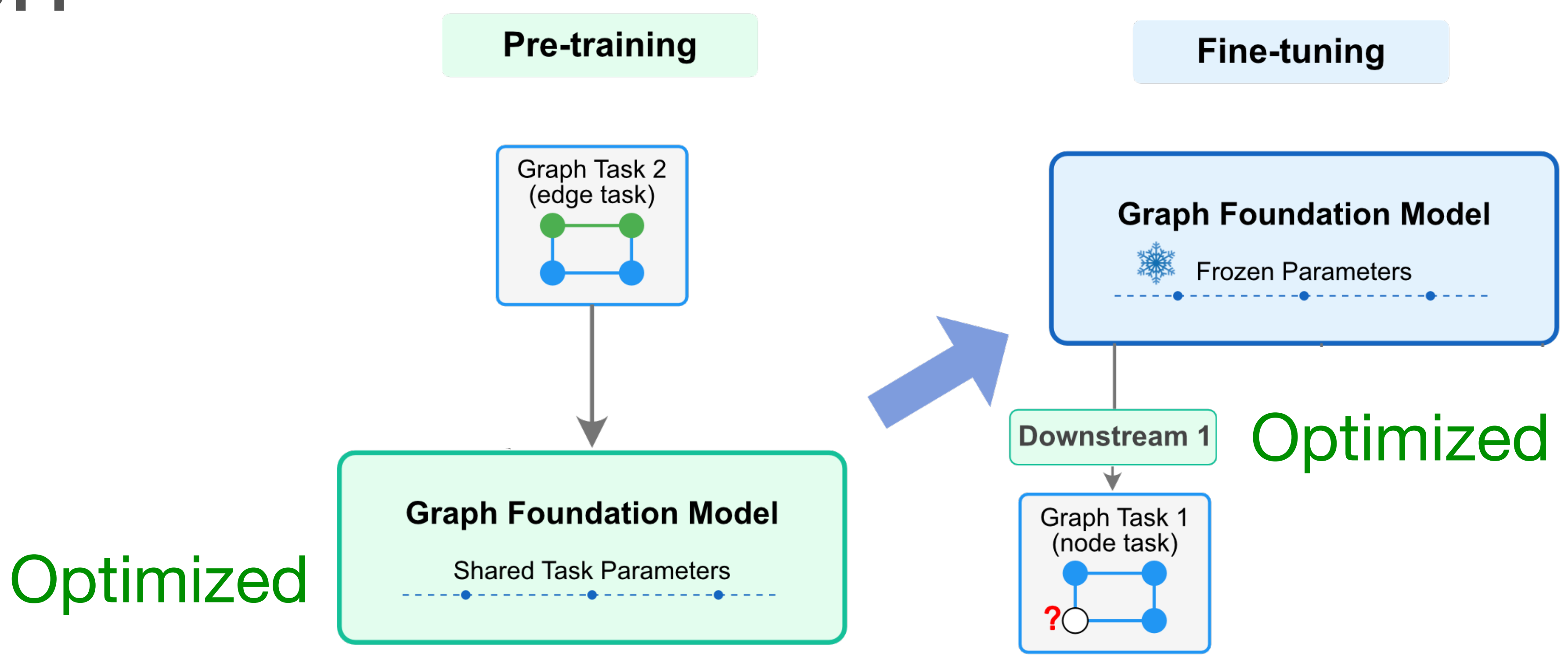
Proposition B.1 (Impossibility of accurate any-order task learning from node embeddings). *Consider simultaneously performing two tasks, $\mathcal{T}_{node}$ and $\mathcal{T}_{link}$, using node embeddings $f(\boldsymbol{A}, \boldsymbol{X}) \in \mathbb{R}^{n \times d}$. There exist $\mathcal{T}_{node}$ and $\mathcal{T}_{link}$ such that, for any MLP$_{node}$ and MLP$_{link}$ achieving the training minima, no f that produces either positional or structural representations can simultaneously satisfy the following two conditions: (1) $\mathcal{L}_{\mathcal{T}_{node}}(\mathcal{D}_{node}) = \mathcal{L}^*_{\mathcal{T}_{node}}$; (2) $\mathcal{L}_{\mathcal{T}_{link}}(\mathcal{D}_{link}) = \mathcal{L}^*_{\mathcal{T}_{link}}$. That is, when using standard (flat) node embeddings, the predictions cannot be simultaneously accurate (in test) for both tasks.*

What happens if pertaining with the wrong k-node equivariance?

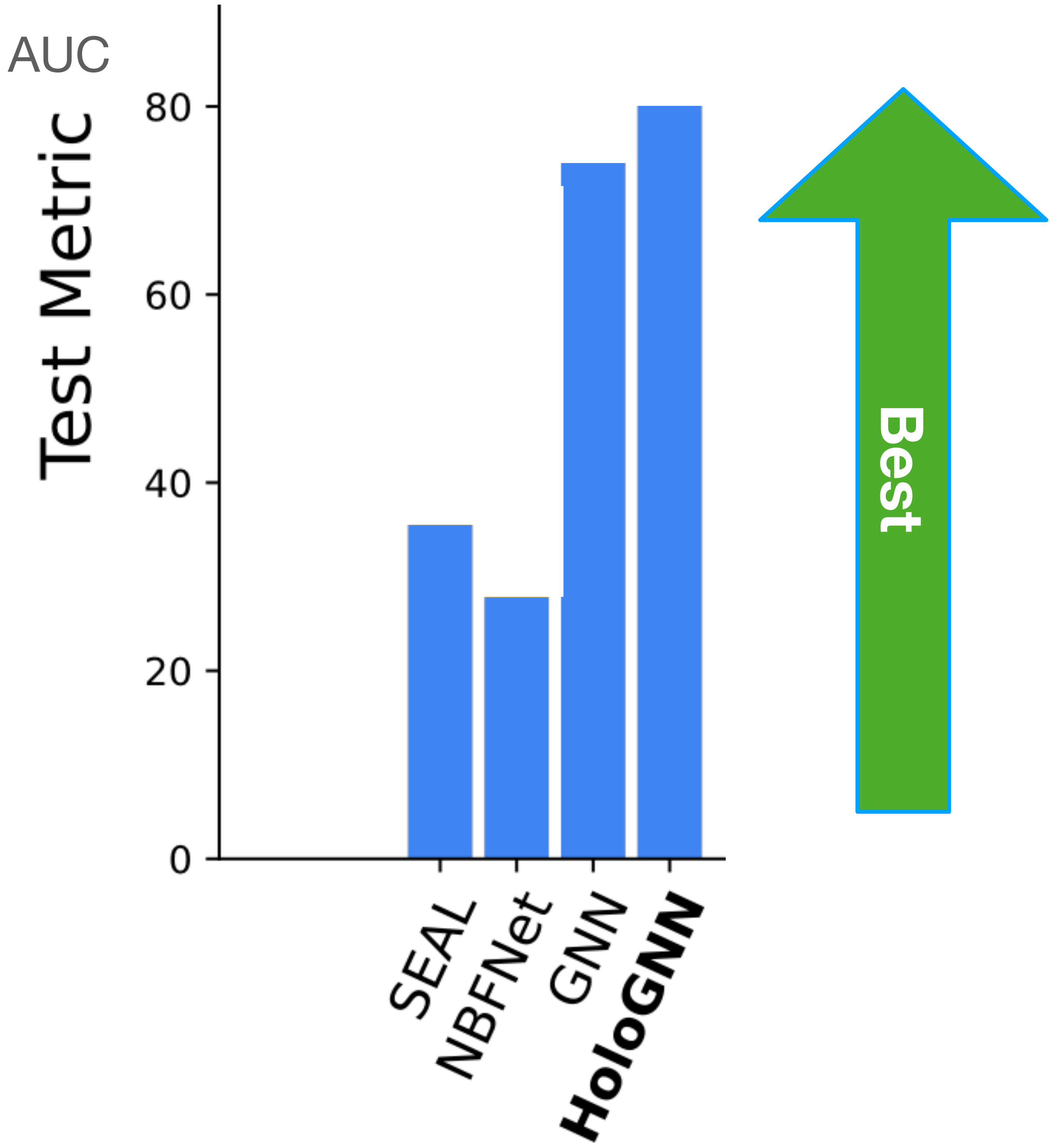
Pretrain: link prediction task (k=2 task)

Task: transfer learning for node classification (k=1 task)

- **SEAL**: has equivariance for k=2 on a node (k=1) task
- **NBFNet**: has equivariance for k=2 on a node (k=1) task
- **GNN**: has equivariance for k=1 on a node (k=1) task
- **HoloGNN**: New equivariant-universal embedding approach



CORA dataset
(Error bars too small to show)

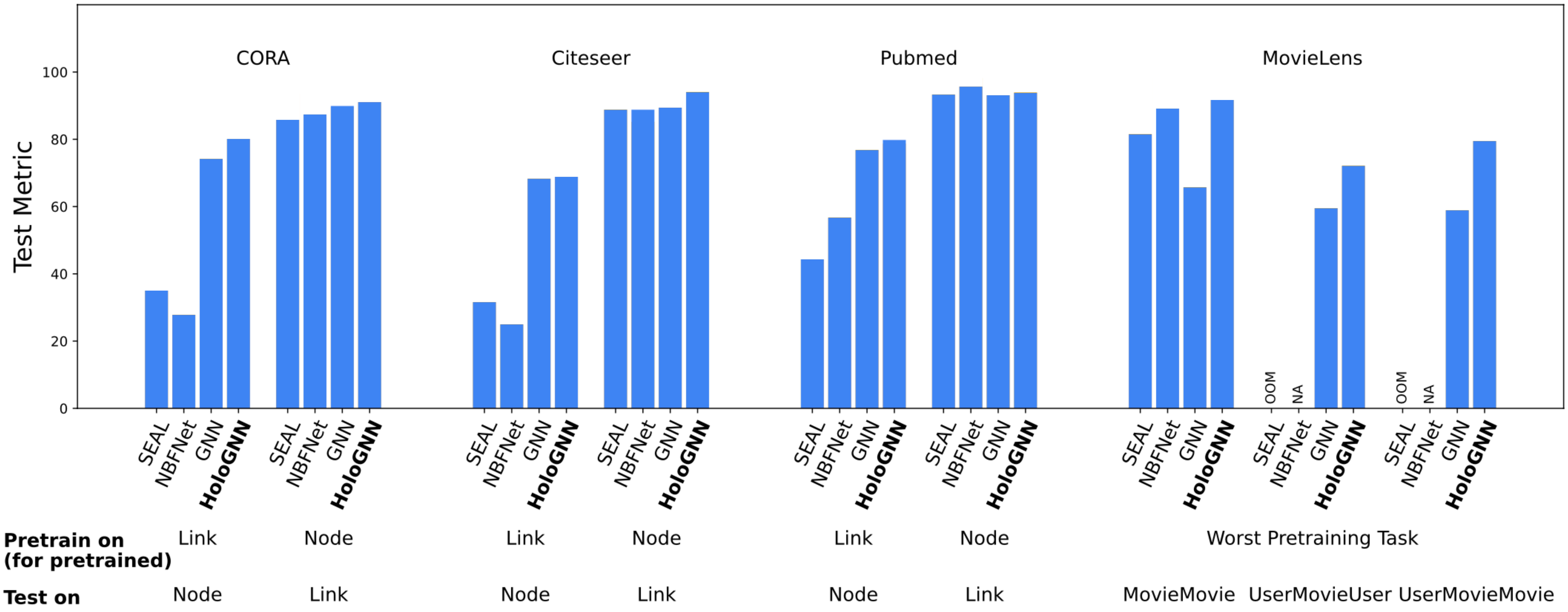


Pretrain on (for pretrained) Link

Test on Node

Figure: Unknown authors, "Holographic Node Representations: Pre-Training Task-Agnostic Node Embeddings" ICLR 2025 submission

Pretrain on a graph task, Transfer learn on another task



Potential Solution: Holographic Node Embeddings

Unknown authors, ”Holographic Node Representations: Pre-Training Task-Agnostic Node Embeddings” ICLR 2025 submission

- Holographic representations is the **first step** to solve **task** pertaining issue 👍
- **Variable-dimensional** node embeddings $z_v \in \mathbb{R}^{m \times d}, v \in V$, where $m \leq n$ would depend on the collection of training graphs.

