中国人民大學
RENMIN UNIVERSITY OF CHINA

MGNN: Graph Neural Networks Inspired by Distance Geometry Problem

Guanyu Cui [1]   Zhewei Wei* [1]

[1]Renmin University of China

KDD2023 AUGUST 6 - 10 LONG BEACH, CA

Graph Neural Networks (GNNs)

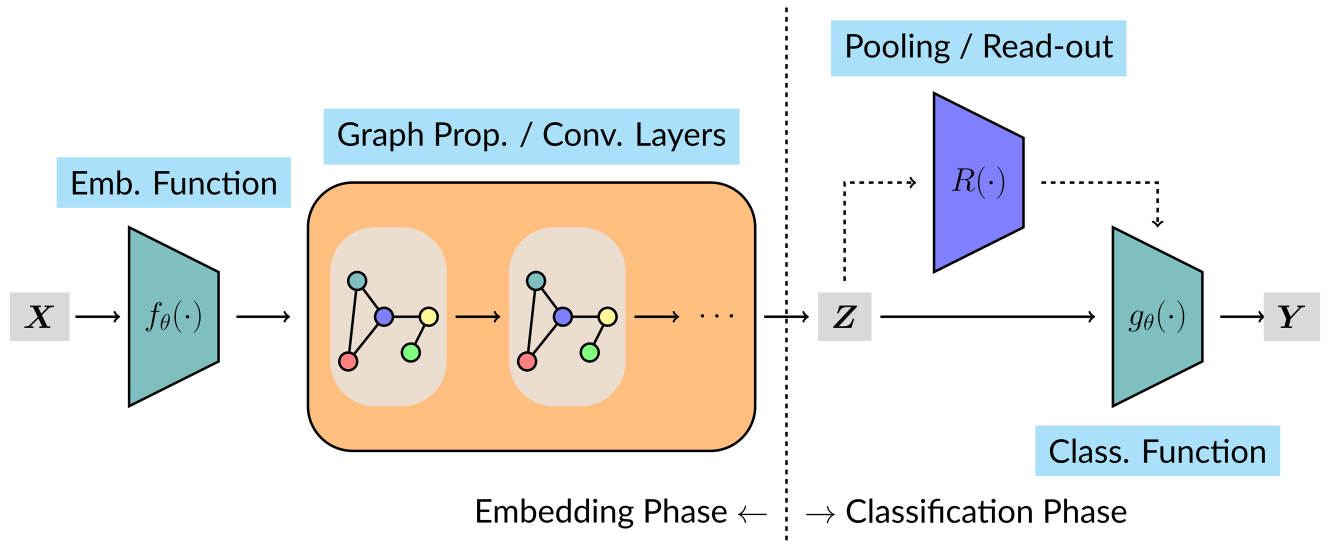


Figure 1. A typical structure of GNNs.

- Embedding function $f(\boldsymbol{X})$: map features $\boldsymbol{X} \in \mathbb{R}^{n\times F}$ to initial embeddings $\boldsymbol{Z}^{(0)} \in \mathbb{R}^{n\times d}$;
- Graph propagation / convolution $\mathrm{GP}(\boldsymbol{Z}^{(0)}; \boldsymbol{A})$: propagate embedding K times;
- Read-out / pooling $\mathrm{R}(\boldsymbol{Z}^{(K)}; \boldsymbol{A})$ (optional): pooling for graph-level tasks;
- Classification function $g(\cdot)$: final classification to generate predictions $\boldsymbol{Y}$.

Expressive Power & Universality of GNNs

- Spectral GNNs: designing **universal filters** (e.g., ChebNet, GPRGNN, BernNet, ...).
- Spatial GNNs: designing GNNs bounded by k**-WL tests** (e.g., GIN, ...).

There are works exploring relations between GNNs and geometric objects (e.g., curvature, cellular sheaves) and physical concepts (e.g., oscillators).

Q: Can we define universality of spatial GNNs from a geometric perspective?

