CS6208 : Advanced Topics in Artificial Intelligence
Graph Machine Learning

Lecture 8 : Benchmarking Graph Neural Networks and Graph Datasets

Semester 2 2022/23

Xavier Bresson

https://twitter.com/xbresson

Department of Computer Science
National University of Singapore (NUS)

Outline

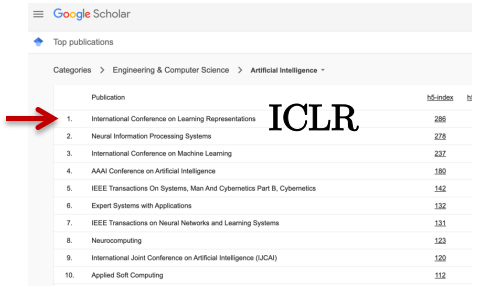
- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

Outline

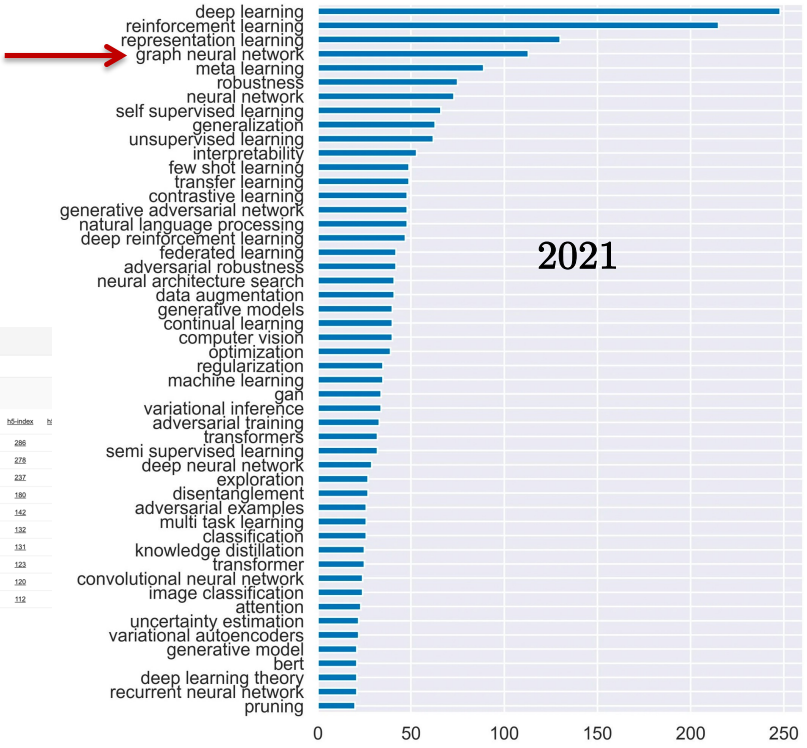
- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

ICLR statistics

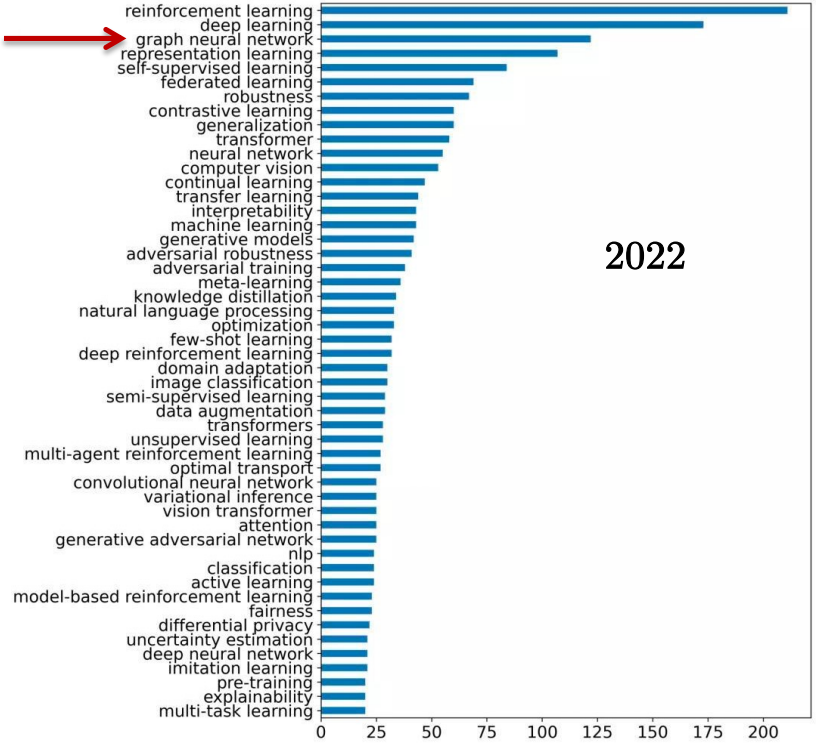
- GNNs have become the standard toolkit for analyzing and learning from data on graphs.
- One of the hottest machine learning topics since 2021.



ICLR



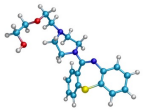
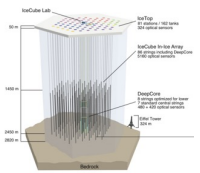
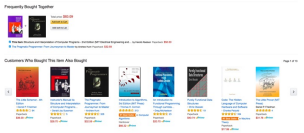
ICLR Submission Top 50 Keywords

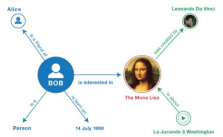
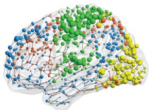


ICLR Submission Top 50 Keywords

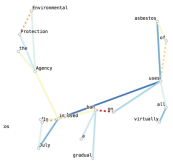
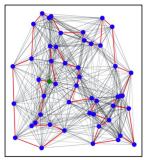
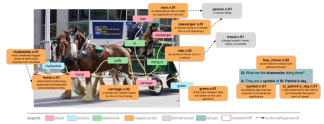
Application domains

- GNNs are flexible to adapt to complex data structure and combine multi-modal features.
- Applicative domains :
  - Chemistry[1,2] (generate new drugs and materials)
  - Physics[3,4,5] (detect particles, accelerate physics)
  - Recommender systems[6,7] (leverage consumer-product choices)
  - Social sciences[8,9] (predict future friends, identify fake news)
  - Knowledge graphs[10,11] (reasoning with entity-relationship)
  - Neuroscience[12] (understanding brain mechanisms and neuro-degenerative diseases)
  - Computer Vision[13] (scene understanding for visual reasoning)
  - Natural Language Processing[14,15] (common sense reasoning)
  - Combinatorial Optimization[16,17] (better/faster approximated solutions to NP-hard problems)

