Graph 6. Other Types of GNNs

- Graph Isomorphism Network (GIN) (ICLR 2019)
- GNNs with Joint Embedding of Nodes & Edges (WWW 2021, CP 2021)

1

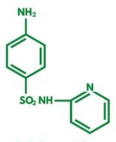
Outline

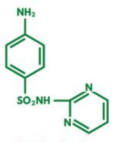
- **Graph Isomorphism Network (GIN) for Graph Classification**
  - Keyulu Xu, Weihua Hu, Jure Leskovec and Stefanie Jegelka. *How Powerful are Graph Neural Networks?,* ICLR 2019
- **GNNs with edge embedding** (in addition to node embedding)
  - Donghan Yu, Yiming Yang, Ruohong Zhang, Yuexin Wu. *Knowledge Embedding Based Graph Convolutional Networks*. WWW 2021.
  - Chaitanya K. Joshi, Quentin Cappart, Louis-Martin Rousseau, Thomas Laurent. *Learning the Travelling Salesperson Problem Requires Rethinking Generalization.* Constraint Programming 2021 https://arxiv.org/abs/2006.07054

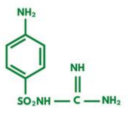
2

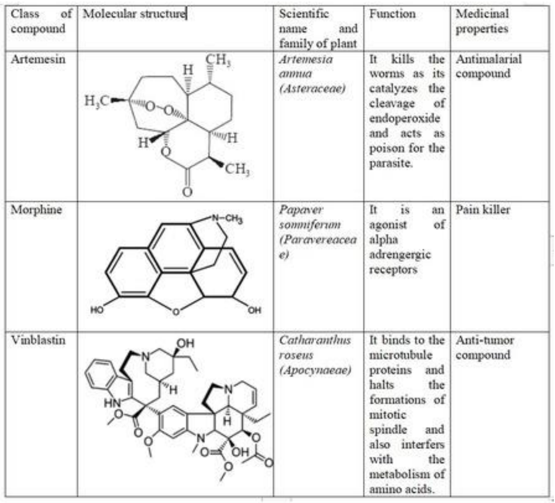
Examples of Graph Classification (Wikipedia)

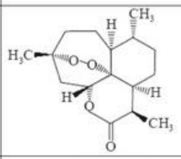
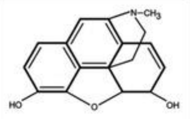
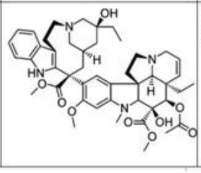
- Chemical Compound Classification


Sulphapyridine


Sulphadiazine

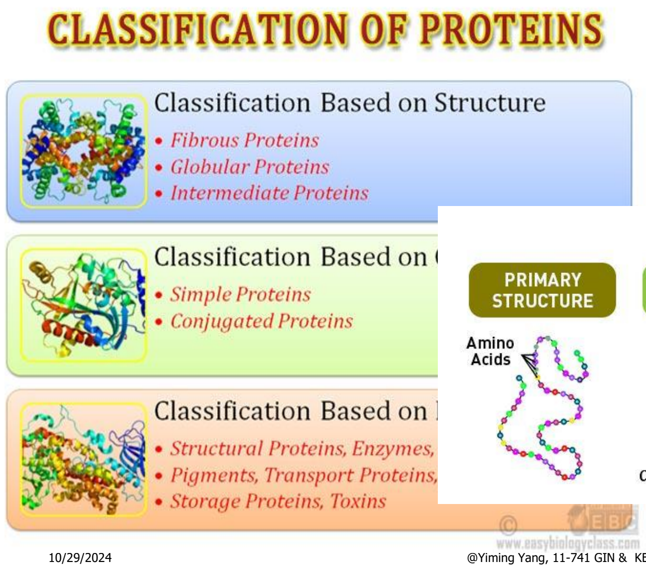

Sulphaguanidine

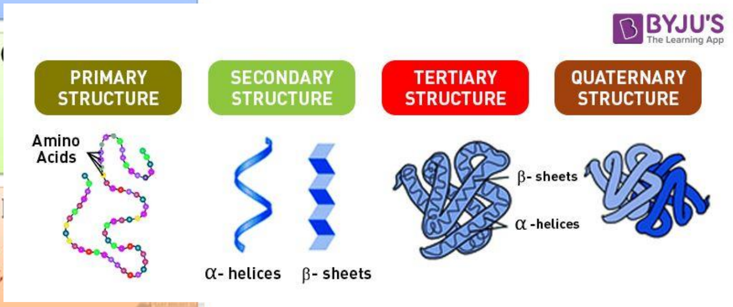


Class of compound	Molecular structure	Scientific name and family of plant	Function	Medicinal properties
Artemesin		*Artemesia annua (Asteraceae)*	It kills the worms as its catalyzes the cleavage of endoperoxide and acts as poison for the parasite.	Antimalarial compound
Morphine		*Papaver somniferum (Paravereacea e)*	It is an agonist of alpha adrengergic receptors	Pain killer
Vinblastin		*Catharanthus roseus (Apocynaeae)*	It binds to the microtubule proteins and halts the formations of mitotic spindle and also interfers with the metabolism of amino acids.	Anti-tumor compound

3

(from Wikipedia)





4

"How powerful are graph neural networks?"

(K. Xu et al., ICLR 2019; cited by 8774 on 10/24/2024)

Main Idea and Contributions

- Analyzing GNN's discriminative power in comparison with the WL test for graph isomorphism;
- Showing that popular GNNs (including GCN by Kipf & Welling, 2017) and GraphSage by Hamilton et al., 2017) cannot distinguish some different graph structures (with "mean" or "max pooling" operations);
- Proposed GIN (**G**raph **I**somorphism **N**etwork), a simple network which is equally powerful as the WL test.
- Achieved SOTA performance on graph classification benchmarks.