Definition 3.1 (Holographic Node Representations). *Holographic node representations* consist of two learnable, parameterized maps:

(1) **Expansion Map:**

Graph neural network equivalent of eigenvalue multiplicities, needed for reduction map

$$E_\theta : \{0,1\}^{n \times n} \times \mathbb{R}^{n \times d_1} \rightarrow \mathbb{R}^{n \times T \times d_e} \times (2^{[n]})^L \times \mathbb{N}^T \quad (2)$$

The expansion map, parameterized by θ, takes as input the adjacency matrix and the initial structural representations, and it outputs: (a) A $(T \times d_e)$-dimensional representation for each of the n nodes (2D representation), denoted by $V_\theta(\boldsymbol{A}, \boldsymbol{V}_1)$; (b) A sequence of L lists of node IDs, where nodes within each list share the same role, and nodes in different lists have distinct roles; (c) A sequence of integers, indicating how the T node representations should be grouped.

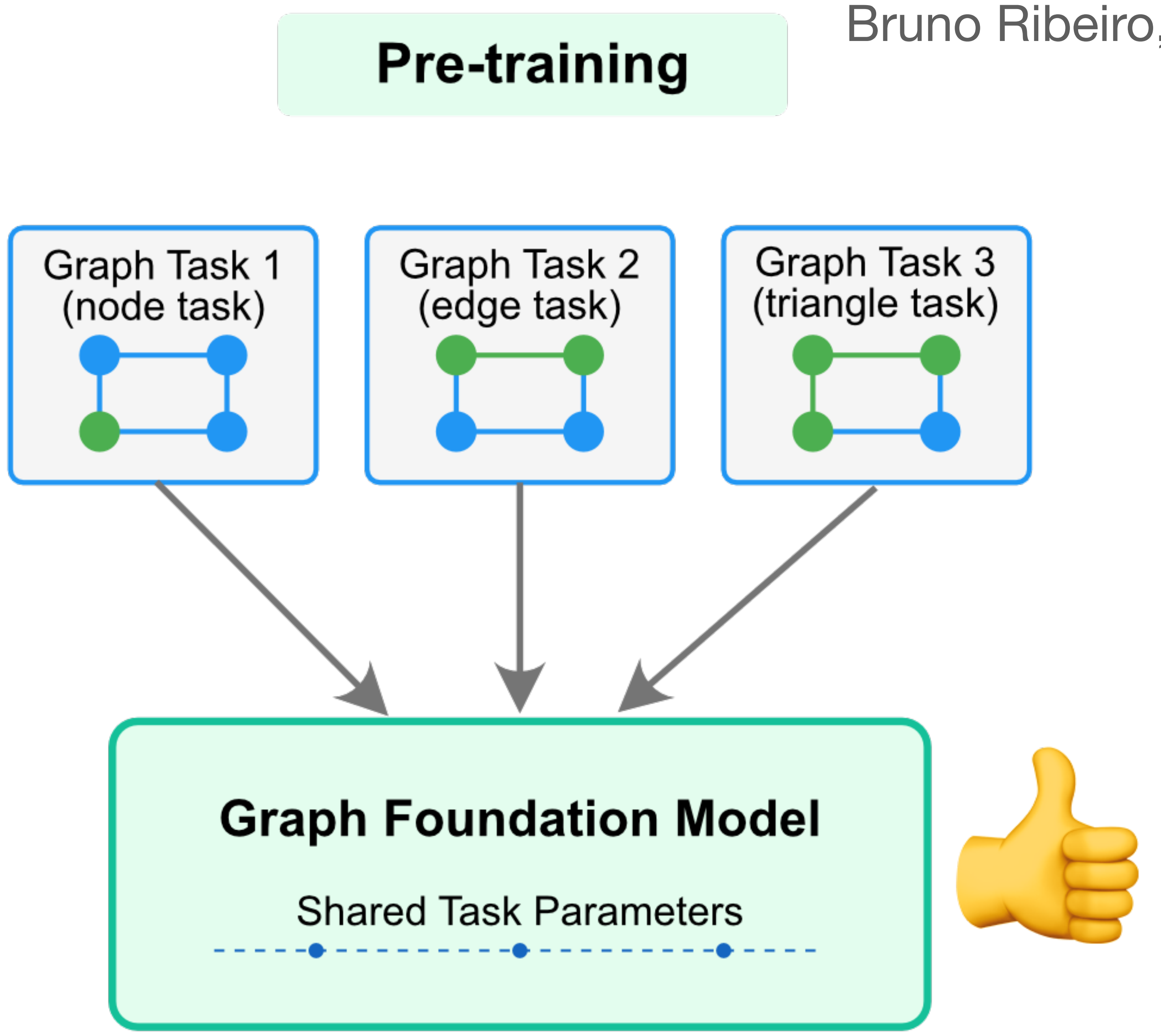
(2) **Reduction Map:**

$$R_\psi : \mathbb{R}^{n \times T \times d_e} \times (2^{[n]})^L \times \mathbb{N}^T \rightarrow \mathbb{R}^{\binom{n}{r} \times d_r} \quad (3)$$

The reduction map, parameterized by ψ, takes the output of the expansion map and produces 1D representations for any set of r nodes.

Bruno Ribeiro, Purdue

Holographic Node Embeddings



- **Expansion map** tries to be permutation sensitive (**not** G-equivariant)
- **Reduction map** restores the appropriate k-node equivariances broken by expansion map

Details:

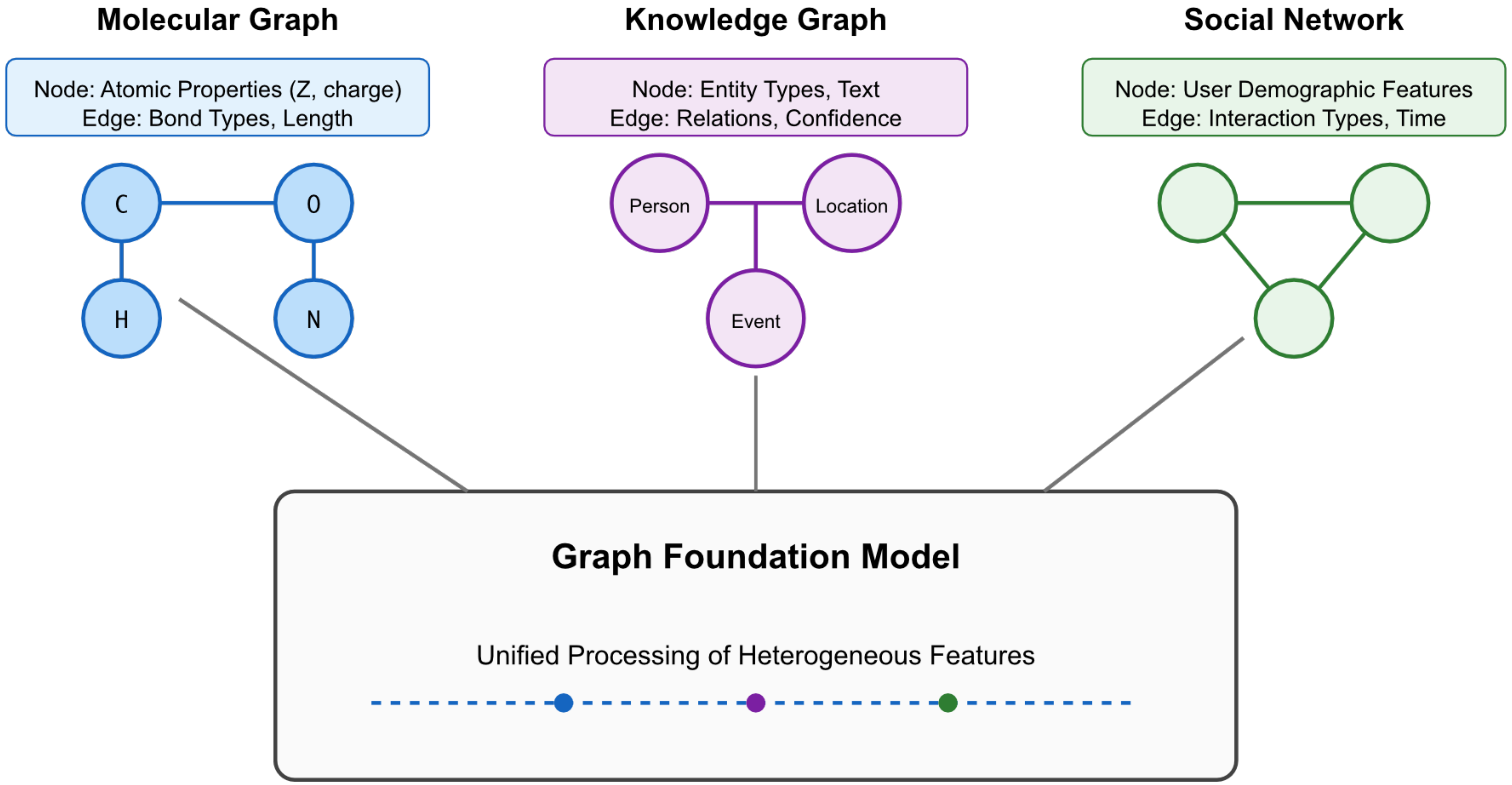
- Property (1): The composition of expansion and reduction ($R_\psi \circ E_\theta$) produces structural representations (one for each set of k nodes), i.e., $\pi \circ R_\psi(E_\theta(\boldsymbol{A}, \boldsymbol{V}_1)) = R_\psi(E_\theta(\pi \circ \boldsymbol{A}, \pi \circ \boldsymbol{V}_1))$ for any $\pi \in \mathbb{S}_n$.
- Property (2): For any undirected graph $G = (V, E, \boldsymbol{X})$ and isomorphic nodes $u, v \in V$, with $u \neq v$ and having different neighborhoods, there exists a θ such that, the expansion maps are different.