Preliminaries

Definition (Equivalent). For a given graph $G = (V, E)$, two node embedding matrices $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)} \in \mathbb{R}^{n\times d}$ are equivalent if for all $(i,j) \in E$, $\|\boldsymbol{Z}^{(1)}_{i:} - \boldsymbol{Z}^{(1)}_{j:}\|_2 = \|\boldsymbol{Z}^{(2)}_{i:} - \boldsymbol{Z}^{(2)}_{j:}\|_2$ holds.

Definition (Congruent). For a given graph $G = (V, E)$, two node embedding matrices $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)} \in \mathbb{R}^{n\times d}$ are congruent if for all $i, j \in V$, $\|\boldsymbol{Z}^{(1)}_{i:} - \boldsymbol{Z}^{(1)}_{j:}\|_2 = \|\boldsymbol{Z}^{(2)}_{i:} - \boldsymbol{Z}^{(2)}_{j:}\|_2$ holds.

Definition (Globally Rigid). For a given graph $G = (V, E)$, an embedding matrix $\boldsymbol{Z}$ is globally rigid if all its equivalent embedding matrices $\boldsymbol{Z}'$ are also congruent to $\boldsymbol{Z}$.

Definition (Rigid). For a given graph $G = (V, E)$, an embedding matrix $\boldsymbol{Z}$ is rigid if all equivalent embeddings that can be obtained by **continuous** motion from $\boldsymbol{Z}$ are congruent to $\boldsymbol{Z}$.

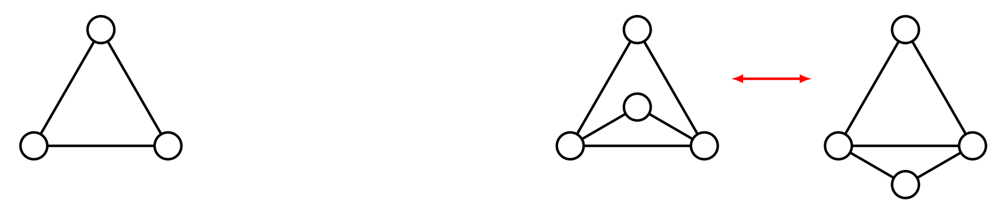
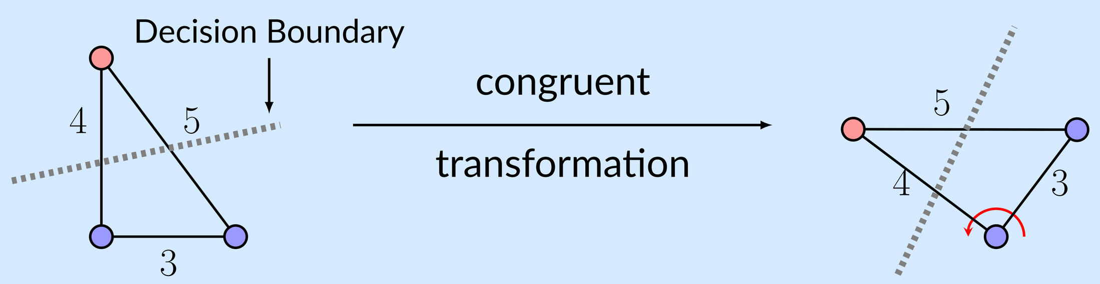


Figure 2. Left: A globally rigid graph in $\mathbb{R}^2$; Right: A rigid but not globally rigid graph in $\mathbb{R}^2$, since it has an equivalent but not congruent embedding.

Definition (Metric Matrix). The metric matrix of an embedding matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$ is defined as $\boldsymbol{M}_{\boldsymbol{Z}} = (\|\boldsymbol{Z}_{i:} - \boldsymbol{Z}_{j:}\|_2)_{ij}$. We also define the mapping from an embedding matrix $\boldsymbol{Z}$ to its metric matrix $\boldsymbol{M}_{\boldsymbol{Z}}$ as $\boldsymbol{M}_{\boldsymbol{Z}} = M(\boldsymbol{Z})$.

