# Wasserstein Embedding for Graph Learning (WEGL)





Soheil Kolouri
HRL Laboratories

Navid NaderiAlizadeh
HRL Laboratories

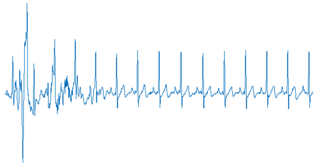
Gustavo K. Rohde
University of Virginia

Heiko Hoffmann
HRL Laboratories

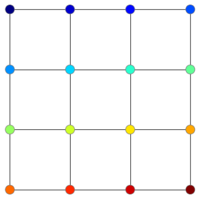

ICLR | 2021

# Learning on (ir)regular data structures

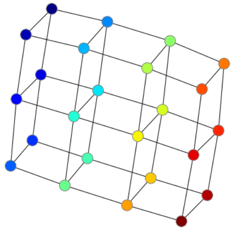
Signals



Images

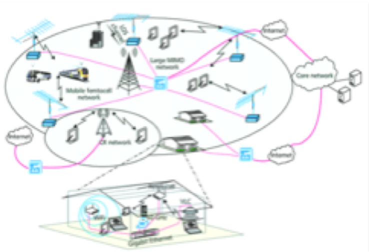


Videos

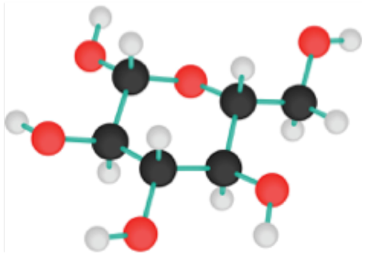


Social networks

Communication systems

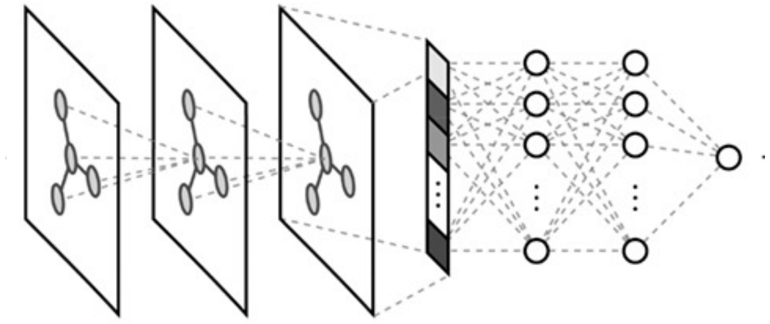


Molecules



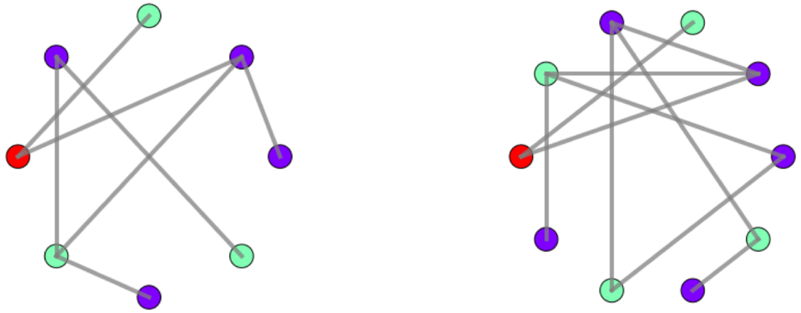
## Graph neural networks (GNNs)