5

Graph Isomorphism Networks (GIN) (K. Xu et al., ICLR 2019)

- **Intermediate layer**: "combine" and "aggregate"

$$h_v^{(k)} = MLP^{(k)}\left((1+\varepsilon^{(k)})\ h_v^{(k-1)} + \sum_{u\in N(v)} h_u^{(k-1)}\right)$$

where $\varepsilon^{(k)}$ can be either a learnable or fixed scaler.

- **Final layer**: concatenate the multi-layer "readouts"

$$h_G = CONCAT\left(READOUT(\{h_v^{(k)} | v \in G\} | k = 0,1,\dots,K\right)$$

where the "$readout$" at each layer can be the summation of node embeddings at the same layer, or a more complicated aggregation function.

6

Graph Classification Task

- Input graphs: $\{G_1, G_2, \cdots, G_N\} \in \mathcal{G}$
- Output: $\{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N\} \in \{0,1\}^K$, predicted class labels
- Representation Learning: Use a labeled training set to learn the embeddings of nodes and aggregate the node embeddings into the embedding (a vector) of the entire graph.
- Classification Module (final step): Feed in the graph embedding to a classifier, e.g., LIB-SVM.

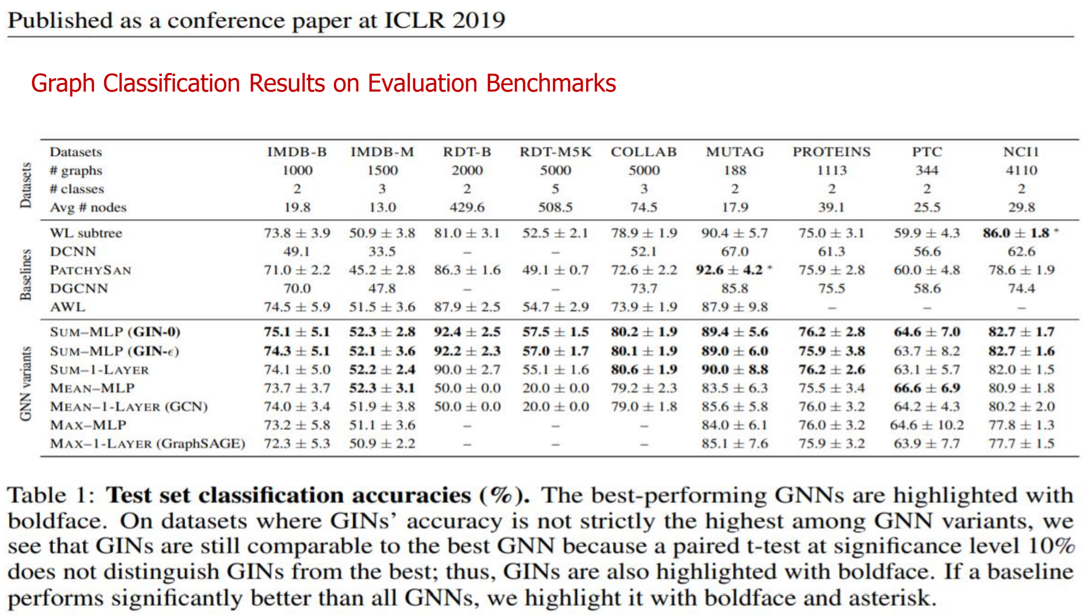
Weisfeiler-Lehman (WL) Graph Isomorphism Test (1968)

- **Graph isomorphism problem** asks whether two graphs are topologically identical. No polynomial-time algorithm is known for it yet.
- **Weisfeiler-Lehman test** offers an effective and computationally efficient test that distinguishes a broad class of graphs. Its 1-dimensional form is analogous to neighbor aggregation in GNNs, which iteratively (1) aggregates the labels of nodes and their neighborhoods, and (2) hashes the aggregated labels into unique new labels.
- The algorithm decides that two graphs are non-isomorphic if at some iteration the labels of the nodes between the two graphs differ.

Graph Classification Evaluation Benchmarks
[Deep graph kernels. KDD 2015]

- MUTAG is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds [7] with 7 discrete labels.
- PTC [31] is a dataset of 344 chemical compounds that reports the carcinogenicity for male and female rats, and it has 19 discrete labels.
- NCI1 and NCI109 [34] datasets (4100 and 4127 nodes, respectively), made publicly available by the National Cancer Institute (NCI) are two subsets of balanced datasets of chemical compounds screened for ability to suppress or inhibit the growth of a panel of human tumor cell lines, having 37 and 38 discrete labels respectively.
- ENZYMES is a balanced dataset of 600 protein tertiary structures obtained from [4] and has 3 discrete labels.
- PROTEINS is a dataset obtained from [4] where nodes are secondary structure elements (SSEs) and there is an edge between two nodes if they are neighbors in the amino-acid sequence or in 3D space. It has 3 discrete labels, representing helix, sheet or turn.

9



Published as a conference paper at ICLR 2019

Graph Classification Results on Evaluation Benchmarks

	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
Baselines	WL subtree	73.8 ± 3.9	50.9 ± 3.8	81.0 ± 3.1	52.5 ± 2.1	78.9 ± 1.9	90.4 ± 5.7	75.0 ± 3.1	59.9 ± 4.3	**86.0 ± 1.8** *
	DCNN	49.1	33.5	–	–	52.1	67.0	61.3	56.6	62.6
	PATCHYSAN	71.0 ± 2.2	45.2 ± 2.8	86.3 ± 1.6	49.1 ± 0.7	72.6 ± 2.2	**92.6 ± 4.2** *	75.9 ± 2.8	60.0 ± 4.8	78.6 ± 1.9
	DGCNN	70.0	47.8	–	–	73.7	85.8	75.5	58.6	74.4
	AWL	74.5 ± 5.9	51.5 ± 3.6	87.9 ± 2.5	54.7 ± 2.9	73.9 ± 1.9	87.9 ± 9.8	–	–	–
GNN variants	SUM–MLP (**GIN-0**)	**75.1 ± 5.1**	**52.3 ± 2.8**	**92.4 ± 2.5**	**57.5 ± 1.5**	**80.2 ± 1.9**	**89.4 ± 5.6**	**76.2 ± 2.8**	**64.6 ± 7.0**	**82.7 ± 1.7**
	SUM–MLP (**GIN-ϵ**)	**74.3 ± 5.1**	**52.1 ± 3.6**	**92.2 ± 2.3**	**57.0 ± 1.7**	**80.1 ± 1.9**	**89.0 ± 6.0**	**75.9 ± 3.8**	63.7 ± 8.2	**82.7 ± 1.6**
	SUM–1-LAYER	74.1 ± 5.0	**52.2 ± 2.4**	90.0 ± 2.7	55.1 ± 1.6	**80.6 ± 1.9**	**90.0 ± 8.8**	**76.2 ± 2.6**	63.1 ± 5.7	82.0 ± 1.5
	MEAN–MLP	73.7 ± 3.7	**52.3 ± 3.1**	50.0 ± 0.0	20.0 ± 0.0	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	**66.6 ± 6.9**	80.9 ± 1.8
	MEAN–1-LAYER (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	MAX–MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	MAX–1-LAYER (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