2. Feature Space Universality

Maybe an Impossible Dream

Pretrain a single foundation graph model over multiple graphs with distinct feature spaces

- Node/edge features can be a **mix** of
  - $\mathbb{R}$, real-valued features (totally ordered sets)
  - $\mathbb{Z}$, discrete features (totally ordered sets)
  - Categorical features (unordered sets)



PCA, ICA, and other Inverse Mixing Models

- One way to learn over distinct real-valued node feature spaces:
  - Assume features of node i in domain m is: $X_{i,m} = \mathbf{H}_m Z_i$, where
    - $Z_i \sim \mu$ sampled for node i from some distribution μ in a common feature space across domains (**not** sampled independently with respect to other nodes in the graph)
    - $\mathbf{H}_m$ is a source mixing matrix for domain m
  - Goal:
    - Find $\mathbf{H}_m^{-1}$ for each domain as to project features in the same feature space
- **Limitations for Graph Foundation Model use:**
  - Inverse map obtained via test-time adaptation (solving optimization on test data)
  - Inverse map may need to depend on graph structure (for most methods they are not)
  - No categorical features (unordered sets)
  - Inverse function space must be known (often restricted to linear maps)

R., "A Mathematical Framework for Graph Foundation Models", in preparation

A Different Paradigm:
New Equivariances for Feature Space Embedding

A Special Case of this New Paradigm: Relation Types in Knowledge Graphs

Double Equivariance for Inductive Link Prediction for Both New Nodes and New Relation Types

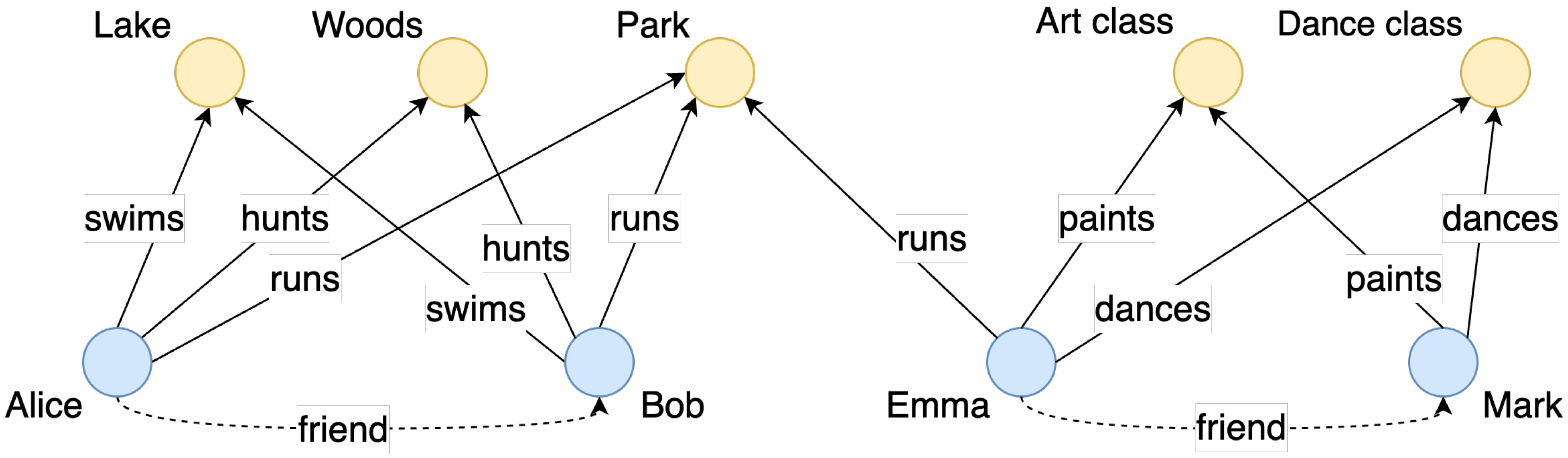
Jianfei Gao, Yangze Zhou, Jincheng Zhou, Bruno Ribeiro

https://arxiv.org/abs/2302.01313

Further Equivariances in Knowledge Graphs

Note that **entity** and **relation ids** are arbitrarily defined.

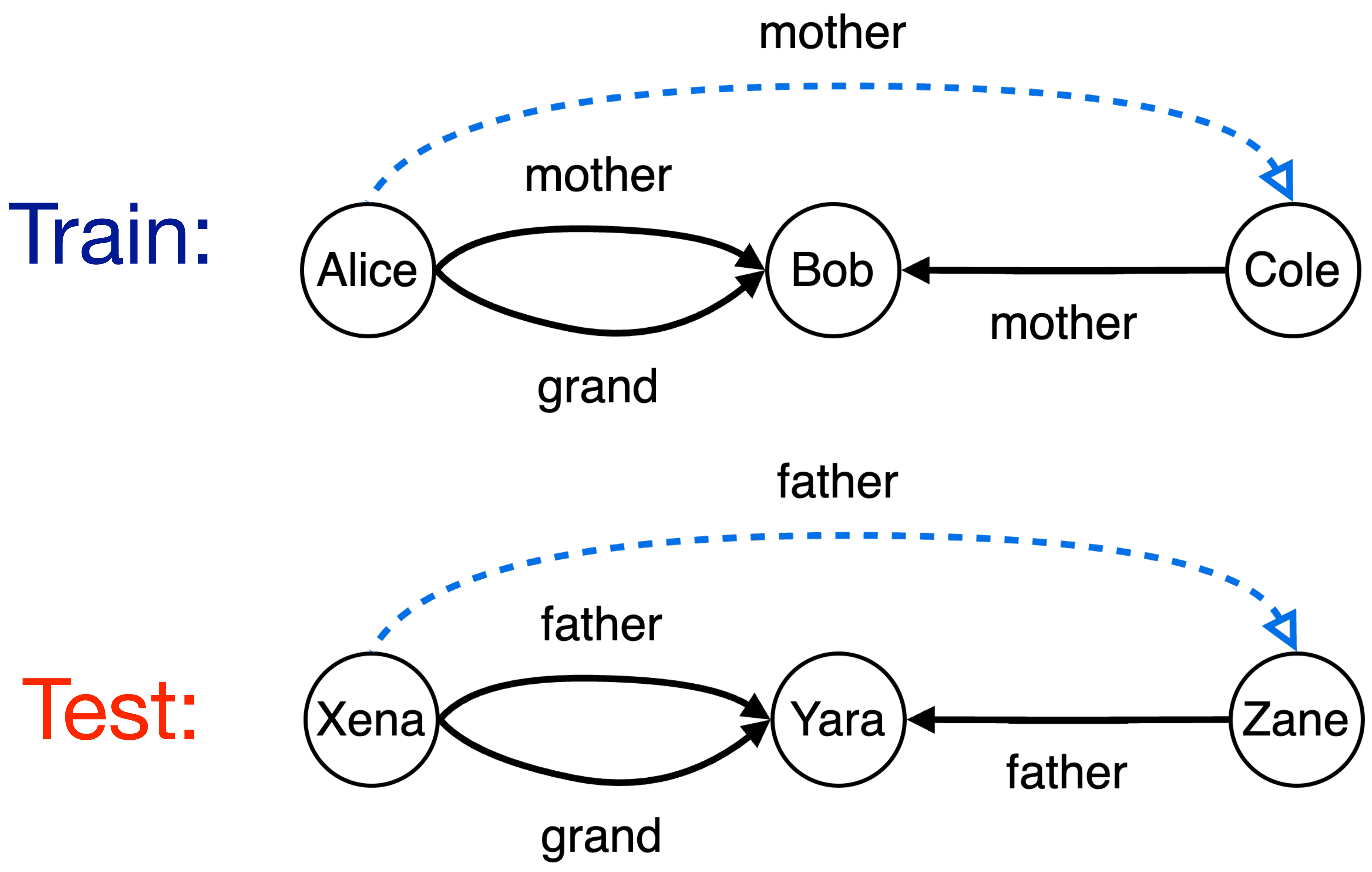
- Pattern transferability: E.g.: Common interests (relations) could imply friendship, regardless of what the interests are.



Double-Equivariance

Solution: assume permutation **symmetries of both entity ids and relation ids**, a notion dubbed **Double-Exchangeability** [Gao et al., 2023].

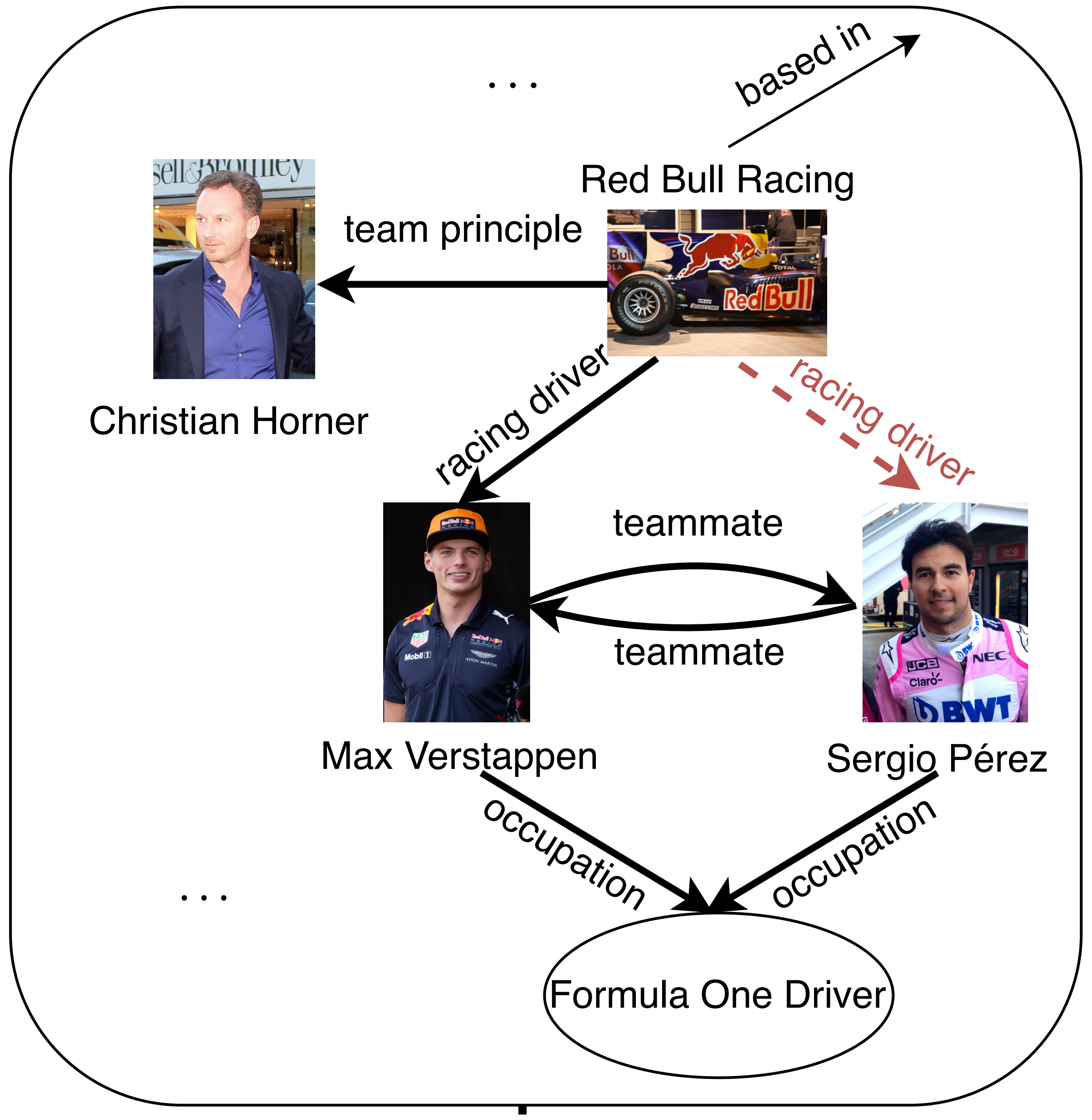
Double equivariant models can learn **higher-order logical relations** beyond what can be learned from data alone.



Transferability over Multiple Domains?