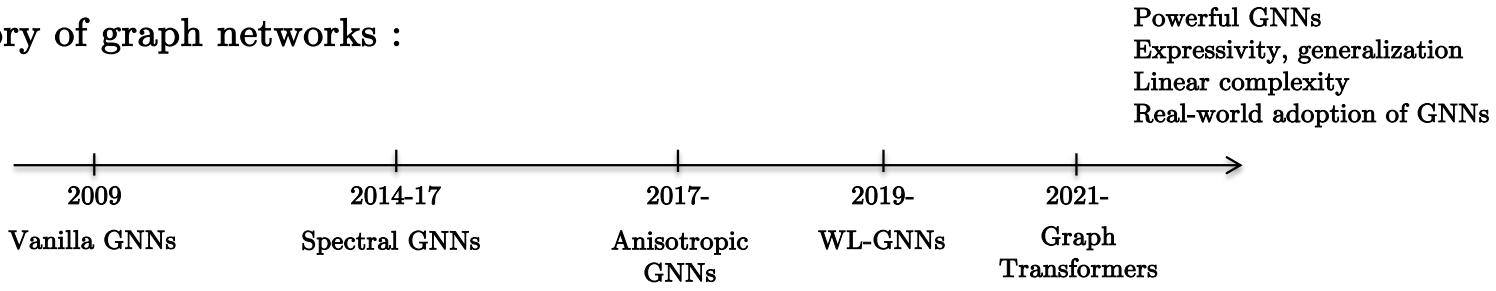



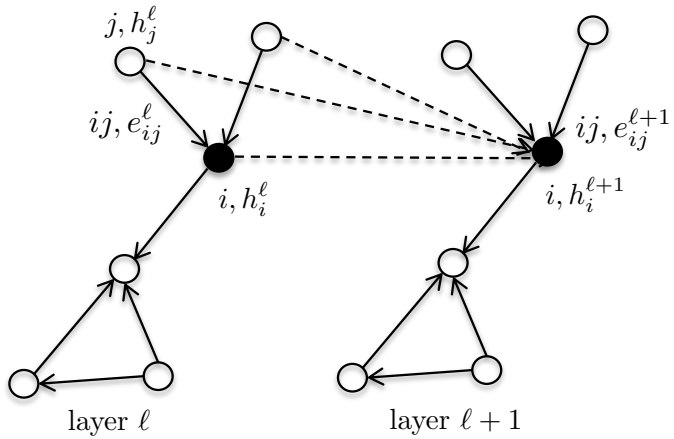
[1] Duvenaud, Maclaurin, Iparraguirre, Bombarell, Hirzel, Aspuru-Guzik, Adams, Convolutional networks on graphs for learning molecular fingerprints, 2015
[2] Gilmer, Schoenholz, Riley, Vinyals, Dahl, Neural message passing for quantum chemistry, 2017
[3] Battaglia, Pascanu, Lai, Rezende, Danilo, Interaction networks for learning about objects, relations and physics, 2016
[4] Cranmer, Xu, Battaglia, Ho, Learning symbolic physics with graph networks, 2019
[5] Sanchez-Gonzalez, Godwin, Pfaff, Ying, Leskovec, Battaglia, Learning to simulate complex physics with graph networks, 2020
[6] Monti, Bronstein, Bresson, Geometric matrix completion with recurrent multi-graph neural networks, 2017
[7] Ying, He, Chen, Eksombatchai, Hamilton, Leskovec, Graph convolutional neural networks for web-scale recommender systems, 2018
[8] Kipf, Welling, Semi-supervised classification with graph convolutional networks, 2017
[9] Monti, Frasca, Eynard, Mannion, Bronstein, Fake news detection on social media using geometric deep learning, 2019
[10] Schlichtkrull, Kipf, Bloem, VanDenBerg, Titov, Welling, Modeling relational data with graph convolutional networks, 2018
[11] Chami, Wolf, Juan Sala Ravi, Re, Low-dimensional hyperbolic knowledge graph embeddings, 2020
[12] Parisot, Ktena, Ferrante, Lee, Guerrero, Glocker, Rueckert, Disease prediction using graph networks: Application to Autism Spectrum Disorder and Alzheimer's disease, 2018
[13] Johnson, Gupta, Fei-Fei, Image generation from scene graphs, 2018
[14] Vaswani, Shazeer, Parmar, Uszkoreit, Jones, N Gomez, Kaiser, Polosukhin, Attention is all you need, 2017
[15] Yasunaga et-al, Deep Bidirectional Language-Knowledge Graph Pretraining, 2022
[16] Vinyals, Fortunato, Jaitly, Pointer, 2015
[17] Bello, Pham, Le, Norouzi, Bengio, Neural combinatorial optimization with reinforcement learning, 2016

A decade of GNN development

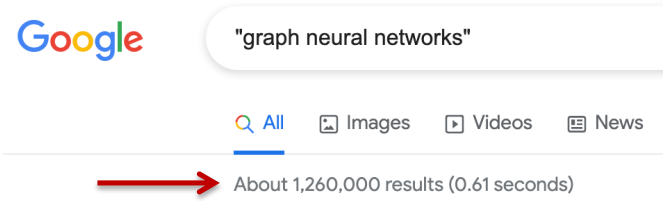
- **Brief history of graph networks :**





$$\text{Node-update}: \quad h_i^{\ell+1} = f_{\text{node}}\left(h_i^\ell, \{h_j^\ell, e_{ij}^\ell : j \in \mathcal{N}_i\}\right) \in \mathbb{R}^d$$

$$\text{Edge-update}: \quad e_{ij}^{\ell+1} = f_{\text{edge}}\left(e_{ij}^\ell, h_i^\ell, h_j^\ell\right) \in \mathbb{R}^d$$



Outline

- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

Progress or not

- The field has grown extensively in the last few years ☺
- Unfortunately, evaluating the effectiveness of new architectures/ideas has become difficult for two major reasons.
  - We have been evaluated progress on small datasets such as Cora[1], Citeseer[2] and TU[3].
    - Cora is a single graph of 2.7K nodes, TU-IMDB has 1.5K graphs with 13 nodes on average, and TU-MUTAG has 188 molecules with 18 nodes.
    - Simple or graph-agnostic architectures provide statistically same performance as complex architectures[4,5] (74.2 ± 6.3 vs. 75.6 ± 7.2).
  - We have not rigorously enforced standardized experimental settings for fair comparisons between models[6].
- It has become critical to solve these issues to trust the current models and new ideas.