Table 1: **Test set classification accuracies (%).** The best-performing GNNs are highlighted with boldface. On datasets where GINs' accuracy is not strictly the highest among GNN variants, we see that GINs are still comparable to the best GNN because a paired t-test at significance level 10% does not distinguish GINs from the best; thus, GINs are also highlighted with boldface. If a baseline performs significantly better than all GNNs, we highlight it with boldface and asterisk.

10

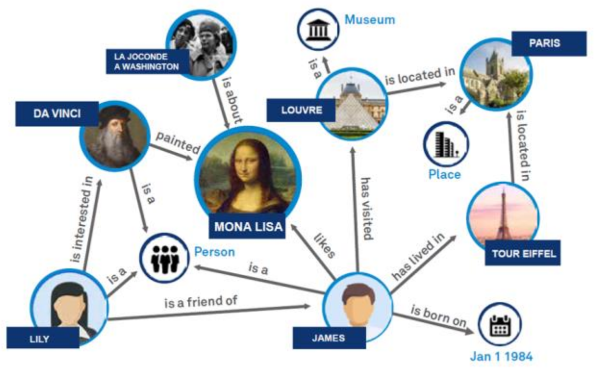
Outline

- **Graph Isomorphism Network** (GIN) for Graph Classification
  - Keyulu Xu, Weihua Hu, Jure Leskovec and Stefanie Jegelka. *How Powerful are Graph Neural Networks?,* ICLR 2019
- **GNNs with edge embedding** (in addition to node embedding)
  - Donghan Yu, Yiming Yang, Ruohong Zhang, Yuexin Wu. *Knowledge Embedding Based Graph Convolutional Networks*. WWW 2021.
  - Chaitanya K. Joshi, Quentin Cappart, Louis-Martin Rousseau, Thomas Laurent. *Learning the Travelling Salesperson Problem Requires Rethinking Generalization.* Constraint Programming 2021 https://arxiv.org/abs/2006.07054

Why do we care about edge embedding?

- Common Assumption in GCN, GAT, GIN, etc.
  - Links are assumed to be homogeneous, i.e., if two nodes are connected, then propagate their features to each other during the process of node embedding.
- OK Case
  - If A is synonym of B and B is a synonym of C, then **A and C should have similar embeddings.**
- Problematic Case
  - If A is synonym of B and B is an antonym C, then **we should not** force A and C to have similar embeddings.

KG with Heterogeneous Entities & Relations



- Naïve belief aggregation over those links could be misleading.
- E.g., we may end up with similar profiles (embeddings) Da Vinci and Mona Lisa, which is rather silly.
- E.g., our system may predict Mona Lisa being born in 1984 because James is a direct neighbor.

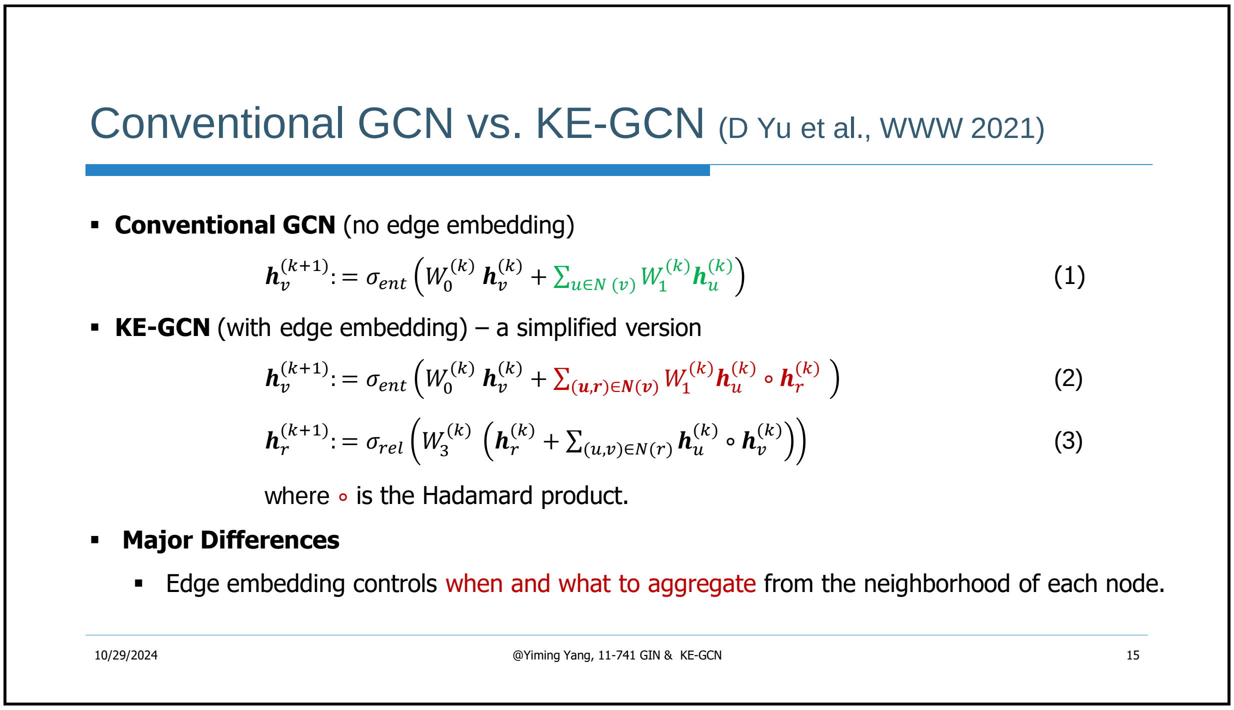
We must discriminate edge differences for effective belief propagation.

