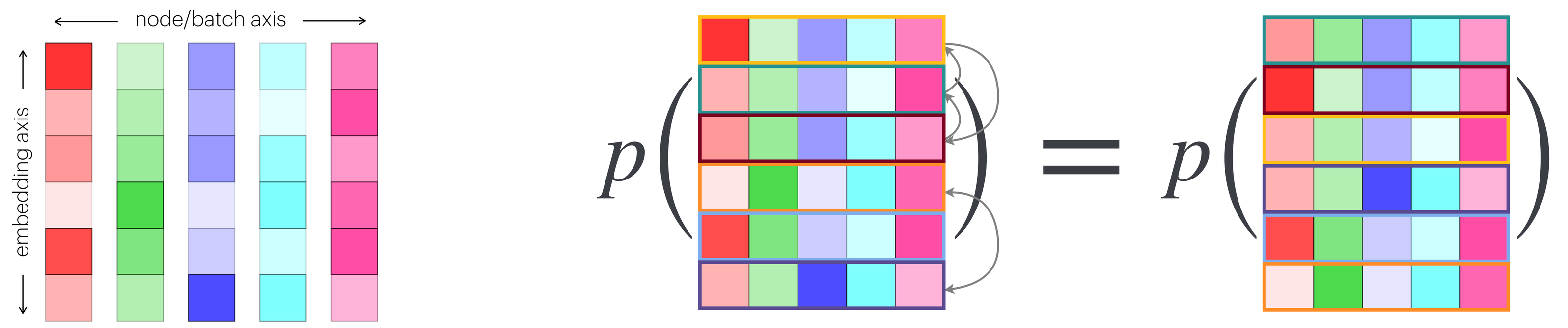
Exchangeability of GNN Representations with Applications to Graph Retrieval

Kartik Nair[1,2], Indradyumna Roy[1], Soumen Chakrabarti[1], Anirban Dasgupta[3], and Abir De[1]

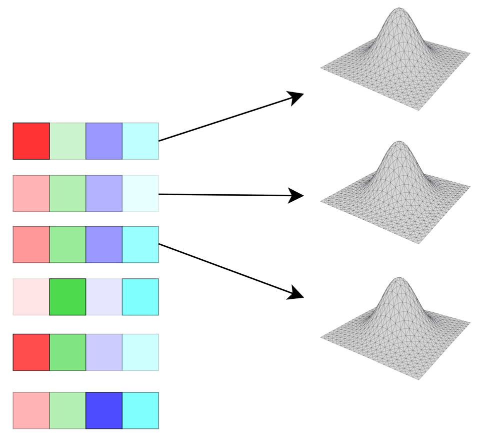
We discover a new form of probabilistic symmetry in neural network representations, which holds for a *variety of architectures* (incl. GNNs) and training objectives.
Tl;dr each co-ordinate of the embedding follows the same distribution. In fact, we show a stronger condition: **the probability density** of said embedding is **invariant to permutations of its co-ordinates**.