Extending deep neural networks
to graph structured data

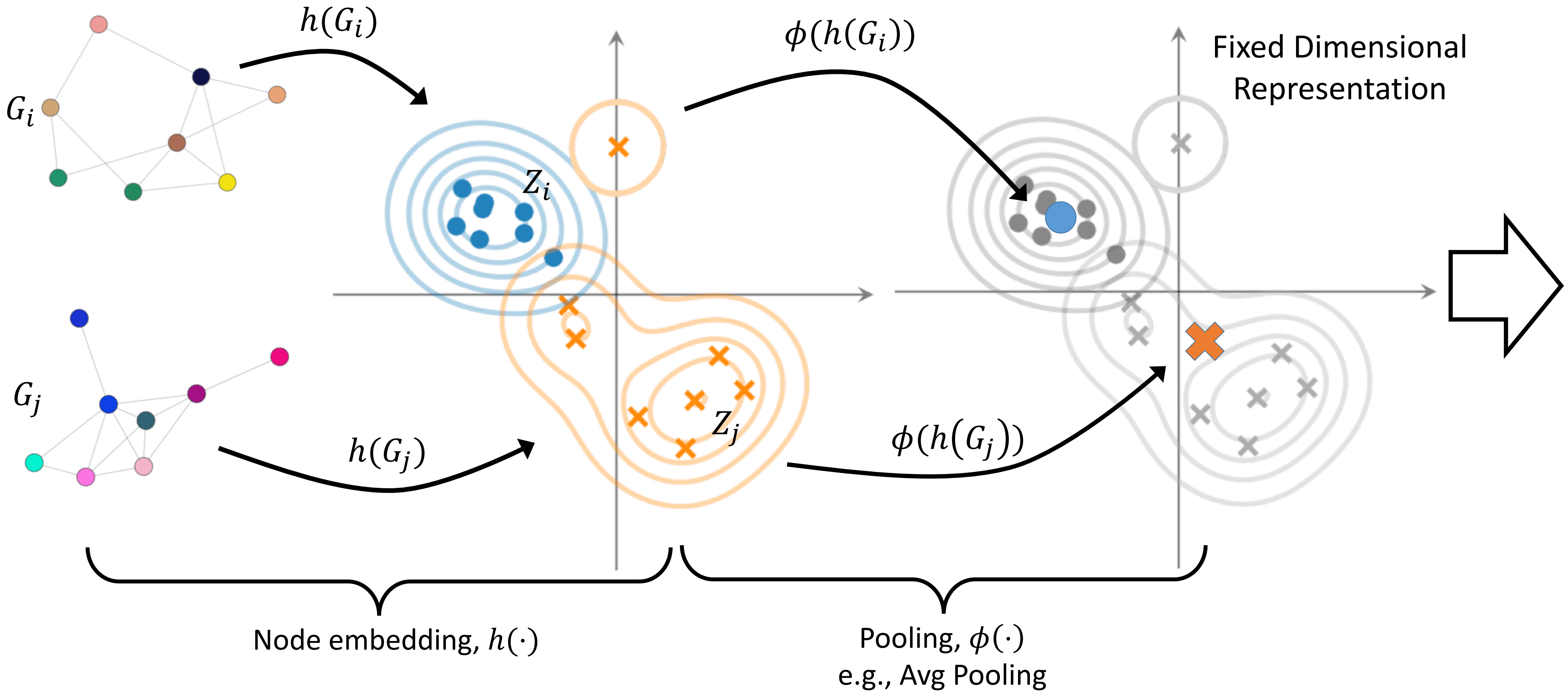


## Graph kernels (GKs)

Measures for estimating pairwise
similarities between graphs