13

Remedy: Introducing relation embedding in GNNs

- Key Idea
  - Using edge embedding to automatically control which features should be propagated across nodes (e.g., Da Vinch and Mona Lisa are related as the painter and the paint) and which features should not (e.g., Da Vinch is a man, but Mona Lisa is not).
- KE GCN: KG-based Node/Edge Embedding (D Yu et al., WWW 2021)
- Anisotropic Graph Neural Networks (Z Sun et al, NeurIPS 2022)

14



Conventional GCN vs. KE-GCN (D Yu et al., WWW 2021)

- **Conventional GCN** (no edge embedding)

$$\boldsymbol{h}_v^{(k+1)} := \sigma_{ent}\left(W_0^{(k)}\,\boldsymbol{h}_v^{(k)} + \sum\nolimits_{u\in N\,(v)} W_1^{(k)}\boldsymbol{h}_u^{(k)}\right) \quad (1)$$

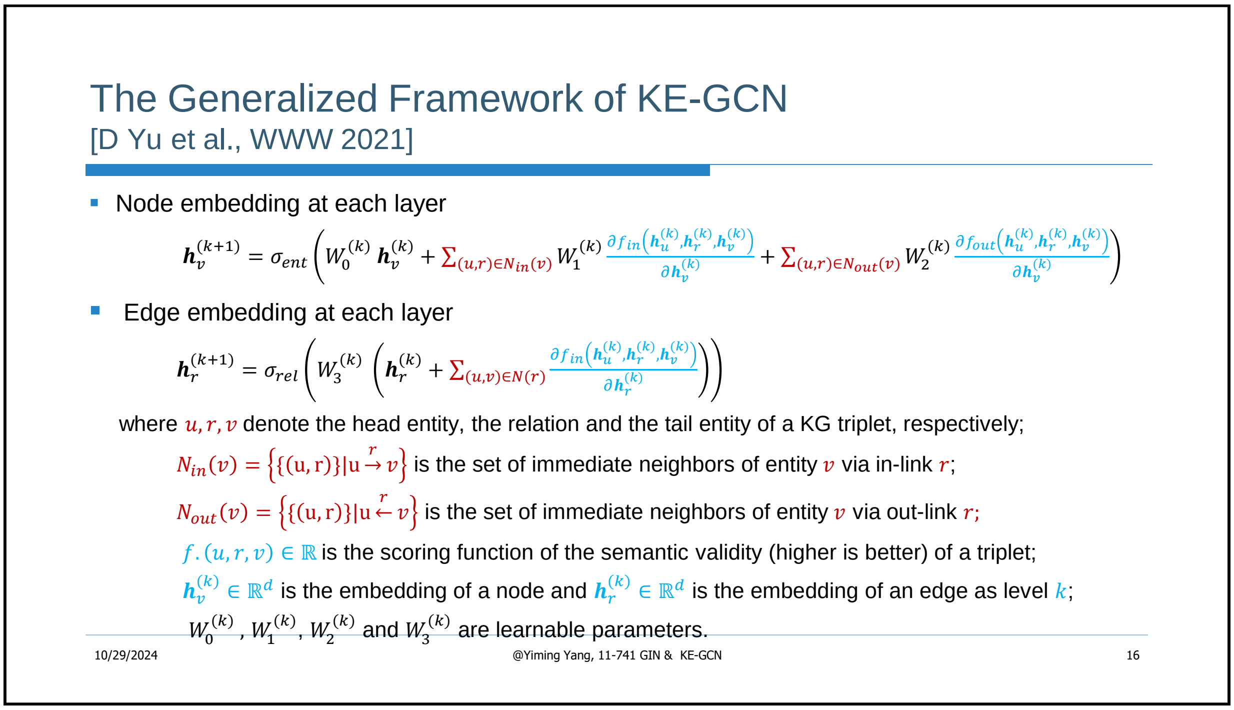
- **KE-GCN** (with edge embedding) – a simplified version

$$\boldsymbol{h}_v^{(k+1)} := \sigma_{ent}\left(W_0^{(k)}\,\boldsymbol{h}_v^{(k)} + \sum\nolimits_{(\boldsymbol{u},\boldsymbol{r})\in N(\boldsymbol{v})} W_1^{(k)}\boldsymbol{h}_u^{(k)} \circ \boldsymbol{h}_r^{(k)}\right) \quad (2)$$

$$\boldsymbol{h}_r^{(k+1)} := \sigma_{rel}\left(W_3^{(k)}\left(\boldsymbol{h}_r^{(k)} + \sum\nolimits_{(u,v)\in N(r)} \boldsymbol{h}_u^{(k)} \circ \boldsymbol{h}_v^{(k)}\right)\right) \quad (3)$$

where $\circ$ is the Hadamard product.

- **Major Differences**
  - Edge embedding controls when and what to aggregate from the neighborhood of each node.