[1] McCallum, Nigam, Rennie, Seymore, Automating the construction of internet portals with machine learning, 2000
[2] Getoor, Link-based classification, 2005
[3] Kersting, Kriege, Morris, Mutzel, Neumann, Benchmark data sets for graph kernels, 2020
[4] Hoang, Maehara, Revisiting graph neural networks, 2019
[5] Chen, Bian, Sun, Are powerful graph neural nets necessary? a dissection on graph classification, 2019
[6] Errica, Podda, Bacciu, Micheli, A fair comparison of graph neural networks for graph classification, 2019

Benchmarking

- How to identify and quantify what types of architectures, first principles are universal, generalizable, and scalable to larger and more challenging datasets?
- Theoretical approach : Assumptions on data, design model, prove expressivity and generalization bound (latter is hard because it requires to simultaneously analyze data property, architecture, loss energy landscape and optimization).
- Empirical approach : Benchmarking has been beneficial for driving progress, identifying essential ideas, and solving domain-specific problems[1].
  - The 2012 ImageNet challenge[2] has provided a benchmark dataset that has triggered the deep learning revolution.
- Challenges
  - Developing a rigorous experimental setting for fair comparisons and being reproducible.
  - Using datasets that can statistically separate model performance.
  - Benchmarking distinct fundamental graph tasks.

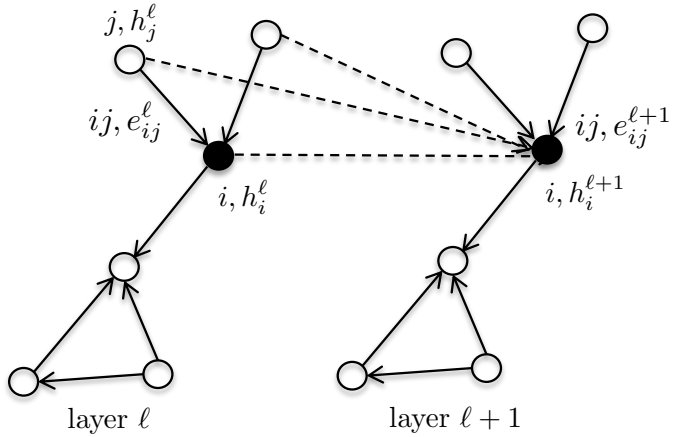
[1] Weber, Saelens, Cannoodt, Soneson, Hapfelmeier, Gardner, Boulesteix, Saeys, Robinson, Essential guidelines for computational method benchmarking, 2019
[2] Deng, Dong, Socher, Li, Li, Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, 2009

Outline

- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

Message Passing GNNs

- The most popular and oldest class of GNNs – obsolete?
- What are the minimal inner structures to design a MP-GNN?
  - Invariant by node permutation (equivariant/invariant layers)
  - Independent of graph size n and neighborhood size
  - Locality (local reception field - only neighbors are considered)
  - Graph convolution operator (weight sharing across graph)
  - Linear complexity O(E), E being the number of edges (reducing to O(n) for sparse/real graphs)



Node/edge-update :

$$h_i^{\ell+1} = f_{\text{node}}\big(h_i^\ell, \{h_j^\ell, e_{ij}^\ell : j \in \mathcal{N}_i\}\big) \in \mathbb{R}^d$$
$$e_{ij}^{\ell+1} = f_{\text{edge}}\big(e_{ij}^\ell, h_i^\ell, h_j^\ell\big) \in \mathbb{R}^d$$

Isotropic MP-GNNs

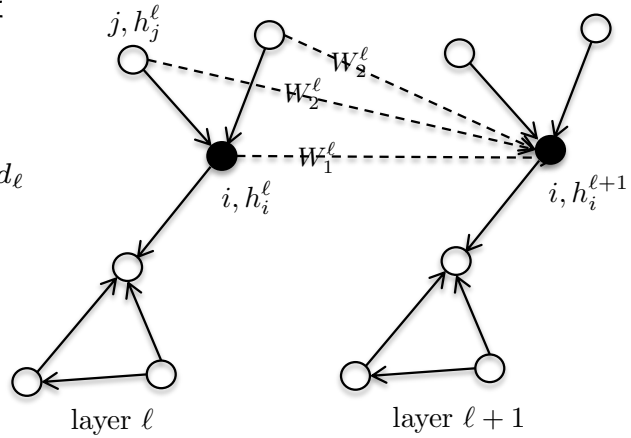
- We consider the class of isotropic MP-GNNs when the node update equation treats every "edge direction" equally, i.e. each neighbor contributes equally to the update of the central node :

$$h_i^{\ell+1} = \sigma\Big(W_1^\ell \; h_i^\ell + \sum_{j \in \mathcal{N}_i} W_2^\ell h_j^\ell\Big),$$

ReLU

Sum/mean/max

$$h^{\ell+1} \in \mathbb{R}^{n \times d_{\ell+1}}, \; h^\ell \in \mathbb{R}^{n \times d_\ell}, \; W_{1,2}^\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$$

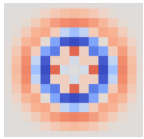


- Models :
  - GCNs[1,2]
  - GraphSage[3]
  - ChebNets[4]

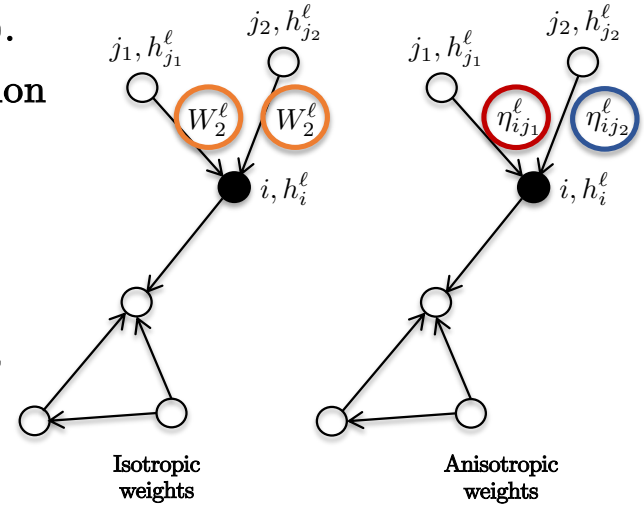
- Isotropic MP-GNNs are limited to learn two template weights; one for the central node h_i (weight W_1) and one for the 1-hop neighbors h_j (weight W_2).