A New Axis of (Probabilistic) Symmetry in Neural Representations

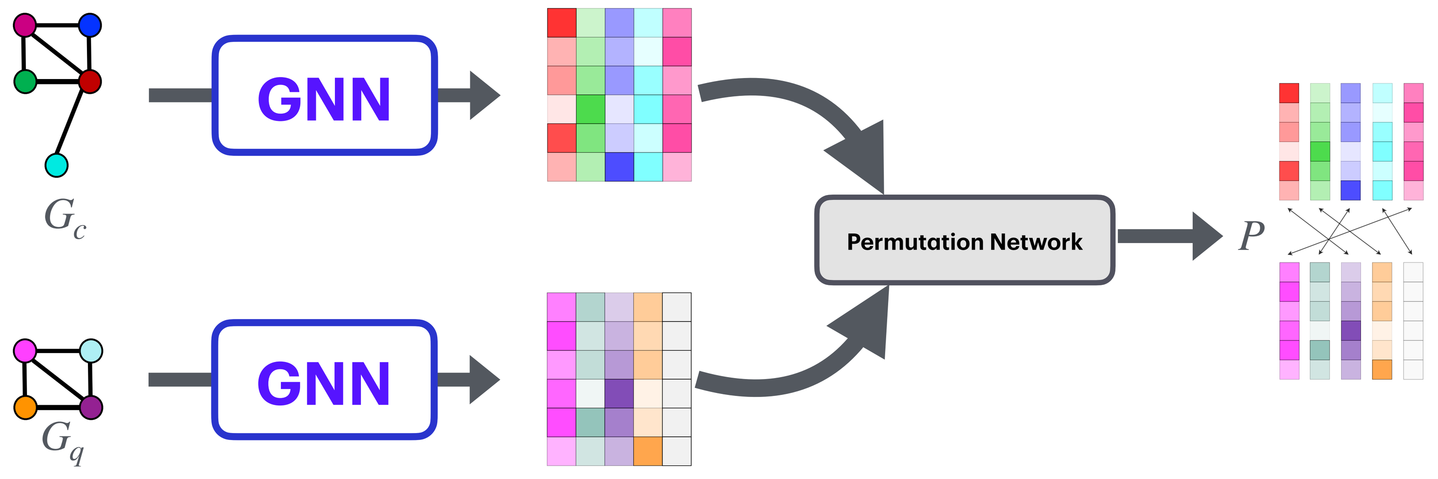
This probability space corresponds of the network parameters, not the input/data distribution. *It is a symmetry in the neural network function.*

This implies: each co-ordinate along the embedding axis has the **same marginal distribution**, and all pairs of co-ordinates share the same distribution (and so on for *higher order combinations*).



The Graph Retrieval Problem: Developing a late-interaction retrieval system for this class of transportation based costs–

A `distance' over sets: matched each query node with corpus node to minimize a *convex, possibly asymmetric cost*



$$\Delta(G_c, G_q) \longrightarrow \min_{\mathbf{P}\in\mathcal{P}_n} \sum_{u,u'} \sum_{d\in[D]} \rho(\mathbf{H}_q[u,d] - \mathbf{H}_c[u',d])\mathbf{P}_{u,u'}$$

Subgraph Matching $\rho(\bullet) = [\bullet]_+$ Graph Edit Distance $\rho(\bullet) = e^+ \times [\bullet]_+ + e^- \times [-\bullet]_+$

The challenge: complex dependence of the node matching. Distance cannot be neatly separated into corpus & query terms.