The Generalized Framework of KE-GCN [D Yu et al., WWW 2021]

- Node embedding at each layer

$$\boldsymbol{h}_v^{(k+1)} = \sigma_{ent}\left(W_0^{(k)}\,\boldsymbol{h}_v^{(k)} + \sum\nolimits_{(u,r)\in N_{in}(v)} W_1^{(k)}\frac{\partial f_{in}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_v^{(k)}} + \sum\nolimits_{(u,r)\in N_{out}(v)} W_2^{(k)}\frac{\partial f_{out}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_v^{(k)}}\right)$$

- Edge embedding at each layer

$$\boldsymbol{h}_r^{(k+1)} = \sigma_{rel}\left(W_3^{(k)}\left(\boldsymbol{h}_r^{(k)} + \sum\nolimits_{(u,v)\in N(r)} \frac{\partial f_{in}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_r^{(k)}}\right)\right)$$

where u, r, v denote the head entity, the relation and the tail entity of a KG triplet, respectively;

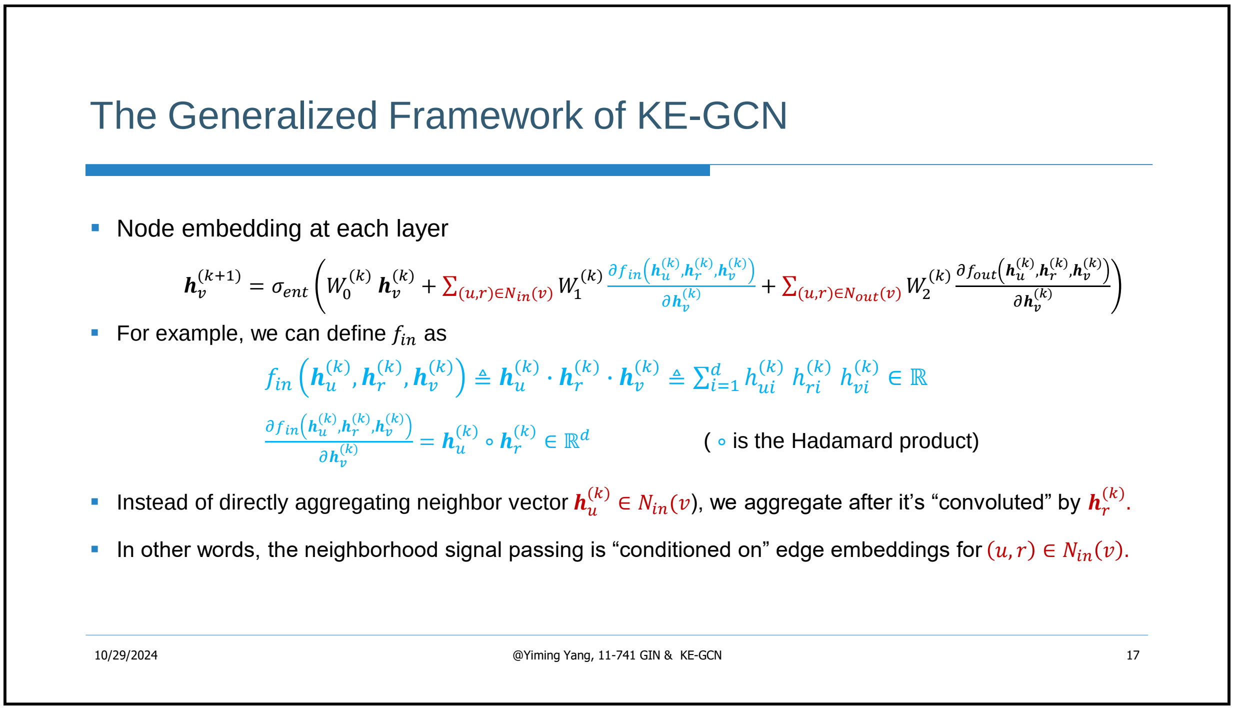
$N_{in}(v) = \left\{\{(\mathrm{u},\mathrm{r})\}|\mathrm{u} \overset{r}{\rightarrow} v\right\}$ is the set of immediate neighbors of entity v via in-link r;

$N_{out}(v) = \left\{\{(\mathrm{u},\mathrm{r})\}|\mathrm{u} \overset{r}{\leftarrow} v\right\}$ is the set of immediate neighbors of entity v via out-link r;

$f.(u, r, v) \in \mathbb{R}$ is the scoring function of the semantic validity (higher is better) of a triplet;

$\boldsymbol{h}_v^{(k)} \in \mathbb{R}^d$ is the embedding of a node and $\boldsymbol{h}_r^{(k)} \in \mathbb{R}^d$ is the embedding of an edge as level k;

$W_0^{(k)}$, $W_1^{(k)}$, $W_2^{(k)}$ and $W_3^{(k)}$ are learnable parameters.



The Generalized Framework of KE-GCN

- Node embedding at each layer

$$\boldsymbol{h}_v^{(k+1)} = \sigma_{ent}\left(W_0^{(k)}\,\boldsymbol{h}_v^{(k)} + \sum\nolimits_{(u,r)\in N_{in}(v)} W_1^{(k)}\,\frac{\partial f_{in}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_v^{(k)}} + \sum\nolimits_{(u,r)\in N_{out}(v)} W_2^{(k)}\,\frac{\partial f_{out}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_v^{(k)}}\right)$$

- For example, we can define f_{in} as

$$f_{in}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right) \triangleq \boldsymbol{h}_u^{(k)}\cdot\boldsymbol{h}_r^{(k)}\cdot\boldsymbol{h}_v^{(k)} \triangleq \sum\nolimits_{i=1}^{d} h_{ui}^{(k)}\,h_{ri}^{(k)}\,h_{vi}^{(k)} \in \mathbb{R}$$