# Common Practice for Graph Embedding

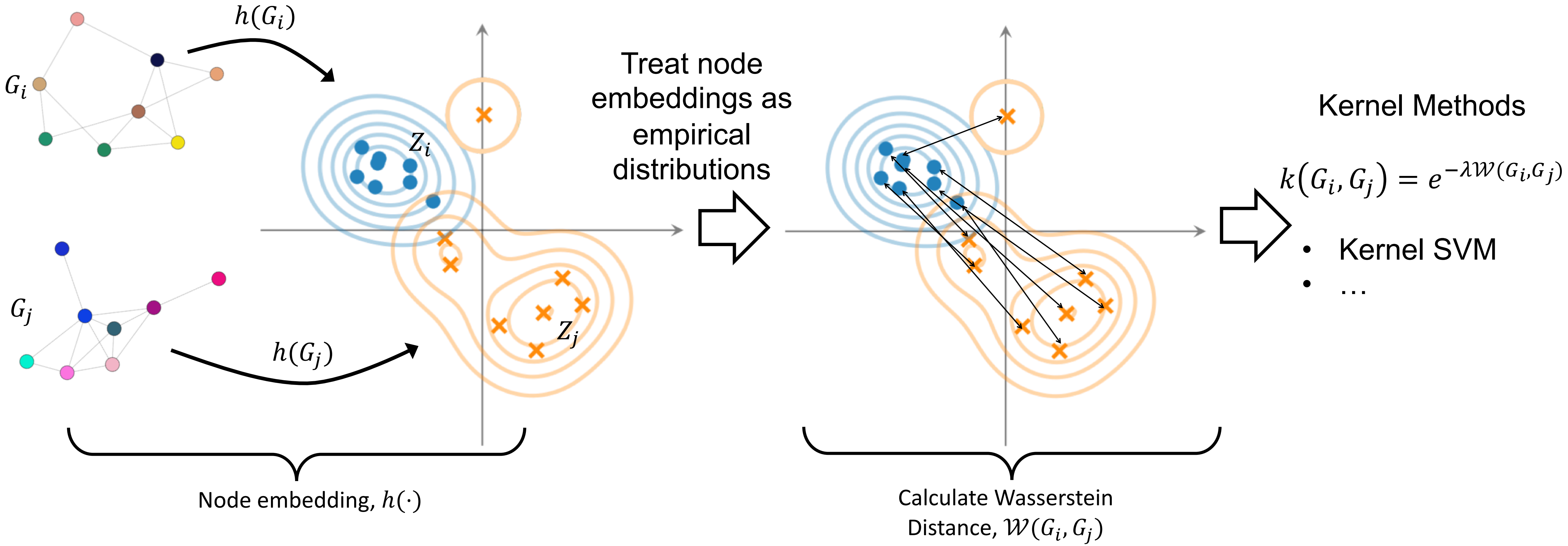



Downstream classification/regression

- Multi-Layer Perceptron
- …

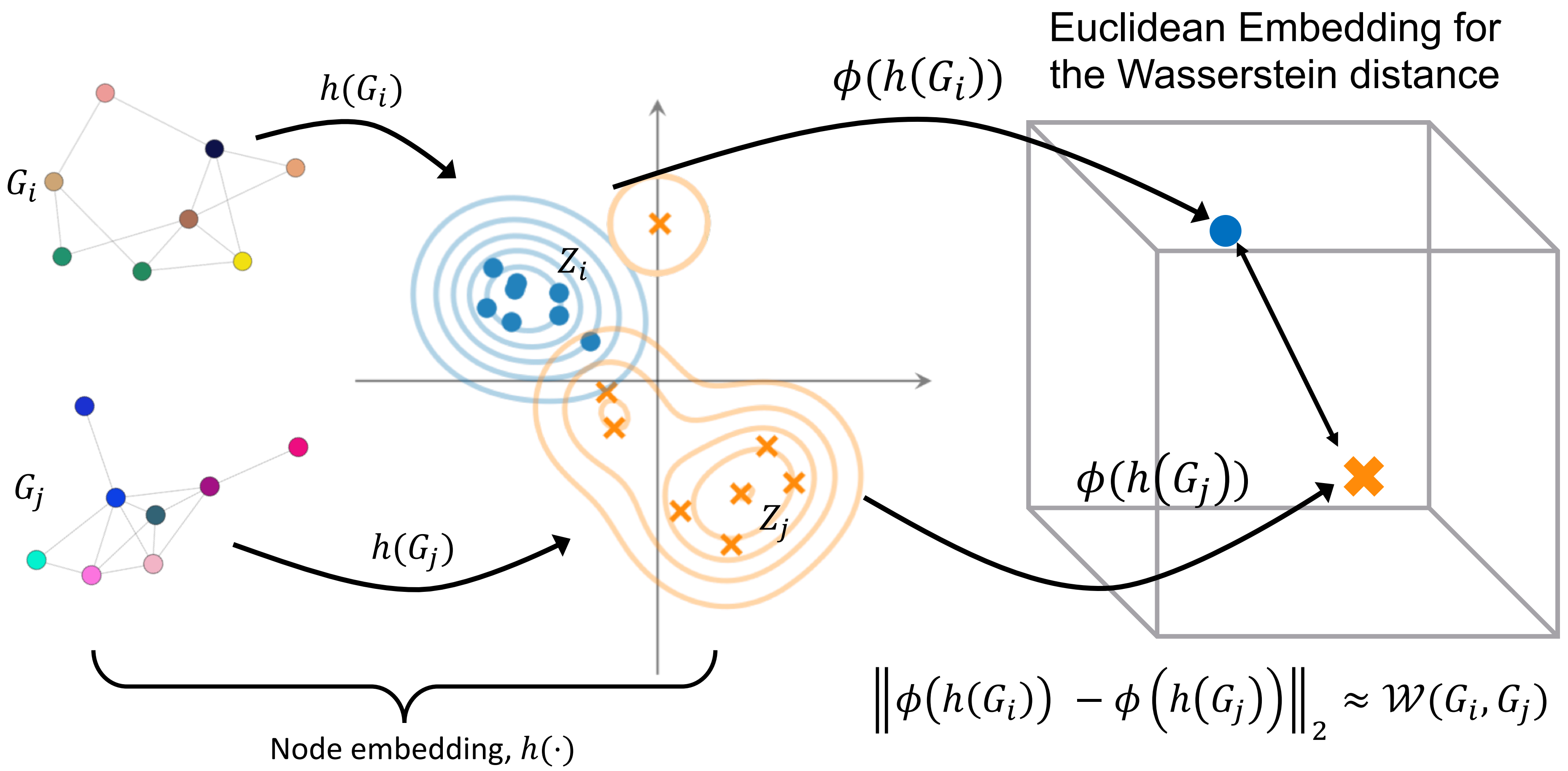
# Wasserstein Weisfeiler-Lehman graph kernels