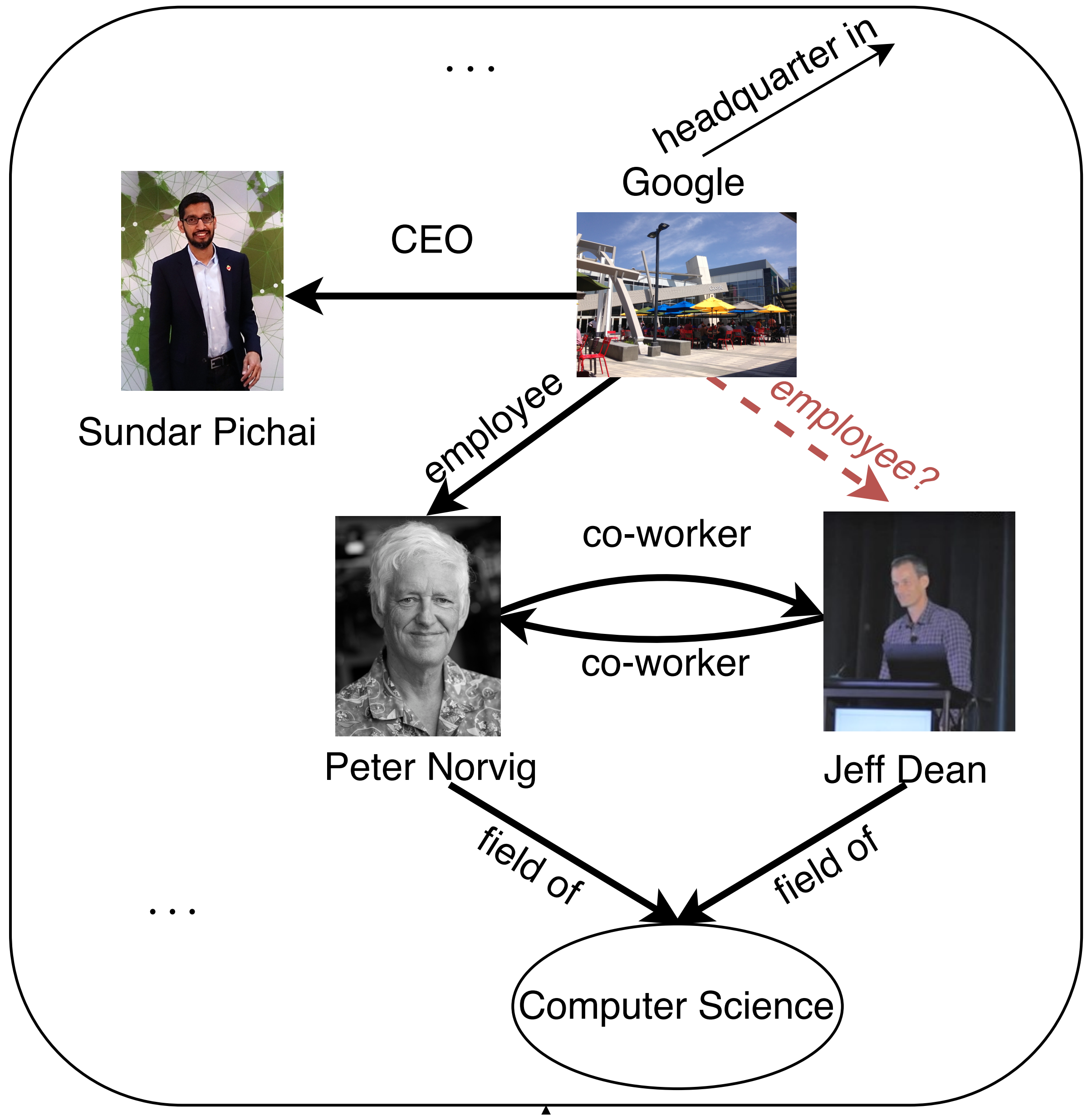
- Can we **transfer** the **relational patters** we learn in Sports to predict relations in Organizations?

Train data: Sports



Change in domain

Test data: Organizations



Our Benchmark: Wikipedia KG Domains

- We created a benchmark for pre-training, zero-shot transferability
- Domains have non-overlapping entities and relations

Domain KG index	Abbreviation	Description
T1	Art	Art and Media Representation
T2	Award	Award Nomination and Achievement
T3	Edu	Education and Academia
T4	Health	Health, Medicine, and Genetics
T5	Infra	Infrastructure and Transportation
T6	Loc	Location and Administrative Entity
T7	Org	Organization and Membership
T8	People	People and Social Relationship
T9	Science	Science, Technology, and Language
T10	Sport	Sport, and Game Competition
T11	Tax	Taxonomy and Biology

	#Nodes	# Relations	#Triplets (Obv.)	#Triplets (Qry.)	Avg. Deg.
Art	10000	45	28023	3113	6.23
Award	10000	10	25056	2783	5.57
Edu	10000	15	14193	1575	3.15
Health	10000	20	15337	1703	3.41
Infra	10000	27	21646	2405	4.81
Loc	10000	35	80269	8918	17.84
Org	10000	18	30214	3357	6.71
People	10000	25	58530	6503	13.01
Sci	10000	42	12516	1388	2.78
Sport	10000	20	46717	5190	10.38
Tax	10000	31	19416	2157	4.32

New Knowledge Graph Isomorphism

If

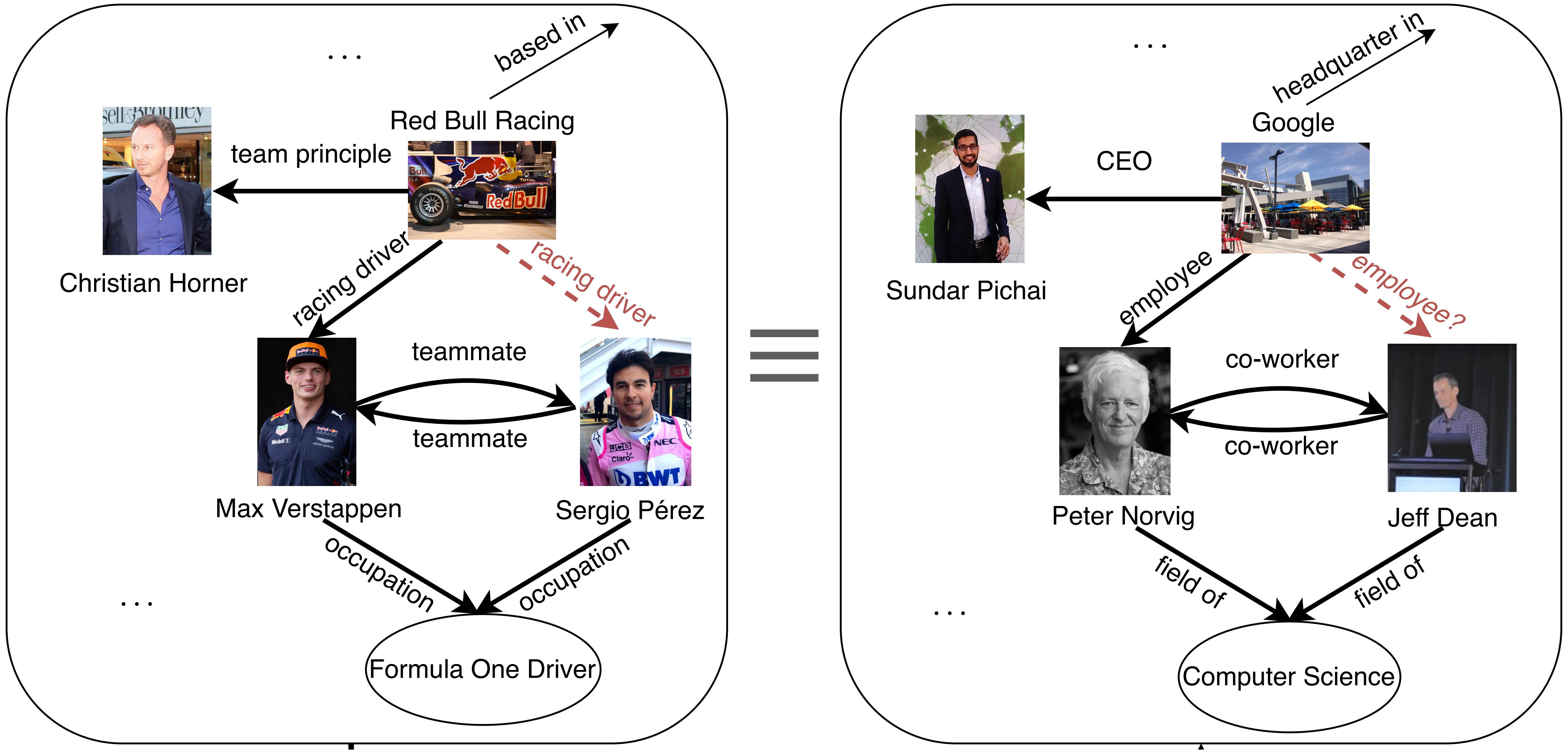
1. $\mathbf{A} \neq \mathbf{A}'$, and
2. $\exists \pi_e \in \mathbb{S}_n, \exists \pi_r \in \mathbb{S}_m,$

$$\mathbf{A}' = \pi_{\mathrm{e}} \circ \pi_{\mathrm{r}} \circ \mathbf{A}$$

we say $\mathbf{A} \equiv \mathbf{A}'$ are **isomorphic**

Train: Sports

Test: Organizations



Double equivariance for knowledge graph predictors

GNN² → Double G-equivariant GNN to node and relation permutations

Double equivariant GNN

$$\hat{\theta}, \hat{\theta}' = \underset{\theta,\theta'}{\text{argmin}} - \log p_{\theta'}(y_{i,r',\star} \,|\, \text{GNN}^2_\theta(r', (A_1, \ldots, A_r))_{ir'}), \text{label } i \text{ and } r')$$

Double equivariance guaranteed by architecture

$$\text{s.t. } \text{GNN}^2_\theta(r', \pi' \circ (\pi \circ A_1, \ldots, \pi \circ A_r)) = \pi \circ \text{GNN}^2_\theta(\pi' \circ r', (A_1, \ldots, A_r)),$$