[1] Sukhbaatar, Szlam, Fergus, Learning multiagent communication with backpropagation, 2016
[2] Kipf, Welling, Semi-supervised classification with graph convolutional networks, 2017
[3] Hamilton, Ying, Leskovec, Inductive representation learning on large graphs, 2017
[4] Defferrard, Bresson, Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, 2016

Anisotropic MP-GNNs



- MP-GNNs such as vanilla GCNs, GraphSage, ChebNets compute isotropic filters as there is no notion of direction on arbitrary graphs.
- How to get anisotropy back?
  - Natural edge features[1] if available (e.g. different bond connections between atoms in molecular graphs or distinct edge attributes in knowledge graphs).
  - Learn anisotropy with a mechanism invariant by index permutation
    - MoNets[2] with edge degrees
    - GatedGCNs [3] with edge gates
    - GAT[4] with attention mechanism[7]
- We instantiate an anisotropic MP-GNN with an update equation that treats every edge direction differently



Learnable anisotropic weights

$$h_i^{\ell+1} = \sigma\Big(W_1^\ell\ h_i^\ell + \sum_{j\in\mathcal{N}_i} \eta_{ij}^\ell W_2^\ell h_j^\ell\Big),\ h^{\ell+1}\in\mathbb{R}^{n\times d_{\ell+1}},\ h^\ell\in\mathbb{R}^{n\times d_\ell},\ W_{1,2}^\ell\in\mathbb{R}^{d_{\ell+1}\times d_\ell},$$

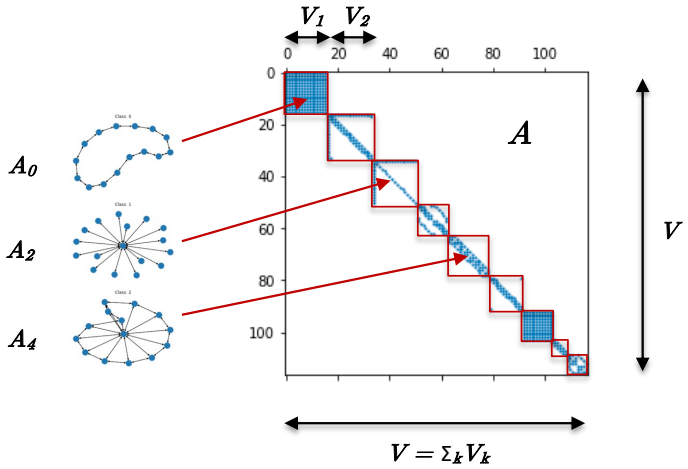
[1] Gilmer, Schoenholz, Riley, Vinyals, Dahl, Neural message passing for quantum chemistry, 2017
[2] Monti, Boscaini, Masci, Rodolà, J. Svoboda, M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model CNNs, 2016
[3] Bresson, Laurent, Residual gated graph convnets, 2017
[4] Velickovic, Cucurull, Casanova, Romero, Lio, Bengio, Graph Attention Networks, 2018

Batch normalization and residual connection

- MP-GNNs benefit from BN[1,2] and RC[1,3]
  - Speed up learning process
  - Improve performance with better generalization

$$h_i^{\ell+1} = h_i^{\ell} + \sigma\Big(\mathrm{BN}\left(\hat{h}_i^{\ell+1}\right)\Big)$$
$$\hat{h}_i^{\ell+1} = f_{\text{MP-GNN}}\left(\, h_i^{\ell}\,,\, \{h_j^{\ell} : j \in \mathcal{N}_i\}\,\right)$$

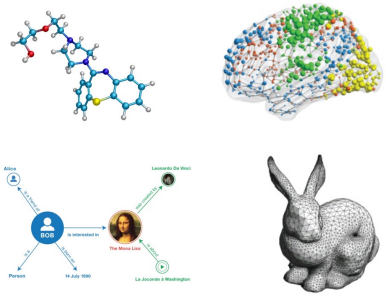
- How to batch graphs of different sizes ?
  - For images of size B x V_x x V_y x d, BN is performed along the batch dimension, i.e. dim=0.
  - For graphs, a (big) sparse block diagonal matrix A of size V x d is first built from K matrices A_k and BN is carried out along the node direction, i.e. dim=0.



[1] Bresson, Laurent, Residual gated graph convnets, 2017
[2] Ioffe, Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015
[3] Li, Muller, Qian, Delgadillo, Abualshour, Thabet, Ghanem, Deepgcns: Making gcns go as deep as cnns, 2019

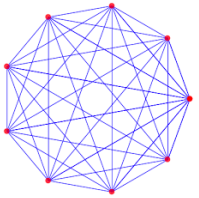
Sparsity and local computations

- MP-GNNs leverage graph sparsity
  - Sparsity is a great inductive bias for generalization
  - Sparse vs full graphs



Sparse graphs

vs.



Fully connected graph

- Local computations
  - Node/edge update equations are local, only depending on neighborhood $\mathcal{N}_i$ of node i, and independent of graph size n, making the space/time complexity O(n) for sparse graphs.
  - MP-GNNs are naturally parallelizable, and fast sparse matrix multiplications are implemented via GPU-based libraries s.a. DGL[1], PyG[2], TensorFlowGNNs[3].

$$h_i^{\ell+1} = f_{\text{node}}\left(h_i^{\ell}, \{h_j^{\ell}, e_{ij}^{\ell} : j \in \mathcal{N}_i\}\right) \in \mathbb{R}^d$$
$$e_{ij}^{\ell+1} = f_{\text{edge}}\left(e_{ij}^{\ell}, h_i^{\ell}, h_j^{\ell}\right) \in \mathbb{R}^d$$

DGL PyTorch geometric TensorFlow GNN

[1] Wang-etal, Deep graph library: Towards efficient and scalable deep learning on graphs, 2019
[2] Fey, Lenssen, Fast graph representation learning with pytorch geometric, 2019
[3] Oleksandr et-al, TF-GNN: Graph Neural Networks in TensorFlow, 2022