Defining Spatial-Universality

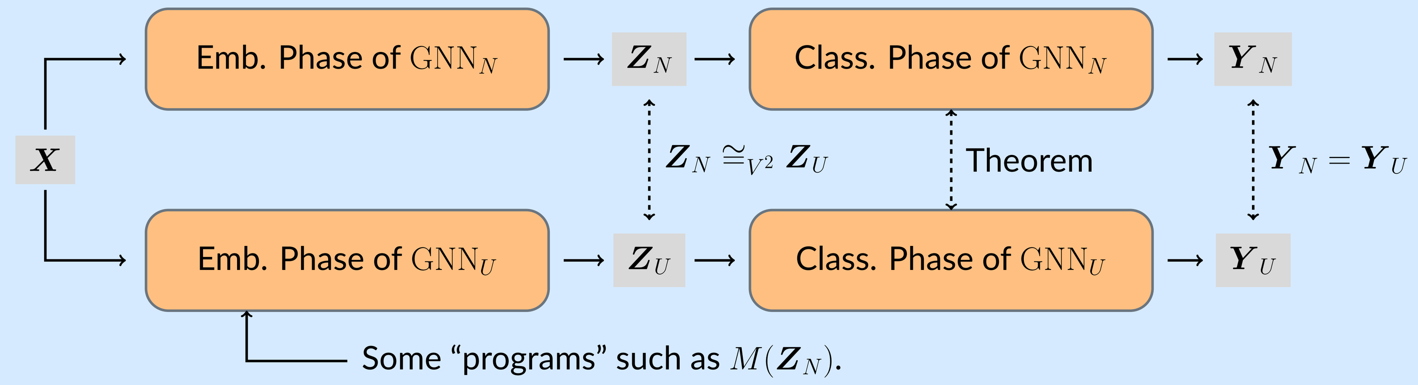
Observation:



Defining Spatial-Universality (Cont'd)

Theorem (MLPs Are Congruent-Insensitive). Given two congruent embedding matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, for any MLP_M (with biases), there always exists another MLP_N (also with biases) such that $\mathrm{MLP}_M(\boldsymbol{Z}_1) = \mathrm{MLP}_N(\boldsymbol{Z}_2)$.

Idea:



Spatial-Universal: It can arrange nodes with a given metric matrix!

This idea is closely related to the **Distance Geometry Problem (DGP)**.

Distance Geometry Problem (DGP)

Given a positive integer d, a graph $G = (V, E)$, and a symmetric non-negative matrix $\boldsymbol{M}$, decide whether there exists an embedding matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$, such that

$$\forall (i,j) \in E, \|\boldsymbol{Z}_{i:} - \boldsymbol{Z}_{j:}\| = \boldsymbol{M}_{ij}.$$

Optimization Objective

About the Optimization Objective

- **Full** metric matrix: $O(n^2) \Rightarrow$ **partial** metric matrix on edges: $O(m)$.
- For **globally rigid** graphs, partial metric matrix is enough to determine the "shape" of the embedding, this modification does not weaken the expressive power.
- However, solving the DGP is **NP-Hard**. We can not directly arrange the nodes, thus we introduce an **error-tolerant objective**:

$$E_p(\boldsymbol{Z}; \boldsymbol{M}, E) = \frac{1}{2}\|\boldsymbol{A} \odot (M(\boldsymbol{Z}) - \boldsymbol{M})\|_F^2 = \sum_{(i,j)\in E} \frac{1}{2}\left(\|\boldsymbol{Z}_{i:} - \boldsymbol{Z}_{j:}\|_2 - \boldsymbol{M}_{ij}\right)^2.$$

- This function is closely related to the raw **Stress function** σ_r in the **Multidimensional Scaling (MDS)** problem and the potential energy of **spring networks**.

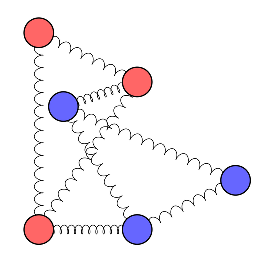


Figure 3. A spring network.