Except… in 1-D – matching is trivial by sorting

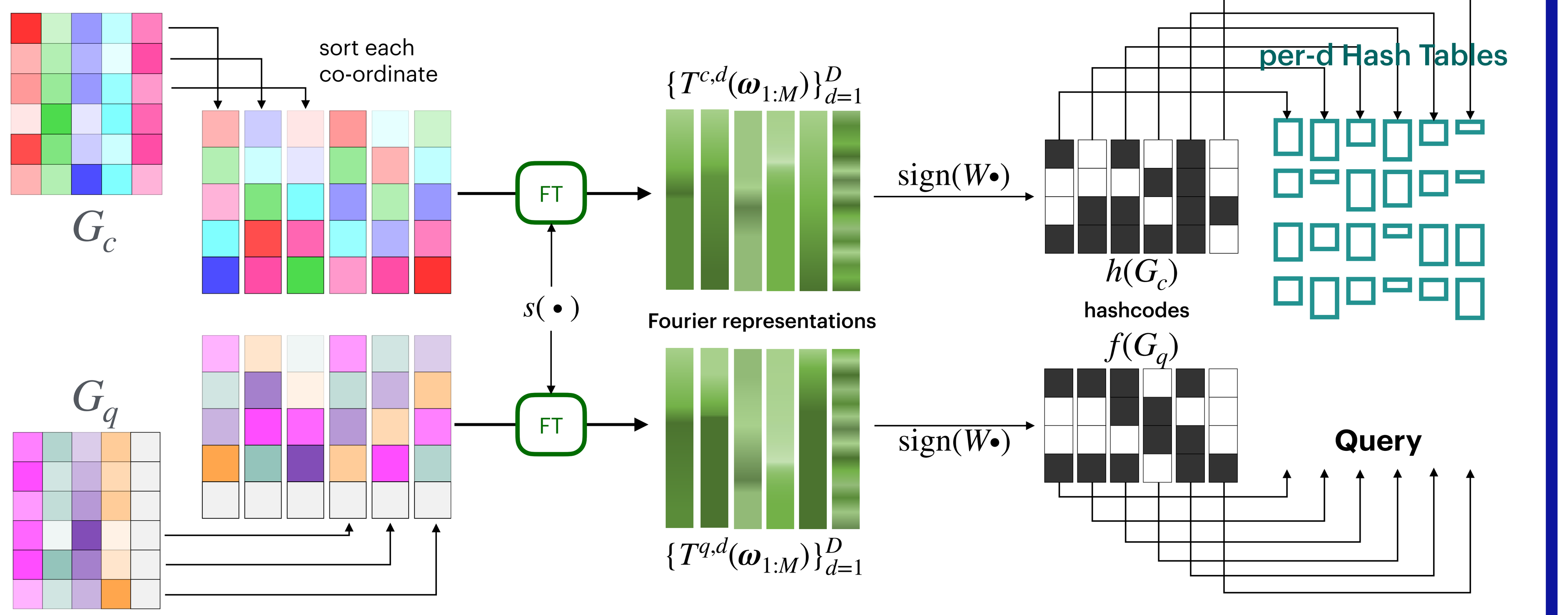
$$\Delta_d(G_c, G_q) \longrightarrow \min_{\mathbf{P}\in\mathcal{P}_n} \sum_{u,u'} \rho(\mathbf{H}_q[u,d] - \mathbf{H}_c[u',d])\mathbf{P}_{u,u'} = \sum \rho(\mathrm{sort}(\mathbf{H}_q[:,d]) - \mathrm{sort}(\mathbf{H}_c[:,d]))$$

Exchangeability implies concentration of distances

$$\Pr(|\Delta(G_c, G_q) - \Delta_d(G_c, G_q)| > \epsilon) < \delta$$

So, an LSH for Δ_d is a `looser' LSH for Δ.

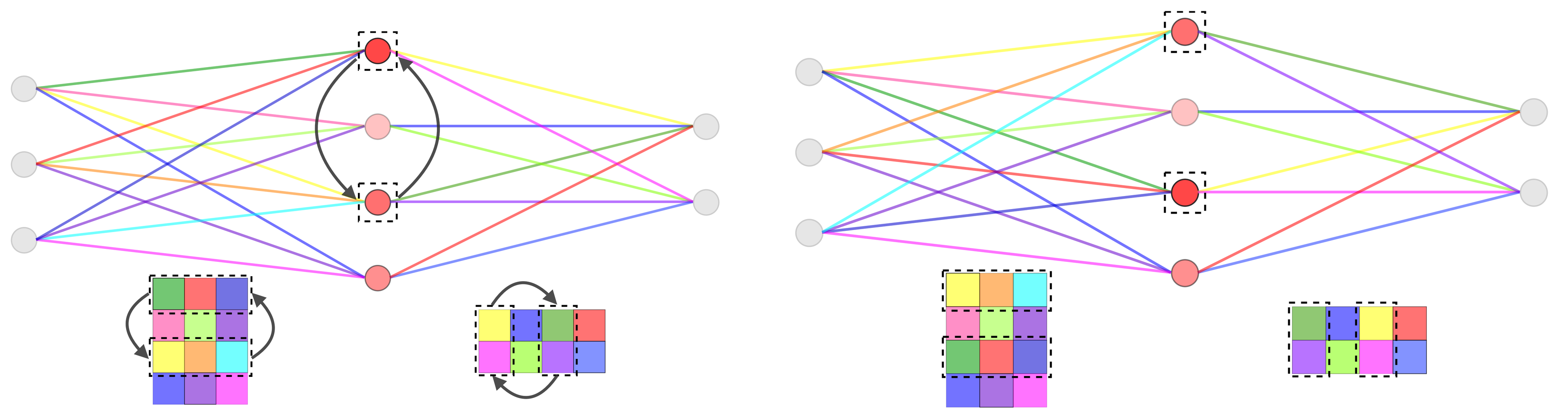
Use multiple hash tables for every co-ordinate. To hash each slice, convert difference to inner product in Fourier space [Roy. et al '23]. Our method, **GraphHash**, summarised –