Graph isomorphism networks

- GINs[1] : MP-GNN as expressive as the 1-WL test to distinguish non-isomorphic graphs.
- Node aggregation of the form :

$$f_{\text{NN}}\left(h_i^\ell, \{h_j^\ell\}_{j\in\mathcal{N}_i}\right) = (1+\varepsilon)g(h_i^\ell) + \sum_{j\in\mathcal{N}_i} g(h_j^\ell)$$

irrational injective sum

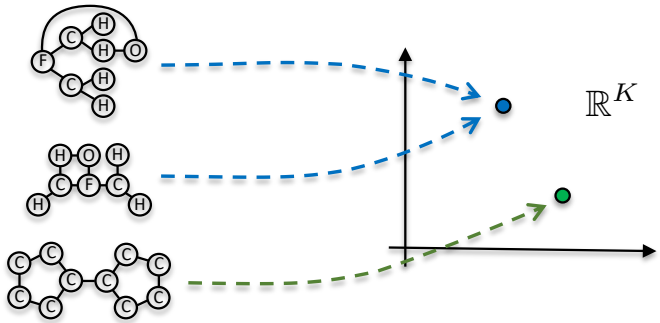
- It is difficult to design an analytical injective function ⇒ MLP is used to approximate g (existence guaranteed by the universal approximation theorem) :

$$h_i^{\ell+1} = f_{\text{GIN}}\left(h_i^\ell, \{h_j^\ell\}_{j\in\mathcal{N}_i}\right) = \text{MLP}^\ell\left((1+\varepsilon)h_i^\ell + \sum_{j\in\mathcal{N}_i} h_j^\ell\right)$$

- Graph readout function must also be injective :

$$h_{\mathcal{G}} = \text{MLP}\left(\sum_{i\in V} h_i^L\right) \in \mathbb{R}^K$$

sum

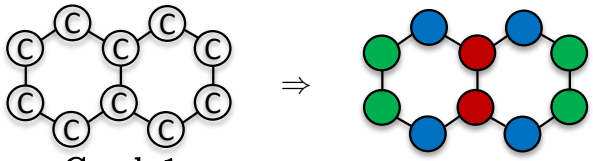


- Pioneer work on GNN expressivity

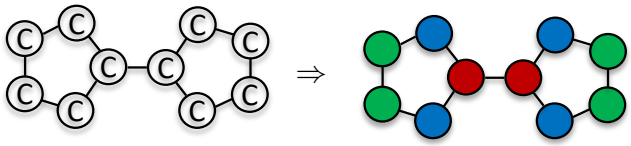
[1] Xu, Hu, Leskovec, Jegelka, How powerful are graph neural networks?, 2019

Limitation of 1-WL test

- WL test[1] is not a sufficient condition, it can fail to differentiate simple non-isomorphic graphs.
  - Example of two simple distinct graphs with same color signatures but not isomorphic



Graph 1



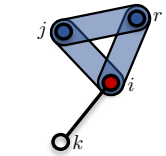
Graph 2

  - Need to improve the expressivity of the original WL test

[1] Weisfeiler, Lehman, A reduction of a graph to a canonical form and an algebra arising during this reduction, 1968

Equivariant GNNs

- How to design GNNs with the same expressivity power as the k-WL test?
  - Let us define a k-order equivariant GNNs[1] :



Hyper-edges $=\{i,j,r\}, \{i,k,r\}, \{i,j,k\}$

Permutation

$f_W(P \circ h) = P \circ f_W(h)$

Equivariant linear layers

MLP

$$y = m_{W^{L+1}} \circ g_{W^L} \circ \sigma \circ f_{W^{L-1}} \circ \sigma \circ f_{W^{L-2}} \circ \ldots \sigma \circ f_{W^0} \circ h_0$$

$$\text{with } k = \max_{\ell \in [0, L-1]} k_\ell$$

$g_W(P \circ h) = g_W(h) \in \mathbb{R}^K, K \geq 1$

Invariant linear layer

$$\text{and } f_{W^\ell} : \mathbb{R}^{n^{k_\ell} \times d_\ell} \to \mathbb{R}^{n^{k_{\ell+1}} \times d_{\ell+1}}$$

- Theorem : There exist k-order E-GNNs that can distinguish non-isomorphic graphs with the k-WL test.
  - However, k-order E-GNNs requite $O(n^k)$ memory/speed complexities.
    - Note that we need at least k=3, therefore $O(n^3)$, to be more powerful than GINs as GINs have the discriminative power of the 1-WL/2-WL test[2].

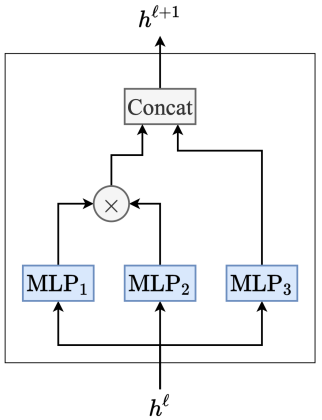
[1] Maron, Ben-Hamu, Shamir, Lipman, Invariant and equivariant graph networks, 2019
[2] Cai, Furer, Immerman, An optimal lower bound on the number of variables for graph identification, 1992

3-WL GNNs

- How to design GNNs that are 3-WL expressive but do not require $O(n^3)$ memory complexity?
- 3-WL GNNs[1] : To achieve higher interactions between nodes, it is sufficient to multiply second-order tensors feature-wise.
  - Theorem : There exist 3-WL GNNs as expressive as the 3-WL test.
  - Memory is quadratic $O(n^2)$ but matrix multiplication implies $O(n^3)$ speed.
    - Also, matrix multiplication densifies sparse matrix.
  - This is one of the simplest, most scalable and most expressive GNNs in terms of 3-WL test.
- 3-WL GNN update layer :

$$h^{\ell+1} = \text{Concat}\big(m_{W_1^\ell}(h^\ell) \cdot m_{W_2^\ell}(h^\ell),\ m_{W_3^\ell}(h^\ell)\big)$$

where $h^{\ell+1} \in \mathbb{R}^{n\times n\times d_{\ell+1}}, h^\ell \in \mathbb{R}^{n\times n\times d_\ell}, W_1, W_2, W_3 \in \mathbb{R}^{2\times d_\ell \times d_{\ell+1}}$