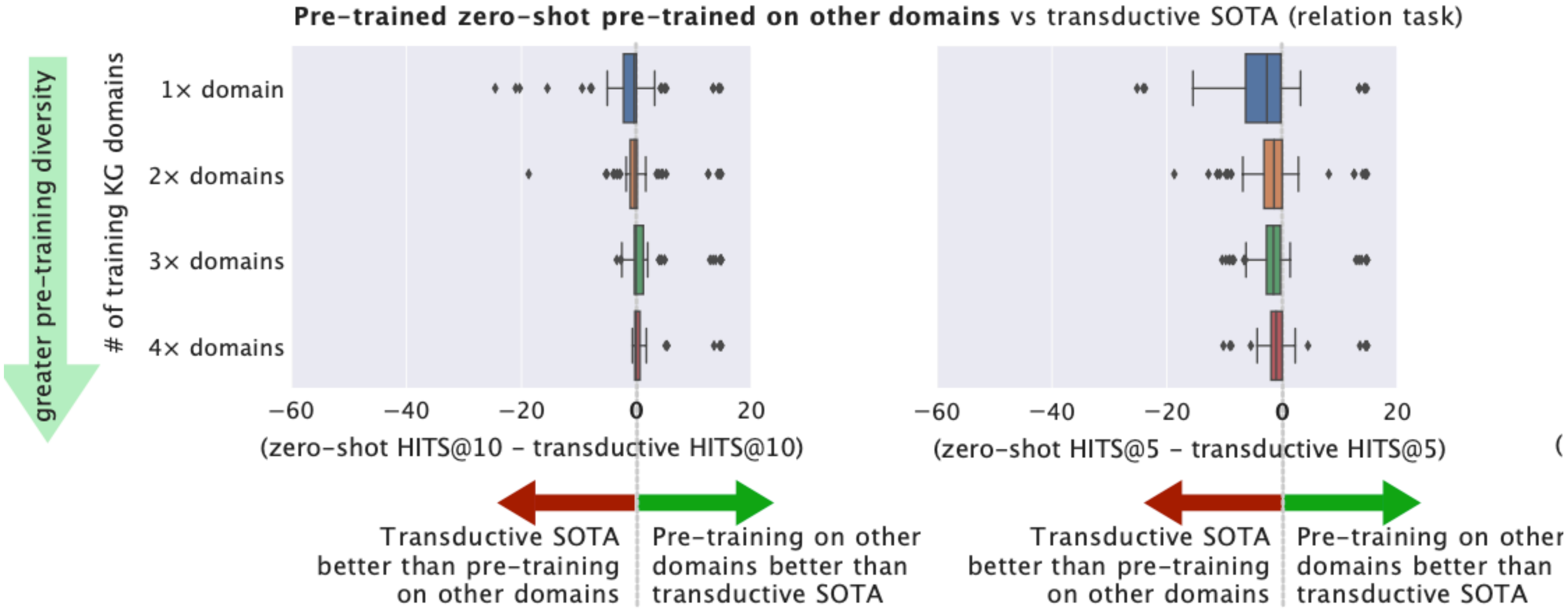
$$\forall \pi' \in \mathbb{S}_r, \forall \pi \in \mathbb{S}_n, \forall A_1, \ldots, A_r$$

For all relation sequence permutations

For all graphs

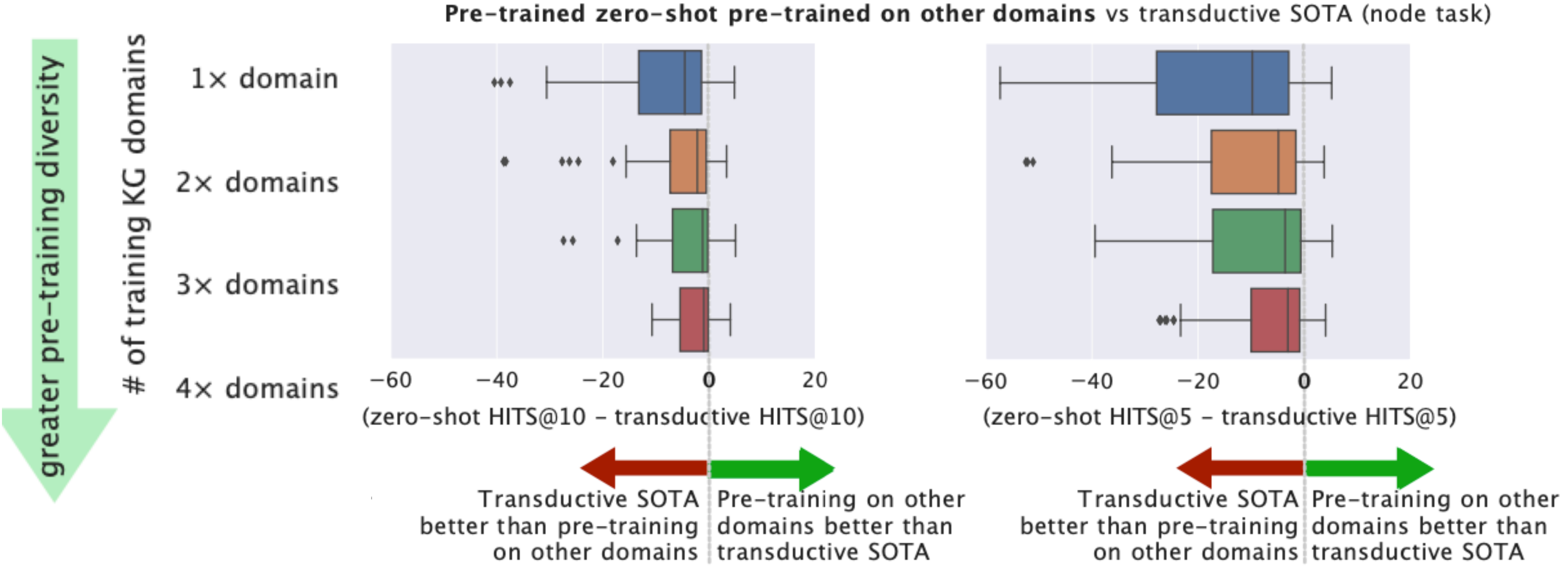
Transferability through double equivariance (1)

- **Pretrain** up to **4x domains, zero-shot test on new domain (no overlapping relations)** to predict relation **(i,?,j)**



Transferability through double equivariance (2)

- **Pretrain** up to **4x domains, zero-shot test on new domain (no overlapping relations)** to predict relation **(i,r,?)**



Examples of double-equivariant models in the literature

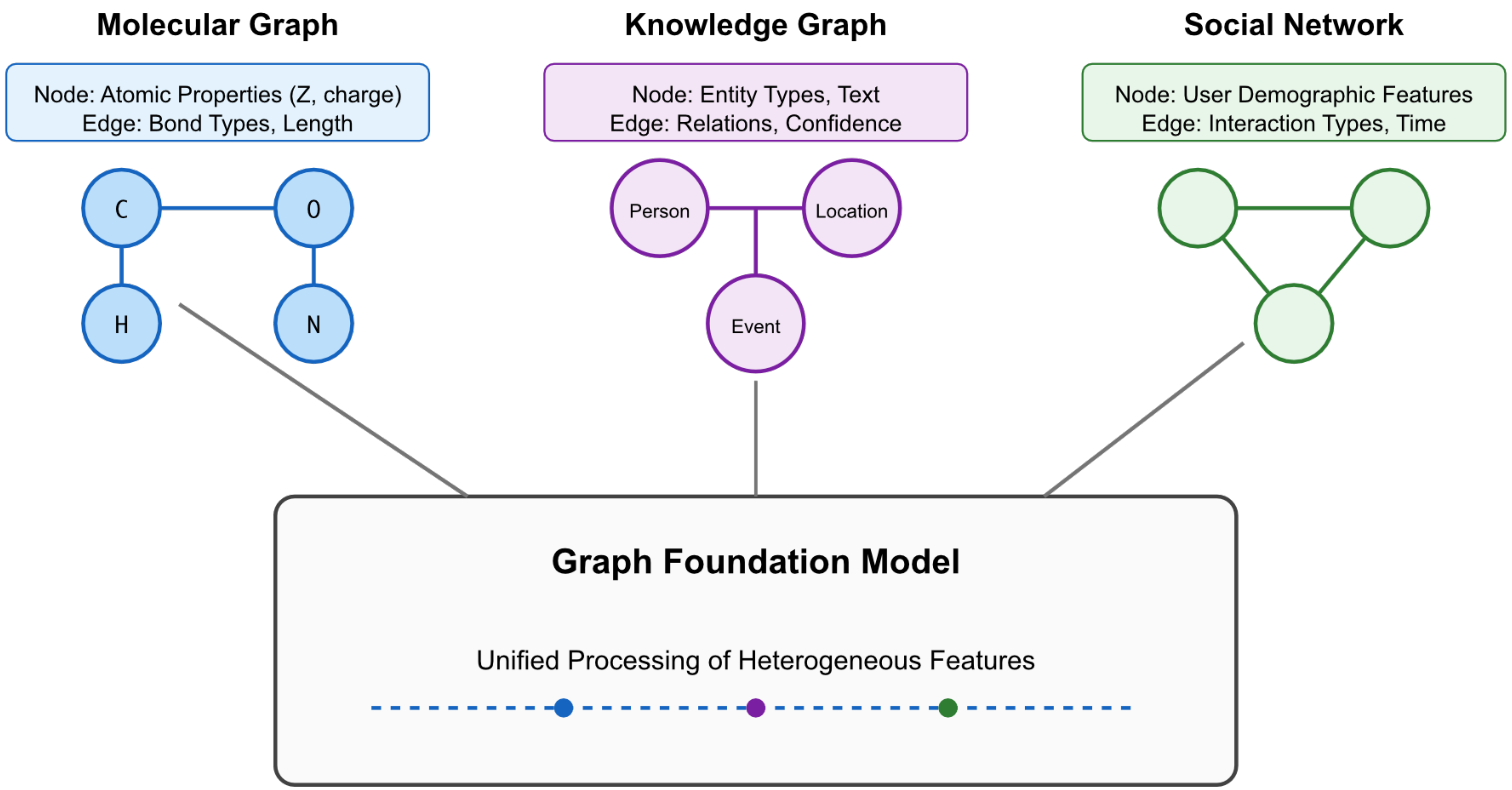
- (**ISDEA+**) (Gao et al., 2023) arXiv:2302.01313
- (**ULTRA**) (Galkin et al., ICML 2024)
- (**ULTRA-Query**) (Galkin et al., NeurIPS 2024)
- (**INGRAM**) (Lee et al, ICML 2023)* [*embeddings equivariant in distribution, see (Gao et al., 2023)]

Can we further generalize this approach?

(Recap) Maybe an Impossible Dream

Pretrain a single foundation graph model over multiple graphs with distinct feature spaces

- Node/edge features can be a **mix** of
  - $\mathbb{R}$, real-valued features (totally ordered sets)
  - $\mathbb{Z}$, discrete features (totally ordered sets)
  - Categorical features (unordered sets)



Further Generalization of Feature Domain Transferability

Zero-Shot Generalization of GNNs Over Distinct Attribute Domains,
Shen, Zhou, Bevilacqua, Robinson, Kanatsoulis, Leskovec, Ribeiro, 2024 *under submission*

Distinct Feature Domains (Example)

- Domain e-commerce **beds**
- Features:
  - Type: ’Twin’, ’Twin XL’, ’Full’, ’Queen’, ’King’, ’California King’
  - Material: ‘Wood’, ’Metal’, ’Upholstered’, ’Bamboo’, ’Particle Board’, ’Composite’
  - Bed frame included: True/False
  - Headboard included: True/False
  - Footboard included: True/False
  - Box spring required: True/False
  - Weight capacity lbs: int
  - Bed size length (inches): int
  - Bed size width (inches): int
  - Bed size height (inches): int

- Domain e-commerce **H&M (clothes)**
- Features:
  - Product type name: categorical
  - Graphical appearance name: categorical
  - Color: categorical
  - Perceived color: categorical
  - Perceived color master name: categorical
  - Department name: categorical
  - Index group name: categorical
  - Section name: categorical
  - Garment type: categorical