The Mechanism: Exchangeability arises due to a weight-space symmetry at initialisation, which is maintained at each training step. We require these symmetries to hold –

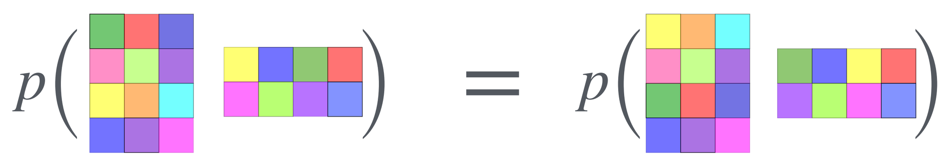
We demonstrate these conditions on a simple example of a 2 layer MLP, where the hidden layer is the representation vector

(1) A class of weight space-transformations that induce corresponding permutations of the representation's co-ordinates, without changing the loss



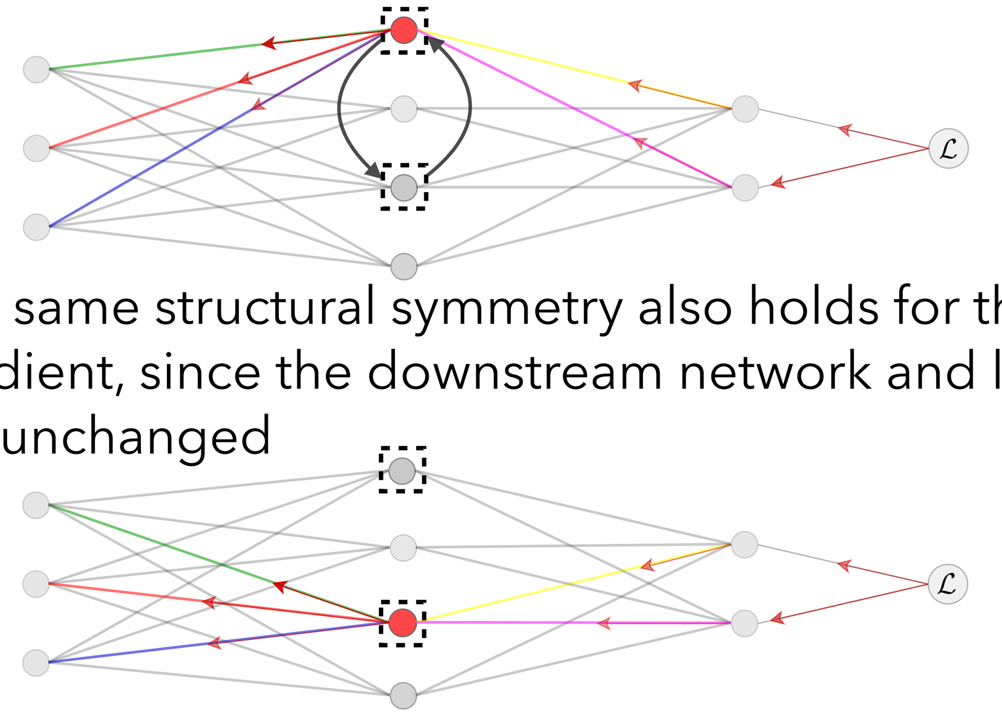
This is achieved by permuting rows/columns of layers appropriately.