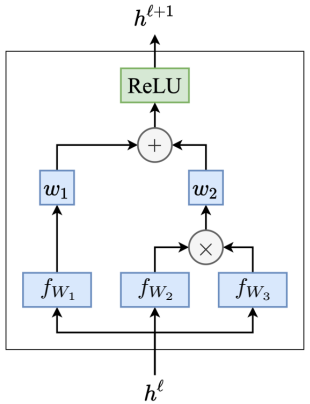
[1] Maron, Ben-Hamu, Serviansky, Lipman, Provably powerful graph networks, 2019

RingGNNs

- Related method to 3-WL GNNs to design more expressive GNNs than GINs.
  - Higher-order interaction between nodes is produced by multiplying equivariant linear layers[2].
  - RingGNN[1] update layer :

$$h^{\ell+1} = \sigma\big(w_1^\ell \; f_{W_1^\ell}(h^\ell) + w_2^\ell \; f_{W_2^\ell}(h^\ell) \cdot f_{W_3^\ell}(h^\ell)\big),$$

$$\text{where } h^{\ell+1} \in \mathbb{R}^{n\times n\times d_{\ell+1}}, h^\ell \in \mathbb{R}^{n\times n\times d_\ell}, w_{1,2}^\ell \in \mathbb{R}, W_1, W_2, W_3 \in \mathbb{R}^{d_\ell \times d_{\ell+1}\times 17},$$



where f_W are the linear equivariant layers defined as :

$$\big(f_W(h)\big)_{\cdot,\cdot,q'} = \sum_{k=1}^{15+2} \sum_{q=1}^{d_\ell} W_{k,q,q'} \; f_k(h_{\cdot,\cdot,q}) \;\; \in \mathbb{R}^{n\times n\times d_{\ell+1}}$$

and f_k are all possible 15 equivariant linear functions $f_k : R^{n\,x\,n} \to R^{n\,x\,n}$ for a given tensor $h \in R^{n\,x\,n}$, and 2 bias functions.

[1] Chen, Villar, Chen, Bruna, On the equivalence between graph isomorphism testing and function approximation with gnns, 2019
[2] Maron, Ben-Hamu, Shamir, Lipman, Invariant and equivariant graph networks, 2019

MLP baseline

- Graph-agnostic NNs : As a sanity check, we compare GNNs to a simple MLP network which updates each node independent of one-other :

$$h_i^{\ell+1} = \sigma\Big(W^\ell \; h_i^\ell\Big), \; h^{\ell+1} \in \mathbb{R}^{n \times d_{\ell+1}}, \; h^\ell \in \mathbb{R}^{n \times d_\ell}, \; W^\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$$

and passes these features to the task-based readout layer.

Outline

- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

Designing datasets

- The main issue with datasets used in most GNN papers is their small size.
- What are the best properties for a dataset?
  - Representative, realistic, and medium/large-scale size
- Challenges
  - What theoretical tool to define the quality of dataset or validate its statistical representativeness for a given task?
  - What are the best features? Several arbitrary choices when preparing graphs, such as node and edge features. For example, e-commerce product features.
  - What size? Appropriate size may depend on the task complexity as well as the dimensionality and statistics of underlying data. Besides, we wish to use the smallest dataset that is representative of the task because GPUs are expensive and scared resources.
  - How to quantify and identify noise/error in real datasets? Synthetic datasets control the noise but are not real. Real datasets are noisy.
- In practice, there is no perfect dataset.

Proposed datasets

- Appropriate datasets :
  - Datasets that are able to statistically separate the performance of GNNs (no more 74.2±6.3 vs. 75.6±7.2).
- Summary of the 7 medium-scale datasets used in the benchmark[1] :

Domain & Construction	Dataset	#Graphs	#Nodes	Total #Nodes	Task
Chemistry: Real-world molecular graphs	ZINC	12K	9-37	277,864	Graph Regression
Mathematical Modelling: Artificial graphs generated from Stochastic Block Models	PATTERN CLUSTER	14K 12K	44-188 41-190	1,664,491 1,406,436	Node Classification
Computer Vision: Graphs constructed with SLIC super-pixels of images	MNIST CIFAR10	70K 60K	40-75 85-150	4,939,668 7,058,005	Graph Classification
Combinatorial Optimization: Uniformly generated artificial Euclidean graphs	TSP	12K	50-500	3,309,140	Edge Classification
Social Networks: Real-world citation graph	COLLAB	1	235,868	235,868	Edge Classification
Circular Skip Links: Isomorphic graphs with same degree	CSL	150	41	6,150	Graph Classification

4 datasets are artificially generated, 2 datasets are semi-artificial, and 2 are real-world.

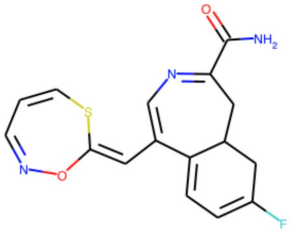
Sizes in terms of total number of nodes vary between 0.27M to 7M.

4 most fundamental graph tasks : graph regression, graph classification, node classification and link prediction.

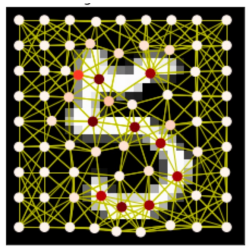
[1] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020

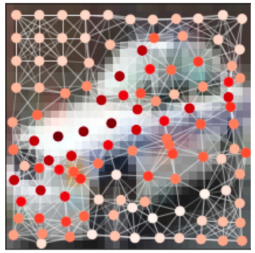
Proposed datasets

- ZINC[1] : A popular real-world molecular dataset of 250K graphs, out of which we randomly select 12K for efficiency. We consider the task of graph property regression for constrained solubility, an important chemical property for designing generative GNNs for molecules[2].
  - Statistics : 10,000 train/1,000 validation/1,000 test graphs of sizes 9-37 nodes/heavy atoms.



- MNIST[3]/CIFAR10[4] : Classical image classification datasets converted into graphs using super-pixels[5] and assigning node features as the super-pixel coordinates and mean intensity. These datasets are sanity-checks, as we expect most GNNs to perform close to 100% for MNIST and well enough for CIFAR10.
  - Statistics : MNIST has 55,000 train/5,000 validation/10,000 test graphs of sizes 40-75 nodes and CIFAR10 has 45,000 train/5,000 validation/10,000 test graphs of sizes 85-150 nodes.