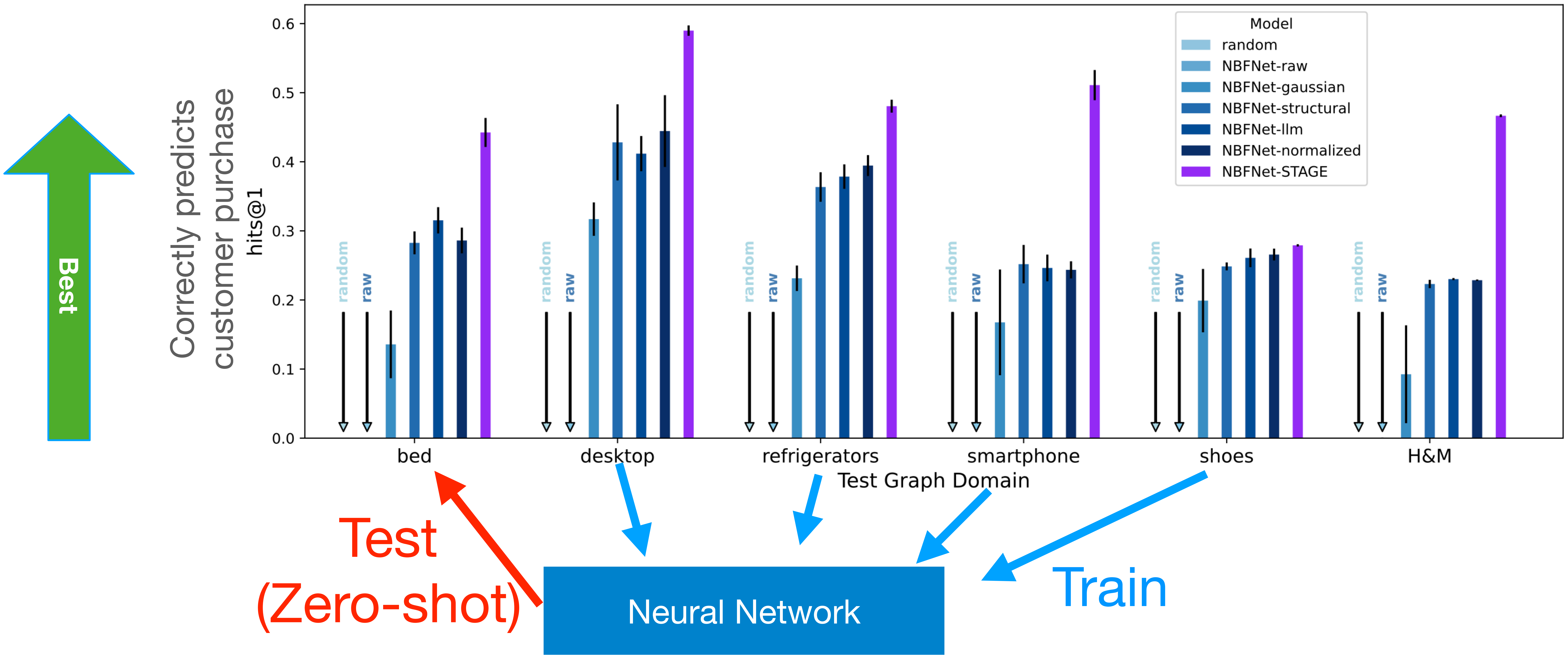
Distinct Feature Domains (Example)

- Domain e-commerce **smartphones**
- Features:
  - Display type:’OLED’, ’LCD’
  - Display size (in): float
  - Display resolution (pixels): <int,int>
  - Processor type: categorical
  - Ram (GB): int
  - Storage (GB): int
  - “Rear camera primary resolution (MP): int
  - Front camera resolution (MP): int
  - Operating system:’Android’, ’iOS’, ’HarmonyOS’, ’KaiOS’, ’Tizen’, ’Ubuntu Touch’, ’PureOS’, ’Sailfish OS’, ’Plasma Mobile’
  - Battery capacity (mAh): int
  - Has gps: True/False
  - Has nfc: True/False

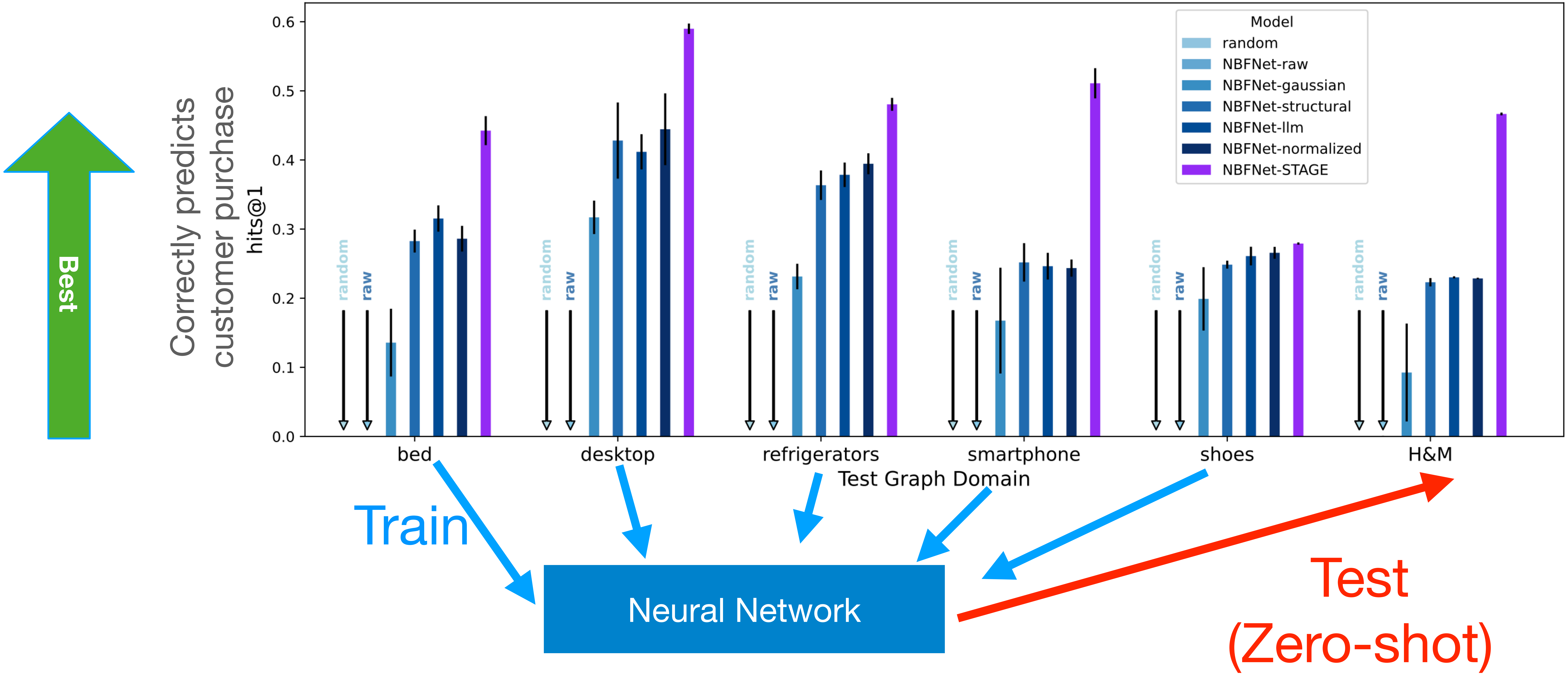
- Domain e-commerce **refrigerators**
- Domain e-commerce **shoes**

⋮

New Equivariances for Encoding Features in GNNs



New Equivariances for Encoding Features in GNNs



A New type of Graph Equivariance to Encode Node Features

- What if instead of learning over the original node features, we learned over the space that defines their **dependencies** with
  - (i) the features of the node's neighbors
  - (ii) the features and the task

- This would be possible if we had a neural network that could learn to perform independence tests over
  - (a) multiple random variables
  - (b) accounts for graph topology

- We would like to translate these requirements into invariances

Finding the right invariances

- Bell (1964) and later Berk & Bickel (1968) proved that a certain type of independence tests (rank tests) have invariances such that they are equivalent to most-powerful almost invariant tests.
  - If most expressive, these tests are called *maximal invariants*
  - *These tests are defined by their invariances.*
  - This means that a neural network can learn independence tests if it abides by the invariances of rank tests.

A New type of Graph Equivariance to Encode Node Features

Component-wise order-preserving groupoids for graphs (COGG) equivariances

- Intuitively, the full set of invariances required for domain transferability over a graph $G = (V, E, X)$, where X are the node features:

1. Invariance or equivariance to transformations of feature values that
   1.1. **Preserve the order statistics** of the feature values of **totally ordered sets**
   1.2. **Invariant to the order statistics** of the feature values of **unordered sets**

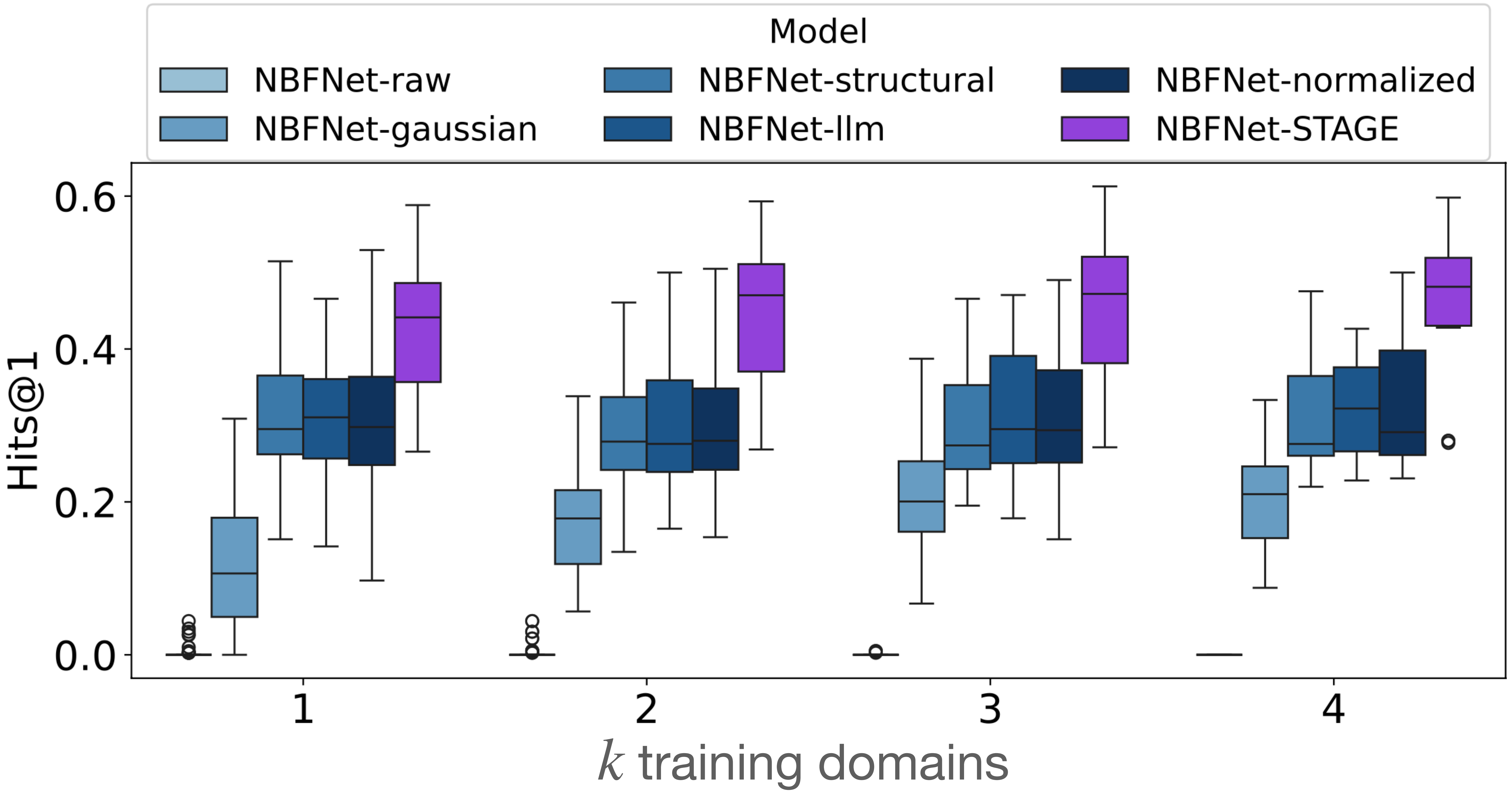
2. Invariance or equivariance to permutations of feature variables
   2.1. i.e., the order of the features should be irrelevant

3. Invariance or equivariance to permutations of entities (nodes) in the graph, affecting both nodes V (and consequently E) and the feature variables in X

Using these equivariances GNNs can generalize across feature domains

- **5 datasets:** E-commerce beds, desktops, refrigerators, smartphones, shoes
- **Task:** Train on $k \in \{1,\ldots,4\}$ dataset domains, zero-shot into held-out dataset.