(2) Symmetry at Initialization



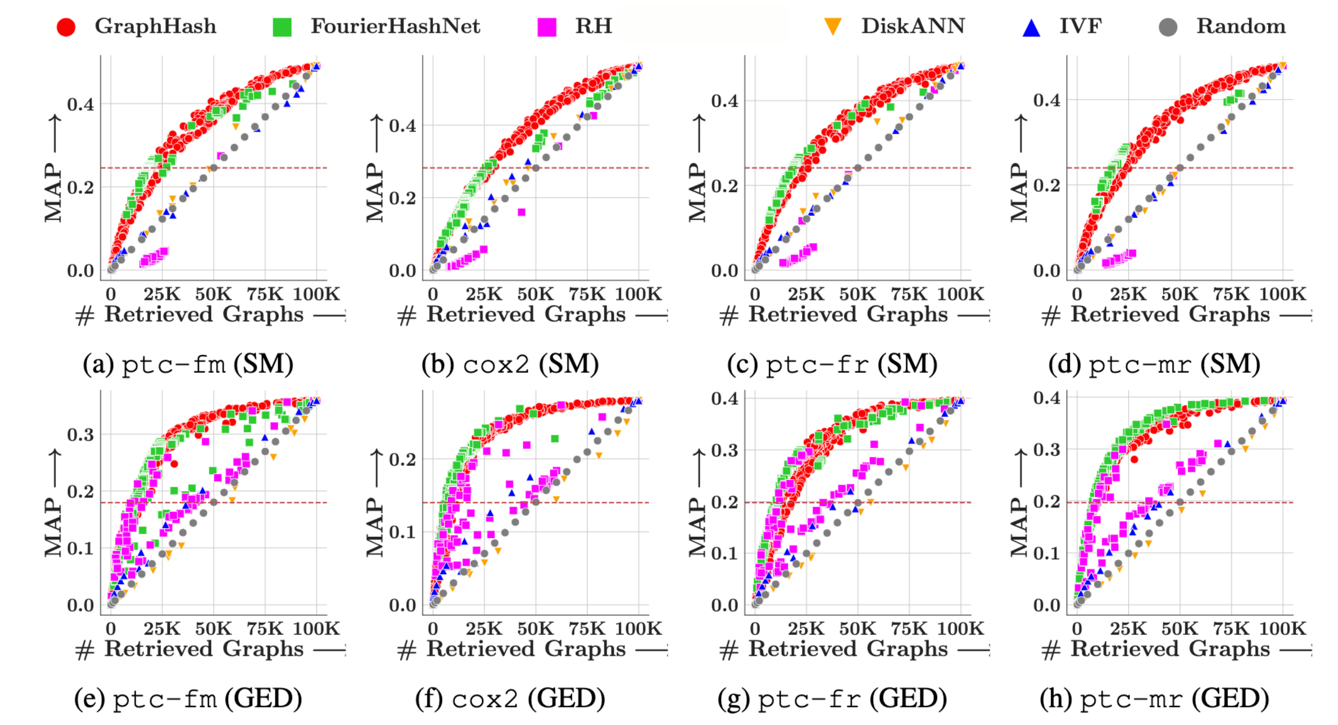
The entries of each weight are initialised i.i.d from a layer-specific distribution. So, the joint densities can be factored into densities of each scalar, which means permutation invariance

(3) Equivariance of the gradient & parameter update



The same structural symmetry also holds for the gradient, since the downstream network and loss are unchanged

Evaluation via the tradeoff between Mean Average Precision* versus #corpus graphs retrieved for different configurations of our pipeline against single vector LSH baselines and node-level lookup baselines. *GraphHash* consistently gives the best trade-off.



[1]IIT Bombay [2]Carnegie Mellon University [3]IIT Gandhinagar

Author contact: ksnair@cs.cmu.edu, {indraroy15,soumen,abir}@cse.iitb.ac.in

* $\mathrm{MAP} = \frac{1}{|\mathcal{Q}_{\text{test}}|} \sum_{G_q \in \mathcal{Q}_{\text{test}}} \mathrm{AP}(G_q)$. $\mathrm{AP}(G_q) = \frac{1}{|\mathcal{C}_{q\oplus}|} \sum_{r=1}^{|\pi_q|} \mathrm{Prec@}r \cdot \mathbb{I}[\Pi_q(r) \in \mathcal{C}_{q\oplus}]$