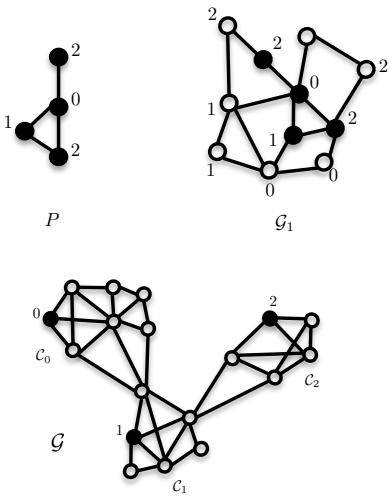




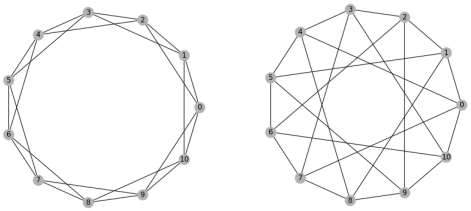
[1] Irwin, Sterling, Mysinger, Bolstad, Coleman, Zinc: a free tool to discover chemistry for biology, 2012
[2] Jia, Lin, Ying, You, Leskovec, Aiken, Redundancy-free computation graphs for graph neural networks, 2019
[3] LeCun, Bottou, Bengio, Haffner, Gradient-based learning applied to document recognition, 1998
[4] Alex Krizhevsky etal, Learning multiple layers of features from tiny images, 2009
[5] Achanta, Shaji, Smith, Lucchi, Fua, Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, 2012

Proposed datasets

- PATTERN/CLUSTER : Node classification tasks generated with Stochastic Block Models[1], which are widely used to model communities in social networks by modulating the intra- and extra-communities connections. PATTERN tests the fundamental task of recognizing specific predetermined subgraphs[2].
  - Statistics : PATTERN has 10,000 train/2,000 validation/2,000 test graphs of sizes 50-180 nodes. CLUSTER has 10,000 train/1,000 validation/1,000 test graphs of sizes 40-190 nodes.



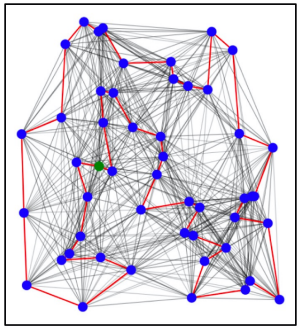
- CSL[3] : Synthetic to test the expressivity of GNNs. Graphs are isomorphic if they have the same degree and the task is to classify non-isomorphic graphs.
  - Statistics : 5-fold cross-validation split of 150 graphs of sizes 41 nodes with train-val-test ratio 3:1:1.



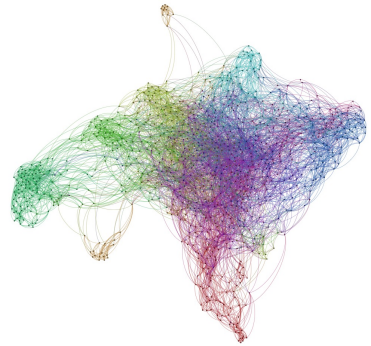
[1] Abbe, Community detection and stochastic block models: recent developments, 2017
[2] Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini, The Graph Neural Network Model, 2009
[3] Murphy, Srinivasan, Rao, Ribeiro, Relational pooling for graph representations, 2019

Proposed datasets

- Traveling Salesman Problem (TSP) : Link prediction on 2D Euclidean graphs to identify edges belonging to the optimal TSP solution given by Concorde[1]. TSP is the most studied NP-hard combinatorial problem with a growing body of literature on leveraging GNNs to learn better solvers[2,3].
  - Statistics : 10,000 train TSPs / 1,000 validation TSPs / 1,000 test TSPs, where the number of nodes is randomly selected in [50, 500].



- OGB-COLLAB[4] : Link prediction dataset proposed by OGB corresponding to a collaboration network between scientists. The task is to predict future author collaboration relationships given past collaboration links.
  - Statistics : A single large temporal graph of size 235K nodes with given train/validation/test edge splits.



[1] Applegate, Bixby, Chvatal, Cook, Concorde tsp solver, 2006
[2] Vinyals, Fortunato, Jaitly, Pointer, 2015
[3] Bello, Pham, Le, Norouzi, Bengio, Neural combinatorial optimization with reinforcement learning, 2016
[4] Hu, Fey, Zitnik, Dong, Ren, Liu, Catasta, Leskovec, Open graph benchmark: Datasets for machine learning on graphs, 2020

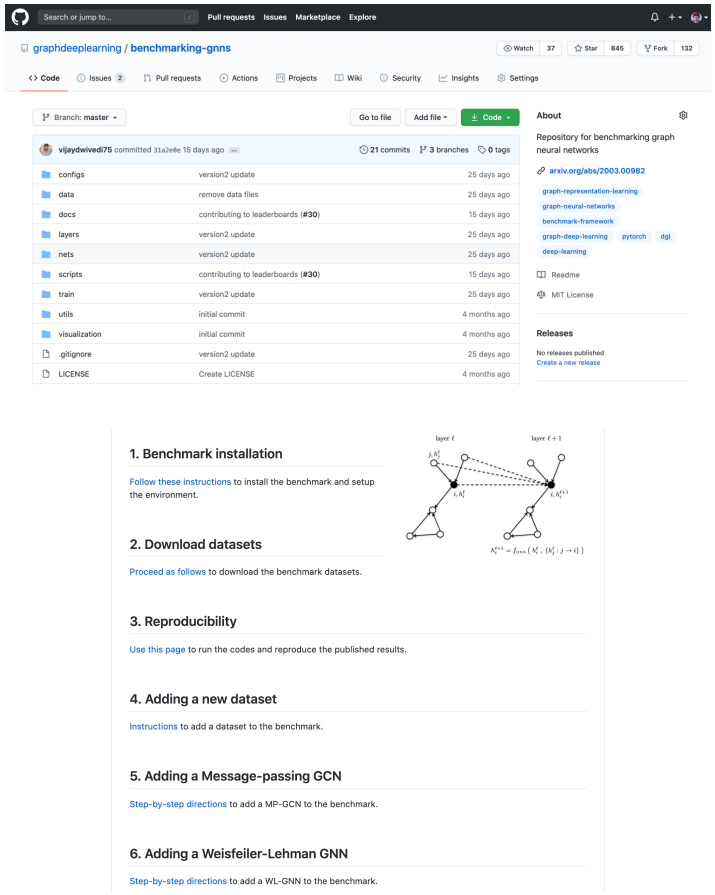
Outline

- Emerging field
- Tracking progress
- **Benchmarking GNNs**
  - Architectures
  - Datasets
  - **Code infrastructure**
  - Experimental setting
  - Results and insights
- Graph datasets
- Conclusion