- To align with other representative GNNs, we modify E_p and add a regularization term to get the final objective:

$$\begin{aligned}\mathcal{L}(\boldsymbol{Z}; \boldsymbol{Z}^{(0)}, \boldsymbol{M}, E) &= (1-\alpha)\tilde{E}_p(\boldsymbol{Z}; \boldsymbol{M}, E) + \alpha\|\boldsymbol{Z} - \boldsymbol{Z}^{(0)}\|_F^2 \\ &= (1-\alpha)E_p(\boldsymbol{D}^{1/2}\boldsymbol{Z}; \boldsymbol{M}, E) + \alpha\|\boldsymbol{Z} - \boldsymbol{Z}^{(0)}\|_F^2.\end{aligned}$$

About the Metric Matrix

- For scenarios with prior knowledge about distances between nodes (e.g., molecular conformation generation, or graph drawing), directly use them; for other scenarios, learn a metric matrix.
- Our idea is to increase the distances between dissimilar nodes and reduce the distances between similar nodes:
  1. Introduce edge attention $\alpha_{ij} \in [-1, 1]$, when $\alpha_{ij} \to 1 \Leftrightarrow i, j$ tend to belong to the same class, and when $\alpha_{ij} \to -1 \Leftrightarrow i, j$ tend to belong to different classes;
  2. Map the initial embedding matrix $\boldsymbol{Z}^{(0)}$ (defined later) to a hidden matrix $\boldsymbol{H}$;
  3. Use attention mechanisms, such as $\alpha_{ij} = \tanh\left(\boldsymbol{a}^\top[\boldsymbol{H}_{i:}^\top \| \boldsymbol{H}_{j:}^\top]\right)$ or $\alpha_{ij} = \tanh\left(\boldsymbol{H}_{i:}\boldsymbol{W}\boldsymbol{H}_{j:}^\top\right)$ to learn the edge attention;
  4. Then we can set $\boldsymbol{M}_{ij} = \frac{1-\alpha_{ij}}{1+\alpha_{ij}+\varepsilon}\|\boldsymbol{Z}^{(0)}_{i:} - \boldsymbol{Z}^{(0)}_{j:}\|$, where ε is a small positive number.

Framework

The Embedding Function

- A linear layer $f(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{1}\boldsymbol{b}^\top$ in linear GNNs, or
- A two-layer MLP $f(\boldsymbol{X}) = \sigma(\sigma(\boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{1}\boldsymbol{b}_1^\top)\boldsymbol{W}_2 + \boldsymbol{1}\boldsymbol{b}_2^\top)$ in spectral GNNs.

Propagation

- Our goal is to design a propagation method that minimizes the objective.
- It's non-convex. Following related works, we employ the **stationary point iteration** method.
- By computing the gradient, setting it to zero, rearranging the terms, rewriting it as an iteration form, substituting $1-\alpha$ with β to allow more flexibility, it leads to:

$$\boldsymbol{Z}^{(k+1)} = (1-\alpha)\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{Z}^{(k)} + \beta\boldsymbol{D}^{-1/2}\boldsymbol{L}_{\boldsymbol{H}}\boldsymbol{D}^{-1/2}\boldsymbol{Z}^{(k)} + \alpha\boldsymbol{Z}^{(0)},$$

  where $\boldsymbol{H} = \boldsymbol{A} \odot \boldsymbol{M} \odot M(\boldsymbol{D}^{-1/2}\boldsymbol{Z})^{\odot -1}$, and $\boldsymbol{L}_{\boldsymbol{H}} = \mathrm{diag}(\boldsymbol{H}\boldsymbol{1}) - \boldsymbol{H}$.
- And we have the message-passing form:

$$\boldsymbol{Z}^{(k+1)}_{i:} = (1-\alpha)\sum_{j\in\mathcal{N}(i)} \frac{\boldsymbol{Z}^{(k)}_{i:}}{\sqrt{d_i d_j}} + \beta\sum_{j\in\mathcal{N}(i)} \frac{\boldsymbol{M}_{ij}\left(\boldsymbol{Z}^{(k)}_{i:} - \boldsymbol{Z}^{(k)}_{j:}\right)}{\sqrt{d_i d_j}\left\|\boldsymbol{Z}^{(k)}_{i:}/\sqrt{d_i} - \boldsymbol{Z}^{(k)}_{j:}/\sqrt{d_j}\right\|_2} + \alpha\boldsymbol{Z}^{(0)}_{i:}.$$