Togninalli, Matteo, et al. "Wasserstein Weisfeiler-Lehman Graph Kernels." *NeurIPS* 2019.




For a dataset of $M$ graphs, WWL requires $\frac{M(M-1)}{2}$ calculations of the Wasserstein distance for training, and generally speaking $M$ distance calculations for testing

# Main idea



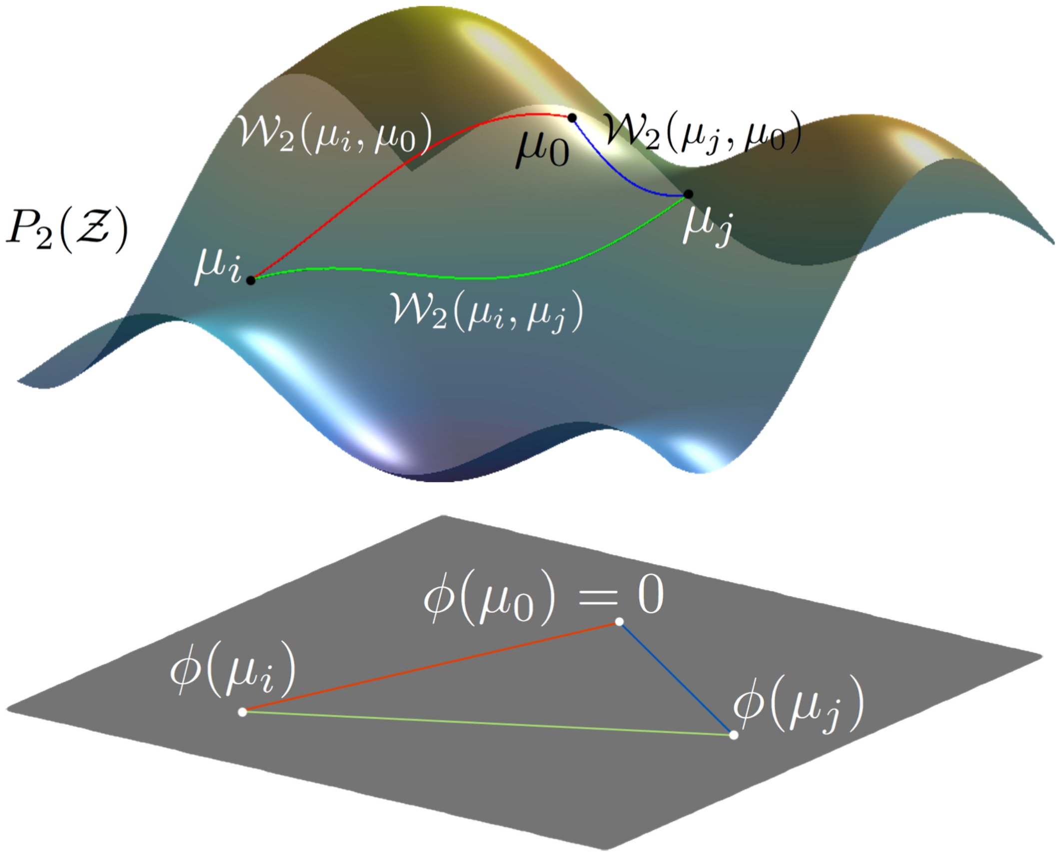

Downstream classification/regression

- Multi-Layer Perceptron
- …

# Linear Wasserstein embedding

- Fix a reference probability measure $\mu_0$ with density $p_0$.
- Let $f_i$ denote the Monge map that pushes $\mu_0$ into a measure $\mu_i$.
- Define $\phi(\mu_i) = (f_i - \text{identity})\sqrt{p_0}$ , which satisfies the following important properties:

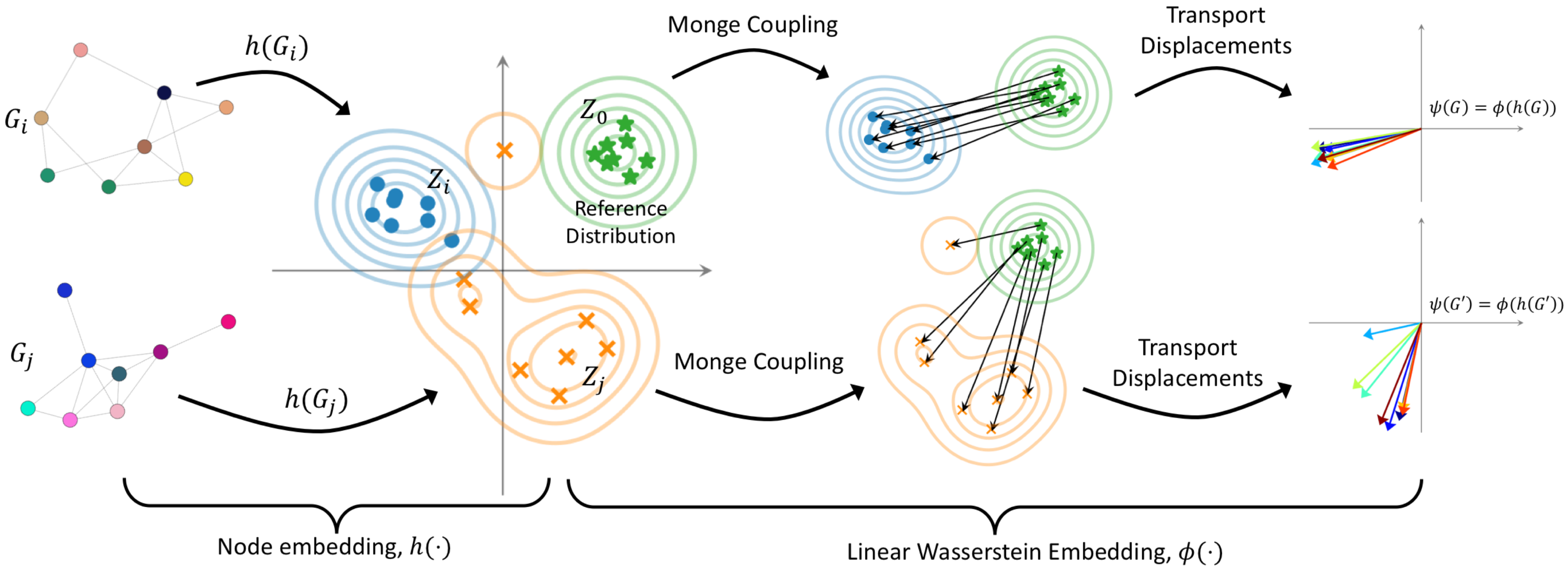
  1. $\phi(\mu_0) = 0$, reference distribution mapped to zero
  2. $\|\phi(\mu_i)\|_2 = \mathcal{W}_2(\mu_i, \mu_0)$: preserves distances to reference
  3. $\left\|\phi(\mu_i) - \phi(\mu_j)\right\|_2 \approx \mathcal{W}_2(\mu_i, \mu_j)$: The $l_2$ distance between $\phi(\mu_i)$ and $\phi(\mu_j)$ approximates $\mathcal{W}_2(\mu_i, \mu_j)$




$$\|\phi(\mu_i) - \phi(\mu_0)\|_2 = \|\phi(\mu_i)\|_2 = \mathcal{W}_2(\mu_i, \mu_0)$$

$$\|\phi(\mu_i) - \phi(\mu_j)\|_2 \approx \mathcal{W}_2(\mu_i, \mu_j)$$

# Wasserstein Embedding for Graph Learning (WEGL)




For a dataset of $M$ graphs, WEGL requires $M$ calculations of the Wasserstein distance for training, and 1 distance calculation for testing