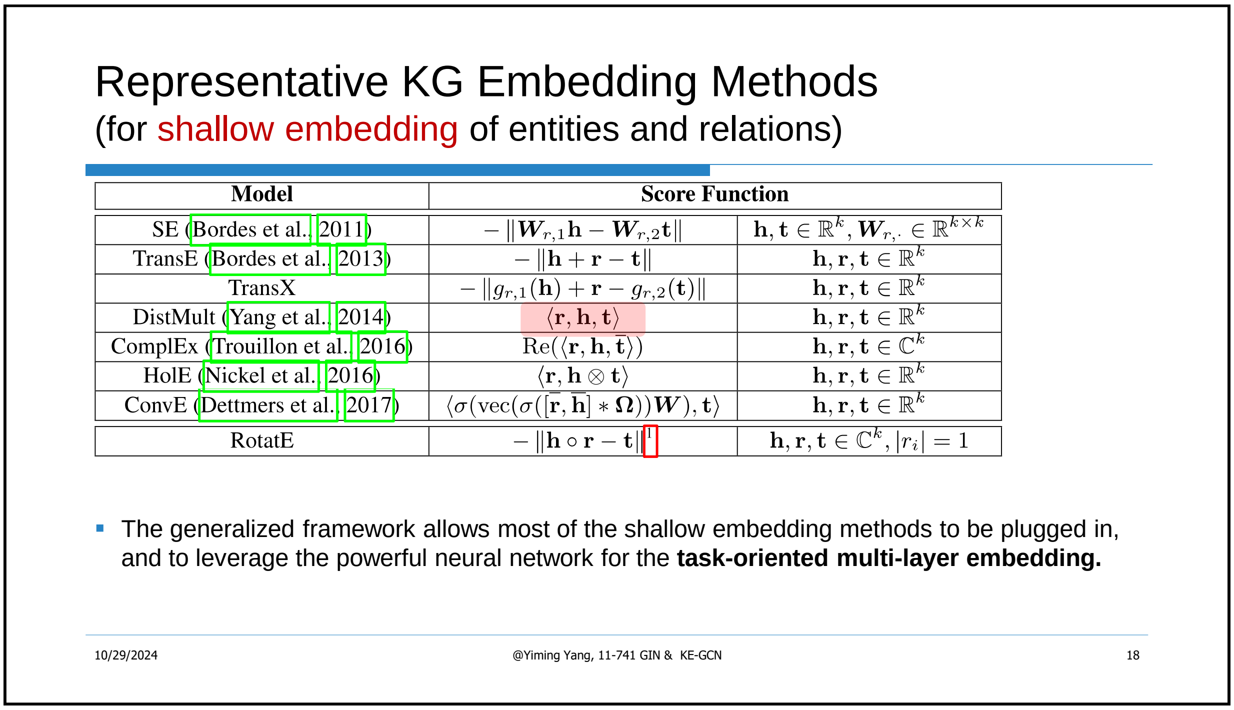
$$\frac{\partial f_{in}\left(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_r^{(k)},\boldsymbol{h}_v^{(k)}\right)}{\partial \boldsymbol{h}_v^{(k)}} = \boldsymbol{h}_u^{(k)}\circ\boldsymbol{h}_r^{(k)} \in \mathbb{R}^d$$

($\circ$ is the Hadamard product)

- Instead of directly aggregating neighbor vector $\boldsymbol{h}_u^{(k)} \in N_{in}(v)$, we aggregate after it's "convoluted" by $\boldsymbol{h}_r^{(k)}$.
- In other words, the neighborhood signal passing is "conditioned on" edge embeddings for $(u,r) \in N_{in}(v)$.

10/29/2024 @Yiming Yang, 11-741 GIN & KE-GCN 17

17



Representative KG Embedding Methods
(for shallow embedding of entities and relations)

Model	Score Function	
SE (Bordes et al., 2011)	$-\|\boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t}\|$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^k, \boldsymbol{W}_{r,\cdot}\in\mathbb{R}^{k\times k}$
TransE (Bordes et al., 2013)	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
TransX	$-\|g_{r,1}(\mathbf{h})+\mathbf{r}-g_{r,2}(\mathbf{t})\|$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
DistMult (Yang et al., 2014)	$\langle\mathbf{r},\mathbf{h},\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle\mathbf{r},\mathbf{h},\bar{\mathbf{t}}\rangle)$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^k$
HolE (Nickel et al., 2016)	$\langle\mathbf{r},\mathbf{h}\otimes\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
ConvE (Dettmers et al., 2017)	$\langle\sigma(\mathrm{vec}(\sigma([\bar{\mathbf{r}},\bar{\mathbf{h}}]*\boldsymbol{\Omega}))\boldsymbol{W}),\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
RotatE	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|^1$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^k, \|r_i\|=1$

- The generalized framework allows most of the shallow embedding methods to be plugged in, and to leverage the powerful neural network for the **task-oriented multi-layer embedding.**

10/29/2024 @Yiming Yang, 11-741 GIN & KE-GCN 18

18

KE GCN vs. Other Design Choices

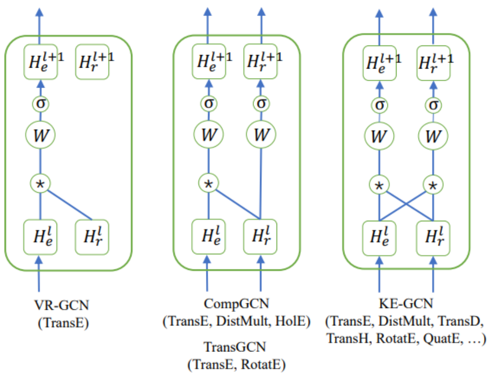


Figure 1: A simple realization of KE-GCN compared to previous works VR-GCN, TransGCN, and CompGCN. H_e^l and H_r^l means the entity (node) embedding and relation (edge) embedding at layer l respectively. $\star$ denotes the graph convolu-