Training on **more** domains (e.g., beds, desktops, shoes, fridges) **improves** prediction on held-out domain (e.g., smartphones)

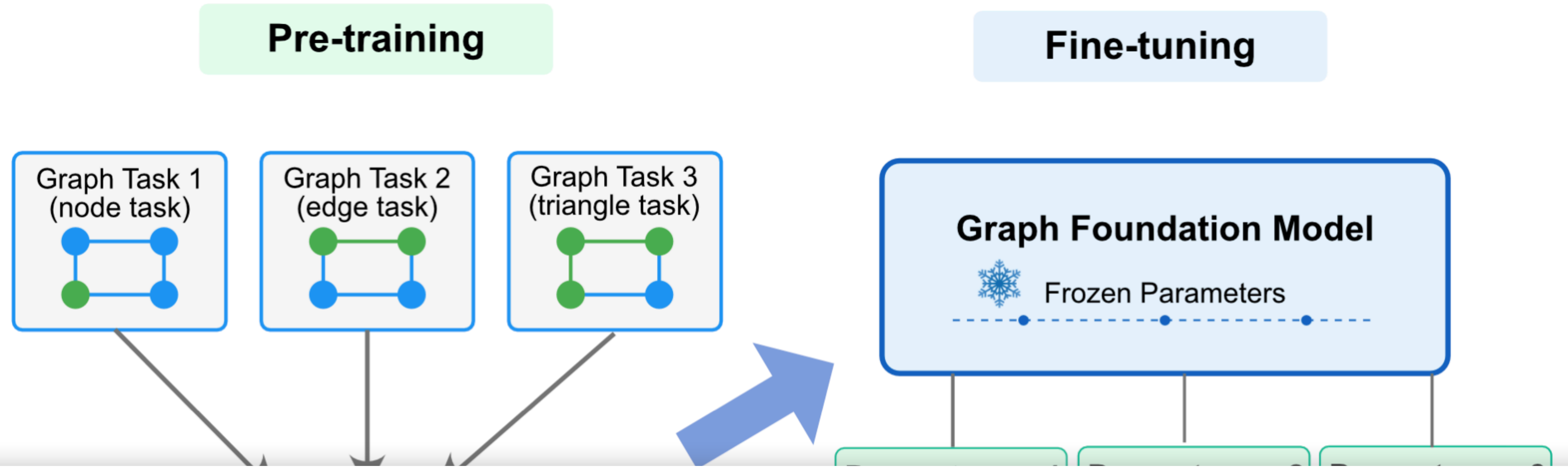


Recap

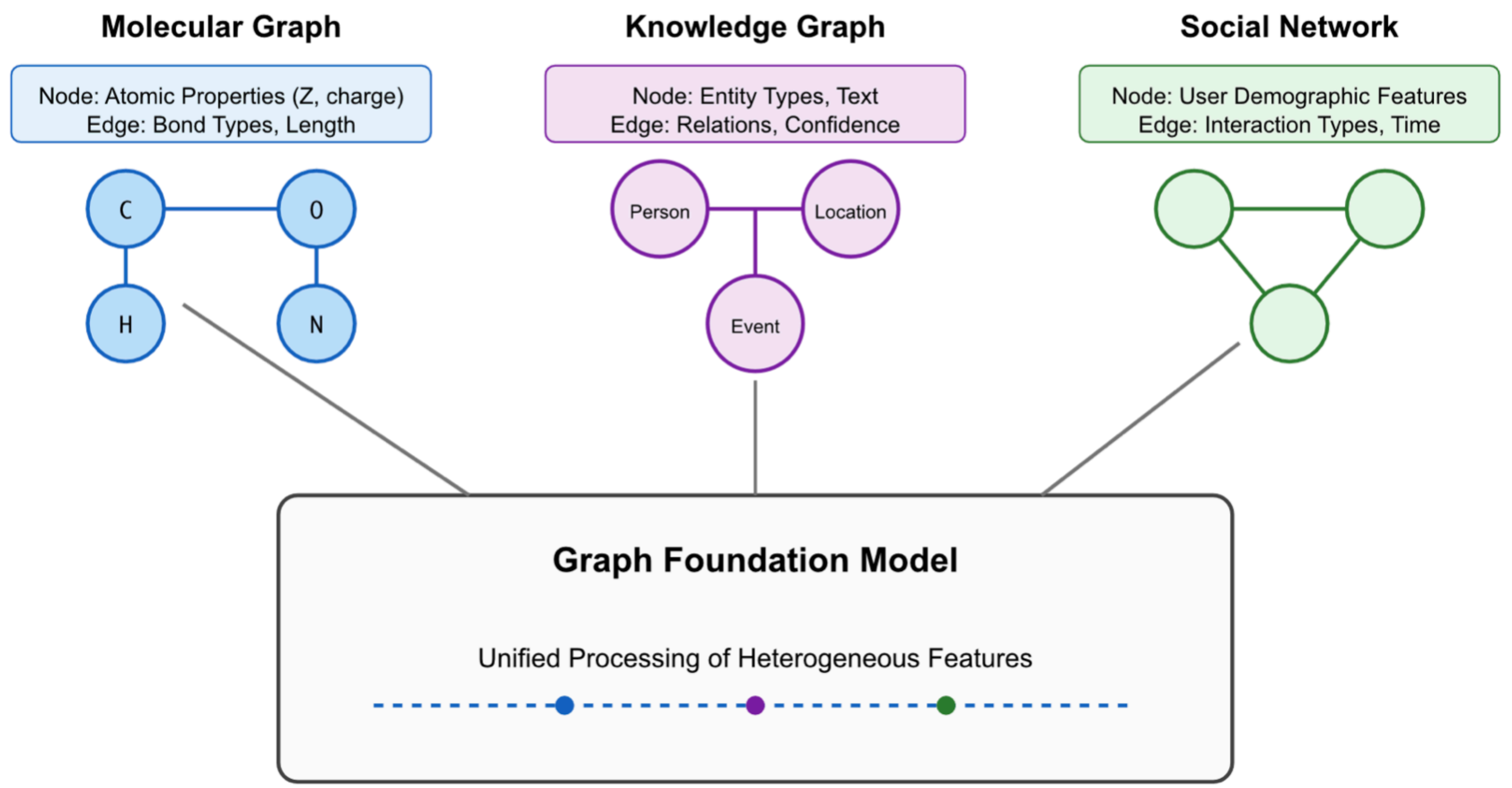
1. **Task transferability**:
- Graph tasks of k nodes need k-permutation symmetries
- New graph representations (holographic) provide first step towards task transferability

1. **Feature space universality**:
- Node/edge feature heterogeneity between domains is a challenging problem
- New neural network symmetries can unleash more universal feature space embeddings by encoding statistical dependencies rather than values

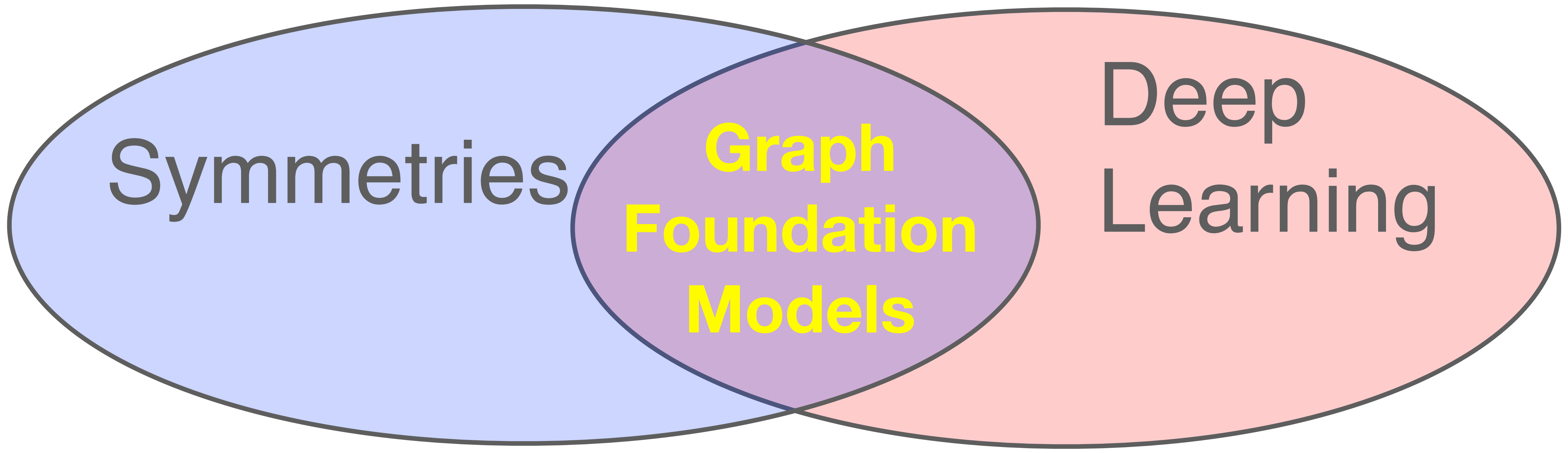
1. **Task transferability:** Graph foundation models should be capable pretraining on a diverse set of tasks **then** fine-tune on diverse downstream tasks



2. **Feature space universality:** Graph foundation models should be able to handle graphs with **heterogeneous node and edge feature spaces** (both categorical, discrete and continuous), allowing for seamless integration of diverse data types and sources



Parting Thoughts



- Imposing new GNN symmetries allows them to have better **task and feature space** transferability

- Graph Foundation Models may not happen through engineering solutions & known methods (e.g., PCA-style) alone
    - GNNs that can learn over diverse graph domains seem to require new neural network symmetries

@brunofmr

ribeiro@cs.purdue.edu

Thank You!

Thank you!