# Experimental evaluation

OGB molhiv

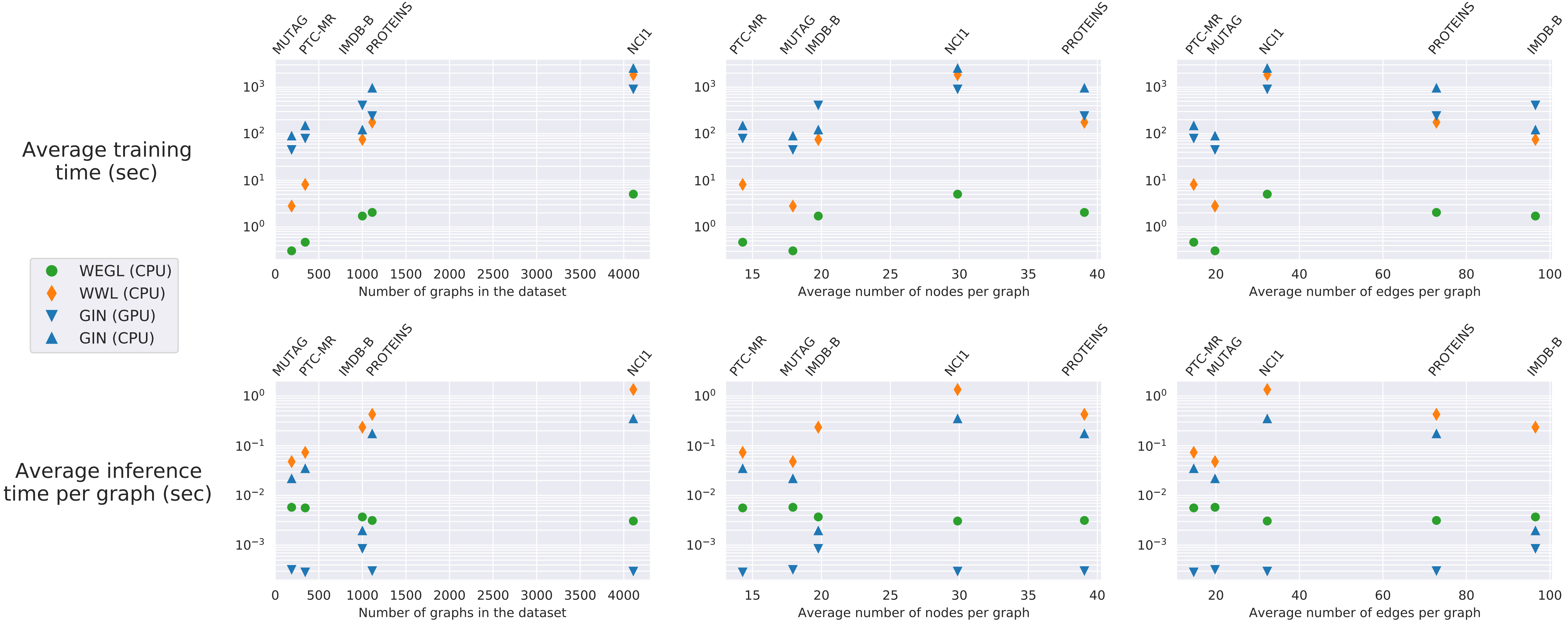
| | Method | Validation ROC-AUC (%) | Test ROC-AUC (%) |
|---|---|---|---|
| GNN | GCN (Kipf & Welling, 2016) | 83.8 ± 0.9 | 76.0 ± 1.2 |
| GNN | GIN + Virtual Node (Xu et al., 2019) | **84.8 ± 0.7** | 77.1 ± 1.5 |
| GNN | DeeperGCN (Li et al., 2020) | 84.3 ± 0.6 | 78.6 ± 1.2 |
| GNN | HIMP (Fey et al., 2020) | - | 78.8 ± 0.8 |
| GNN | GCN + GraphNorm (Cai et al., 2020) | 79.0 ± 1.1 | 78.8 ± 1.0 |
| Ours | WEGL + Random Forest | 79.2 ± 2.2 | 75.5 ± 1.5 |
| Ours | WEGL + Virtual Node + Random Forest | 81.9 ± 1.3 | 76.5 ± 1.8 |
| Ours | WEGL + Virtual Node + AutoML | 81.6 ± 0.6 | **79.1 ± 0.3** |
| | GAP + Virtual Node + Random Forest | 74.9 ± 2.4 | 72.1 ± 1.7 |