KE-GCN for Cross-language KG Entity Alignment

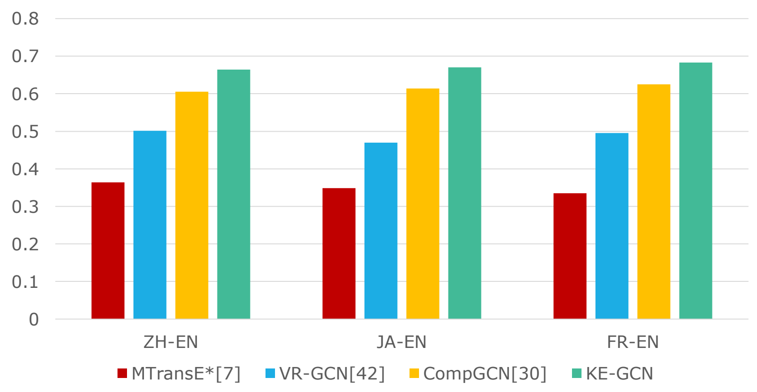
- **Task**
  - To align KG entities across English (EN), Japanese (JA), French (FR) and Chinese (ZH), e.g., Biden ←→ 拜登
- **Training Data per Language Pair**
  - Positive matching pairs $S = \{(u, v)\}$ for entity $u \in KG1$ and entity $v \in KG2$;
  - Negative matching pairs $S' = \{(u', v')\}$ for entity $u' \in KG1$ and entity $v' \in KG2$.
- **Loss Function**

$$\mathcal{L} = -\sum_{(u,v)\in S}\sum_{(u',v')\in S'}\left[\|h_u - h_v\|_1 - \|h_{u'} - h_{v'}\|_1 + \gamma\right]_+$$

where γ is a hyperparameter for large-margin separation in hinge loss $x = [.\}_+$.

- $\mathcal{L}$ makes the positive pairs have closer embeddings than those for the negative pairs..

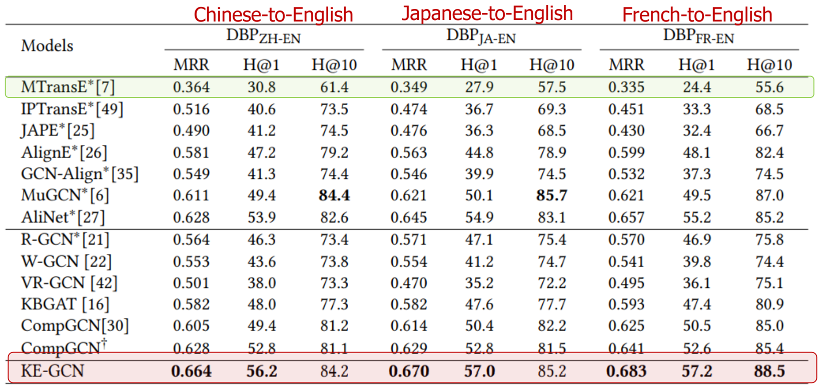
Cross-language KG Entity Alignment Results in MRR



21

Evaluation on KG Entity Alignment (across languages)

Table 2: Experiment results in knowledge graph entity alignment task on DBP15K datasets, where the average results over 5 different runs are reported. * indicate that results are directly taken from [27]. The results of VR-GCN [42] are directly taken from the original paper. CompGCN marked with † incorporates the composition operations in RotatE [24] and QuatE [47] while original CompGCN [30] only contains subtraction, multiplication and circular-correlation operations.



	Chinese-to-English			Japanese-to-English			French-to-English		
Models	$DBP_{ZH\text{-}EN}$			$DBP_{JA\text{-}EN}$			$DBP_{FR\text{-}EN}$		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
MTransE*[7]	0.364	30.8	61.4	0.349	27.9	57.5	0.335	24.4	55.6
IPTransE*[49]	0.516	40.6	73.5	0.474	36.7	69.3	0.451	33.3	68.5
JAPE*[25]	0.490	41.2	74.5	0.476	36.3	68.5	0.430	32.4	66.7
AlignE*[26]	0.581	47.2	79.2	0.563	44.8	78.9	0.599	48.1	82.4
GCN-Align*[35]	0.549	41.3	74.4	0.546	39.9	74.5	0.532	37.3	74.5
MuGCN*[6]	0.611	49.4	**84.4**	0.621	50.1	**85.7**	0.621	49.5	87.0
AliNet*[27]	0.628	53.9	82.6	0.645	54.9	83.1	0.657	55.2	85.2
R-GCN*[21]	0.564	46.3	73.4	0.571	47.1	75.4	0.570	46.9	75.8
W-GCN [22]	0.553	43.6	73.8	0.554	41.2	74.7	0.541	39.8	74.4
VR-GCN [42]	0.501	38.0	73.3	0.470	35.2	72.2	0.495	36.1	75.1
KBGAT [16]	0.582	48.0	77.3	0.582	47.6	77.7	0.593	47.4	80.9
CompGCN[30]	0.605	49.4	81.2	0.614	50.4	82.2	0.625	50.5	85.0
CompGCN†	0.628	52.8	81.1	0.629	52.8	81.5	0.641	52.6	85.4
KE-GCN	**0.664**	**56.2**	84.2	**0.670**	**57.0**	85.2	**0.683**	**57.2**	**88.5**

22

KE-GCN for Entity (Node) Classification

- Notation
  - $\mathcal{D}_l = \{(X_i, Y_i)\}$ is a labeled training set;
  - $Y_{ij} \in \{0,1\}$ indicates the true label of node i with respect to category $j \in \{1, \ldots, K\}$;
  - $\hat{Y}_{ij} \in \mathbb{R}$ is the system's output on node i with respect to category $j \in \{1, \ldots, K\}$.
- Cross Entropy Loss for Multi-class Classification (on the AM and WN datasets)

$$\mathcal{L} = -\sum_{(X_i, Y_i) \in \mathcal{D}_l} \sum_{j=1}^{K} Y_{ij} \ln \hat{Y}_{ij}$$

- Cross Entropy Loss for Multi-label Classification (on the FB15K dataset)

$$\mathcal{L} = -\sum_{(X_i, Y_i) \in \mathcal{D}_l} \sum_{j=1}^{K} \left[Y_{ij} \ln \hat{Y}_{ij} + \left(1 - Y_{ij}\right) \ln\left(1 - \hat{Y}_{ij}\right) \right]$$

Evaluation on KG Entity (Node) Classification

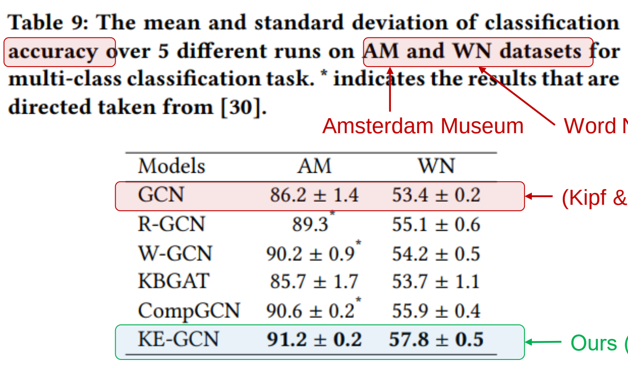


Table 9: The mean and standard deviation of classification accuracy over 5 different runs on AM and WN datasets for multi-class classification task. * indicates the results that are directed taken from [30].