- And thanks to all these amazing collaborators

Beatrice Bevilacqua (Purdue)

Jincheng Zhou (Purdue)

Yucheng Zhang (Purdue)

Leonardo Cotta (UoT/Vector Institute)

Yangze Zhou (Spotify)

Jianfei Gao (Amazon)

S Chandra Mouli (Meta)

Jure Leskovec (Stanford)

Joshua Robinson (Isomorphic Labs)

Yangyi Shen (Stanford)

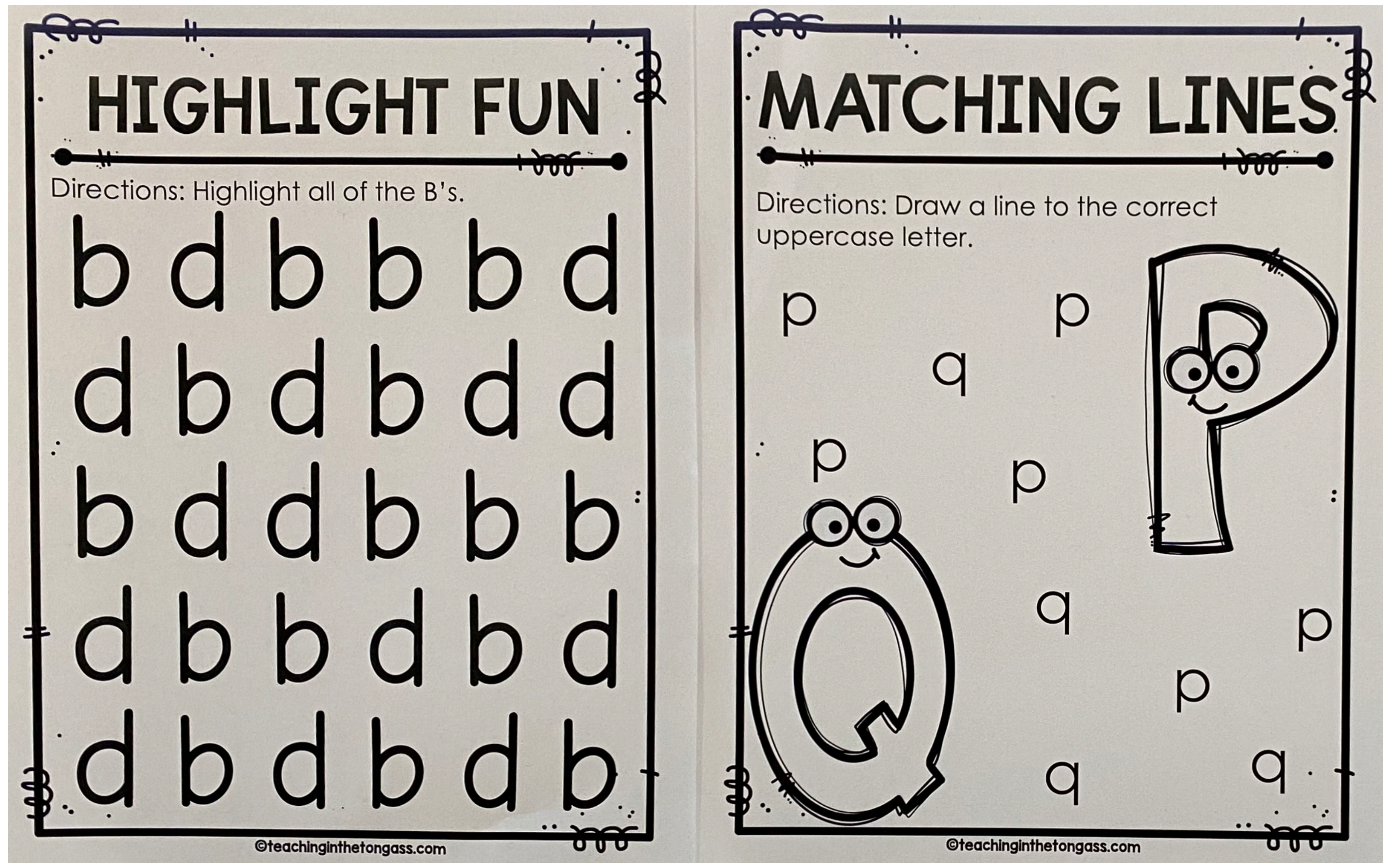
Michael Galkin (Google)

@brunofmr
ribeirob@purdue.edu

(Backup)
Symmetries in Human Relational Reasoning

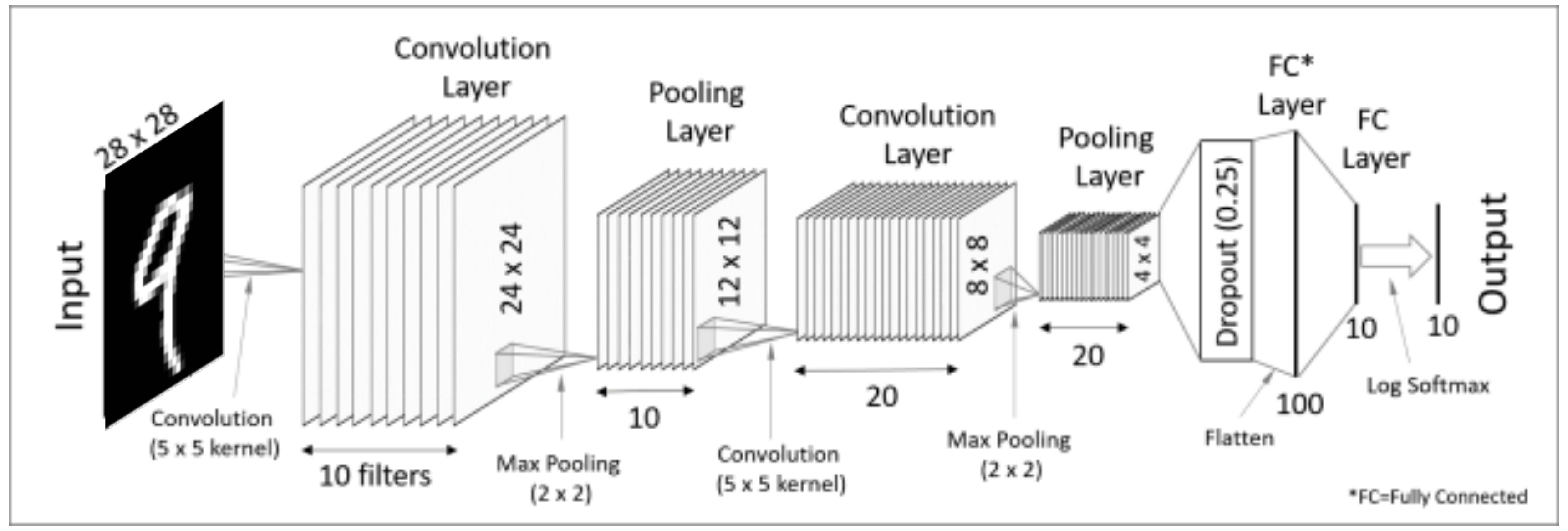
Symmetries in Human Learning

p q d b p q d b p q d b p q d b



His kindergarten homework

VS



Any simple CNN can distinguishing 6 and 9 even before training starts

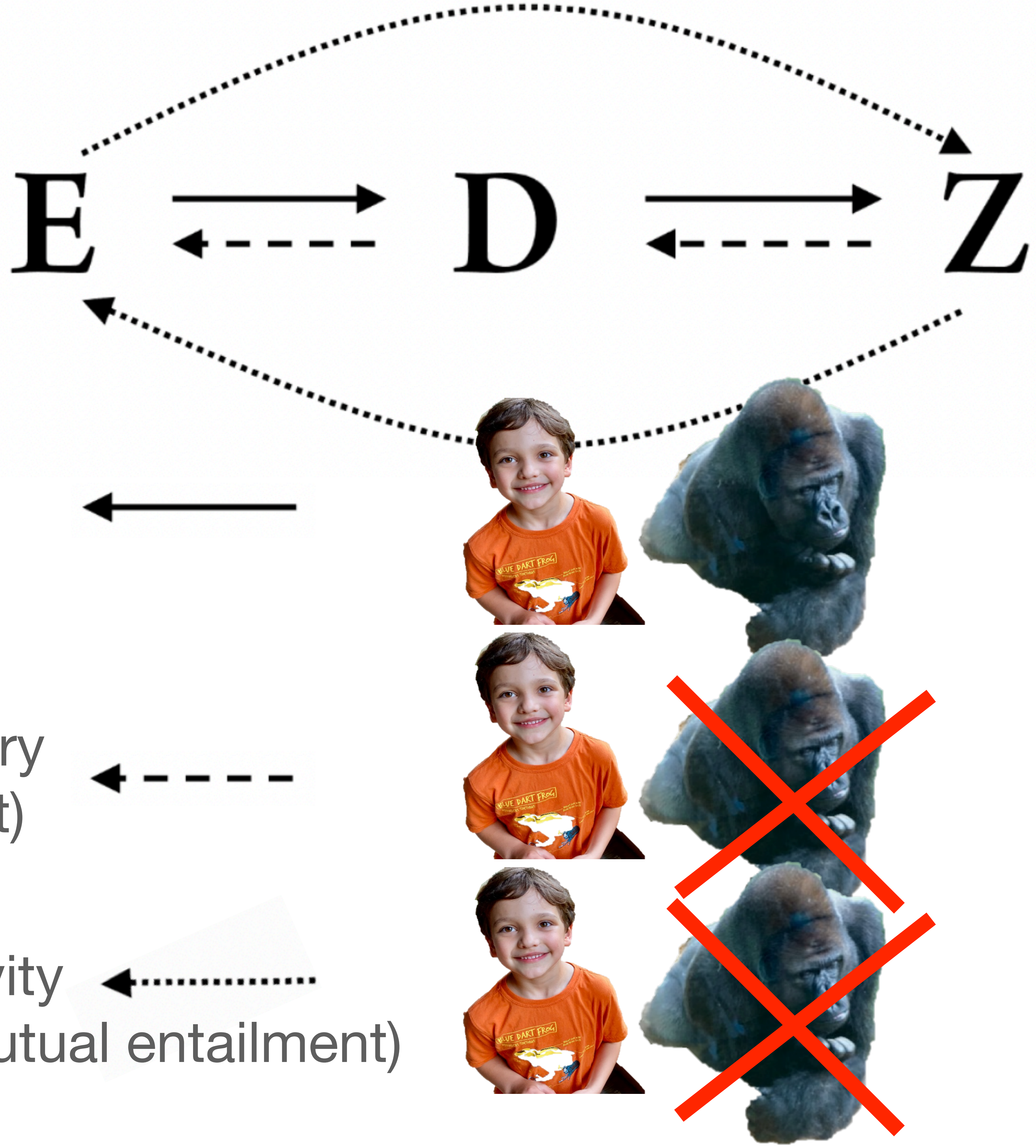
Symmetries and learning

Assuming symmetries is a **key cognitive difference** between the young children and other primates

Young children assume symmetries (and learn asymmetries when wrong)

Human neurons assume some symmetries

Learned
stimuli relations



Directly trained

Assumed symmetry
(Mutual entailment)

Assumed transitivity
(Combinatorial mutual entailment)

[Sidman and Tailby, 1982]
[Sidman et al., 1982, “A search for symmetry in the conditional discriminations of rhesus monkeys, baboons, and children”]

PURDUE UNIVERSITY

Asymmetry Learning for Counterfactually-invariant Classification in OOD Tasks (Mouli, R., ICLR 2022 Oral)

Asymmetry Learning

Hypothesis: **Human brain assumes symmetries** and **learn asymmetries when needed**

Neural networks should also **assume symmetries** and **learn asymmetries when needed**

ASYMMETRY LEARNING FOR COUNTERFACTUAL-INVARIANT CLASSIFICATION IN OOD TASKS

S Chandra Mouli
Department of Computer Science
Purdue University
chandr@purdue.edu

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

NEURAL NETWORKS FOR LEARNING COUNTERFACTUAL G-INVARIANCES FROM SINGLE ENVIRONMENTS

S Chandra Mouli
Department of Computer Science
Purdue University
chandr@purdue.edu

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