TUDataset

| | Method | IMDB-B | IMDB-M | COLLAB | RE-B | RE-M5K | PTC-MR | ENZYMES | PROTEINS |
|---|---|---|---|---|---|---|---|---|---|
| GNN | DGCNN (Zhang et al., 2018a) | 69.2±3.0 | 45.6±3.4 | 71.2±1.9 | 87.8±2.5 | 49.2±1.2 | 58.6 | 38.9±5.7 | 72.9±3.5 |
| GNN | GraphSAGE (Hamilton et al., 2017) | 68.8±4.5 | 47.6±3.5 | 73.9±1.7 | 84.3±1.9 | 50.0±1.3 | 63.9±7.7 | - | 75.9±3.2 |
| GNN | GIN (Xu et al., 2019) | 75.1±5.1 | 2.3±2.8 | 80.2±1.9 | **92.4±2.5** | **57.5±1.5** | 64.6±7.0 | - | 76.2±2.8 |
| GNN | GNTK (Du et al., 2019) | **76.9±3.6** | **52.8±4.6** | **83.6±1.0** | - | - | **67.9±6.9** | - | 75.6±4.2 |
| GNN | CapsGNN (Xinyi & Chen, 2019) | 73.1±4.8 | 50.3±2.6 | 79.6±0.9 | - | 52.9±1.5 | - | 54.7±5.7 | **76.3±3.6** |
| GNN | GraphNorm (Cai et al., 2020) | **76.0±3.7** | - | **80.2±1.0** | **93.5±2.1** | - | 64.9±7.5 | - | **77.4±4.9** |
| GK | DGK (Yanardag & Vishwanathan, 2015) | 67.0±0.6 | 44.6±0.5 | 73.1±0.3 | 78.0±0.4 | 41.3±0.2 | 57.3±1.1 | 27.1±0.8 | 71.7±0.5 |
| GK | WL (Shervashidze et al., 2011) | 73.8±3.9 | 49.8±0.5 | 74.8±0.2 | 68.2±0.2 | 51.2±0.3 | 57.0±2.0 | 53.2±1.1 | 72.9±0.6 |
| GK | RetGK (Zhang et al., 2018b) | 71.0±0.6 | 46.7±0.6 | 73.6±0.3 | 90.8±0.2 | 54.2±0.3 | **67.9±1.4** | 59.1±1.1 | 75.2±0.3 |
| GK | AWE (Ivanov & Burnaev, 2018) | 74.5±5.8 | 51.5±3.6 | 73.9±1.9 | 87.9±2.5 | 50.5±1.9 | - | 35.8±5.9 | - |
| GK | WWL (Togninalli et al., 2019) | 74.4±0.8 | - | - | - | - | 66.3±1.2 | **59.1±0.8** | 74.3±0.6 |
| Ours | WEGL + SVM-RBF | 73.4±2.5 | 51.7±3.1 | 78.6±1.0 | 92.1±1.9 | **56.1±2.3** | 63.4±5.3 | 57.3±4.2 | 76.0±4.4 |
| Ours | WEGL + Random Forest | **75.4±5.0** | **52.0±4.1** | 79.8±1.5 | 92.0±0.8 | 55.1±2.5 | **67.5±7.7** | **60.5±5.9** | **76.5±4.2** |
| Ours | WEGL + GBDT | 75.2±5.0 | **52.3±2.9** | **80.6±2.0** | **92.9±1.9** | **55.4±1.6** | 66.2±6.9 | **60.0±6.3** | 76.3±3.9 |

# Average wall-clock time comparison

## 100X faster training and 10-100X faster testing (on CPU)



# Our code is available!

GitHub https://github.com/navid-naderi/WEGL