Amsterdam Museum

Word Net

Models	AM	WN
GCN	86.2 ± 1.4	53.4 ± 0.2
R-GCN	89.3*	55.1 ± 0.6
W-GCN	90.2 ± 0.9*	54.2 ± 0.5
KBGAT	85.7 ± 1.7	53.7 ± 1.1
CompGCN	90.6 ± 0.2*	55.9 ± 0.4
KE-GCN	**91.2 ± 0.2**	**57.8 ± 0.5**

(Kipf & Welling, ICIR 2017): No edge embedding

Ours (WWW'21): with TransE scoring function

Concluding Remarks

- Using edges for neighbor aggregation without considering edge meanings would be suboptimal for graph-based learning.
- Jointly learning both node embedding and edge embedding is proven to be effective for knowledge-enhanced reasoning in prediction tasks.
- New opportunities for using GNNs for solving combinatorial optimization problems (later).

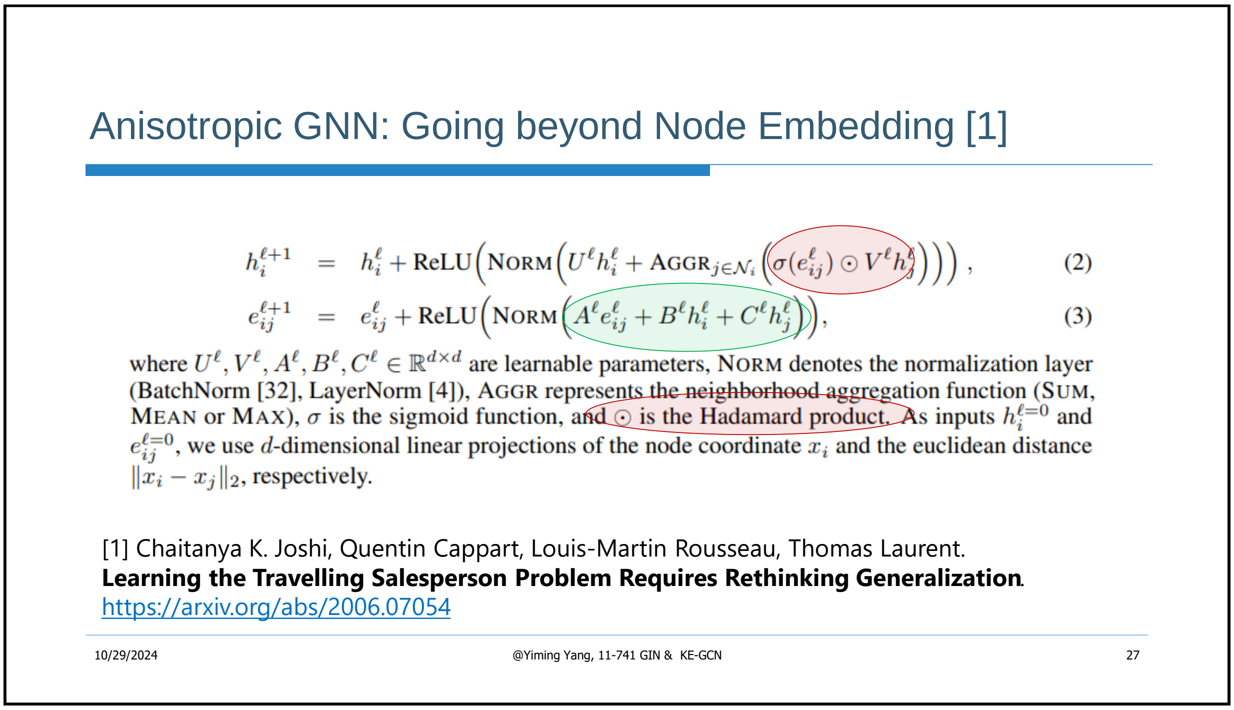
25

Traveling Salesman Problem (NP-Complete)

United States

For finding the optimal tour, edge information (traveling cost between two cities) is important.

26



Anisotropic GNN: Going beyond Node Embedding [1]

$$h_i^{\ell+1} = h_i^\ell + \text{ReLU}\Big(\text{Norm}\Big(U^\ell h_i^\ell + \text{Aggr}_{j\in\mathcal{N}_i}\Big(\sigma(e_{ij}^\ell) \odot V^\ell h_j^\ell\Big)\Big)\Big), \quad (2)$$

$$e_{ij}^{\ell+1} = e_{ij}^\ell + \text{ReLU}\Big(\text{Norm}\Big(A^\ell e_{ij}^\ell + B^\ell h_i^\ell + C^\ell h_j^\ell\Big)\Big), \quad (3)$$

where $U^\ell, V^\ell, A^\ell, B^\ell, C^\ell \in \mathbb{R}^{d\times d}$ are learnable parameters, Norm denotes the normalization layer (BatchNorm [32], LayerNorm [4]), Aggr represents the neighborhood aggregation function (Sum, Mean or Max), σ is the sigmoid function, and $\odot$ is the Hadamard product. As inputs $h_i^{\ell=0}$ and $e_{ij}^{\ell=0}$, we use d-dimensional linear projections of the node coordinate x_i and the euclidean distance $\|x_i - x_j\|_2$, respectively.

[1] Chaitanya K. Joshi, Quentin Cappart, Louis-Martin Rousseau, Thomas Laurent. **Learning the Travelling Salesperson Problem Requires Rethinking Generalization**. https://arxiv.org/abs/2006.07054

27