Optional Linear and Non-linear Transformations

We may incorporate linear and non-linear transformations after each propagation step. In our experiments without pre-designed metric matrices, such as node classification, we utilize three designs from the GCNII model: a linear transformation, the identity mapping, and a non-linear transformation (ReLU).

The Classification Function

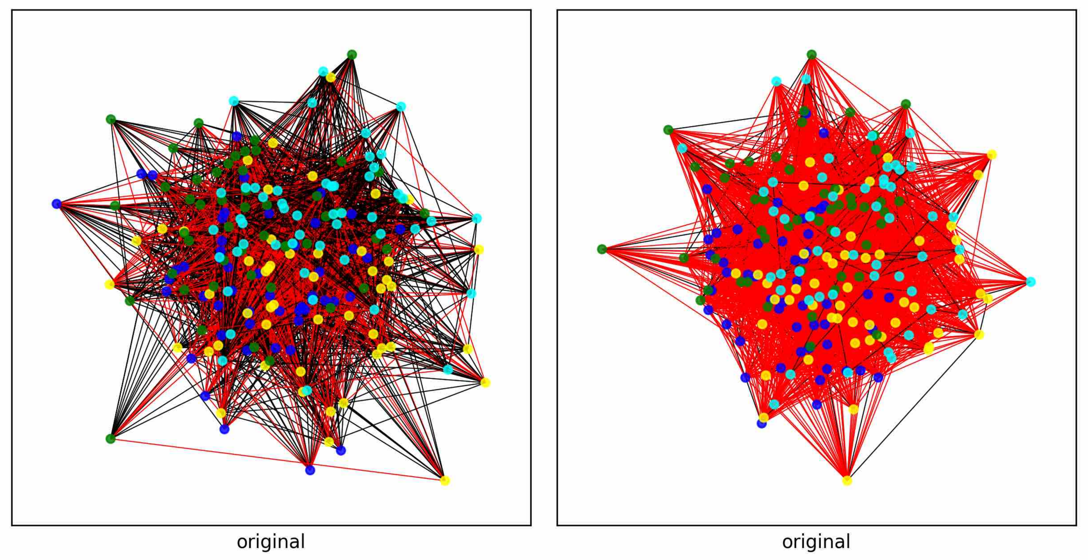
We use a linear layer $g(\boldsymbol{Z}^{(K)}) = \boldsymbol{Z}^{(K)}\boldsymbol{W} + \boldsymbol{1}\boldsymbol{b}^\top$ as the final classification function.

Experiments

We have done the "Arranging Nodes with Given Metric Matrices" experiments on synthetic graphs, supervised node classification and graph regression experiments on real-world graphs.

Arranging Nodes with Given Metric Matrices

- We generate two **Stochastic Block Model (SBM)** graphs, one homophilic and one heterophilic, consisting of four blocks with 50 nodes in each block.
- The node features are sampled from two 2-dimensional Gaussian distributions.



- If i and j are in the same class, we set $\boldsymbol{M}_{ij} = 0$; otherwise, we set $\boldsymbol{M}_{ij} = 5$.
- We pass the node features through 8 MGNN layers, with $\alpha = 0.05$ and $\beta = 0.5$.
- For visualization results, please refer to the smaller posters below or our paper.

Supervised Node Classification and Graph Regression

- Our MGNN model performs well, and the results are promising.
- For experiment details and results, please refer to the smaller posters below or our paper.

*Zhewei Wei is the corresponding author. Contact us: cuiguanyu@ruc.edu.cn; zhewei@ruc.edu.cn