Benchmark infrastructure

- GitHub Repo (2.1k+ stars, 4k+ forks as of Mar 2023)

  https://github.com/graphdeeplearning/benchmarking-gnns
- Objectives
  - Ease-of-use and modular, enabling new users to experiment and study the building blocks of GNNs.
  - Experimentally rigorous and fair for all models being benchmarked.
  - Being comprehensive for tracking progress of new GNNs and novel dataset/task.
- Components
  - Data pipeline
  - GNN layers and models
  - Training and evaluation functions
  - Network and hyperparameter configurations
  - Scripts for reproducibility



Outline

- Emerging field
- Tracking progress
- **Benchmarking GNNs**
  - Architectures
  - Datasets
  - Code infrastructure
  - **Experimental setting**
  - Results and insights
- Graph datasets
- Conclusion

Experimental setting

- Data splits : Given for ZINC, MNIST, CIFAR10, CSL, OGB-COLLAB, random for PATTERN, CLUSTER, TSP.
- Training :
  - Adam optimizer w/ learning rate decay strategy.

    Initial learning rate is 1e-3/1e-4, reduced by half if validation loss does not improve after 5/10 epochs.

    Training is stopped if learning rate $\leq$ 1e-6 or computational time $\geq$ 12 hours.
  - Statistics with 4 results using 4 different seeds are reported.
- Parameter budgets :
  - Our goal is not to find the optimal hyperparameters (computationally expensive) but to compare models within the same parameter budget.
  - Two parameter budgets :
    - 100k (w/ L=4) and 500k (w/ L=16) parameters for each GNNs for all tasks.

Outline

- Emerging field
- Tracking progress
- **Benchmarking GNNs**
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - **Results and insights**
- Graph datasets
- Conclusion

Benchmarking results

- Result #1 : MP-GNNs outperformed WL-GNNs on all datasets.
  - Potential reasons :
    - $O(n^2)/O(n^3)$ memory/speed complexities of WL-GNNs do not scale to medium-scale datasets.
      - Best result for ZINC, which is the dataset with the smallest sizes, n ∈ [9,37].
    - Early stage of developments of WL-GNNs.
- Result #2 : MP-GNNs benefit from batch normalization and residual connection :
  - Boost performance
  - Accelerate training
  - WL-GNNs do not benefit from RC, BN or LN (performance degrades).

NODE CLASSIFICATION

Model	L	PATTERN #Param	PATTERN Test Acc.±s.d.	PATTERN Train Acc.±s.d.	PATTERN #Epoch	PATTERN Epoch/Total	CLUSTER #Param	CLUSTER Test Acc.±s.d.	CLUSTER Train Acc.±s.d.	CLUSTER #Epoch	CLUSTER Epoch/Total
MLP	4	105263	50.519±0.000	50.487±0.014	42.25	8.95s/0.11hr	106015	20.973±0.004	20.938±0.002	42.25	5.83s/0.07hr
GCN	4	100923	63.880±0.074	65.126±0.135	105.00	118.85s/3.51hr	101655	53.445±2.029	54.041±2.197	70.00	65.72s/1.30hr
	16	500823	71.892±0.334	78.409±1.592	81.50	492.19s/11.31hr	501687	68.498±0.976	71.729±2.212	79.75	270.28s/6.08hr
GraphSage	4	101739	50.516±0.001	50.473±0.014	43.75	93.41s/1.17hr	102187	50.454±0.145	54.374±0.203	64.00	53.56s/0.97hr
	16	502842	50.492±0.001	50.487±0.005	46.50	391.19s/5.19hr	503350	63.844±0.110	86.710±0.167	57.75	225.61s/3.70hr
MoNet	4	103775	85.482±0.037	85.569±0.044	89.75	35.71s/0.90hr	104227	58.064±0.131	58.454±0.183	76.25	24.29s/0.52hr
	16	511487	85.582±0.038	85.720±0.068	81.75	68.49s/1.58hr	511999	66.407±0.540	67.727±0.649	77.75	47.82s/1.05hr
GAT	4	109936	75.824±1.823	77.883±1.632	96.00	20.92s/0.57hr	110700	57.732±0.323	58.331±0.342	67.25	14.17s/0.27hr
	16	526990	78.271±0.186	90.212±0.476	53.50	50.33s/0.77hr	527874	70.587±0.447	76.074±1.362	73.50	35.94s/0.75hr
GatedGCN	4	104003	84.480±0.122	84.474±0.155	78.75	139.01s/3.09hr	104355	60.404±0.419	61.618±0.536	94.50	79.97s/2.13hr
	16	502223	85.568±0.088	86.007±0.123	65.25	644.71s/11.91hr	502615	73.840±0.326	87.880±0.908	60.00	400.07s/6.81hr
GatedGCN-PE	16	502457	**86.508±0.085**	86.801±0.133	65.75	647.94s/12.08hr	504253	**76.082±0.196**	88.919±0.720	57.75	399.66s/6.58hr
GIN	4	100884	85.590±0.011	85.852±0.030	93.00	15.24s/0.40hr	103544	58.384±0.236	59.480±0.337	74.75	10.71s/0.23hr
	16	508574	85.387±0.136	85.664±0.116	86.75	25.14s/0.62hr	517570	64.716±1.553	65.973±1.816	80.75	20.67s/0.47hr
RingGNN	2	105206	86.245±0.013	86.118±0.034	75.00	573.37s/12.17hr	104746	42.418±20.063	42.520±20.212	74.50	428.24s/8.79hr
	2	504766	86.244±0.025	86.105±0.021	72.00	595.97s/12.15hr	524202	22.340±0.000	22.304±0.000	43.25	501.84s/6.22hr
	8	505749	Diverged	Diverged	Diverged	Diverged	514380	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	103572	85.661±0.353	85.608±0.337	95.00	304.79s/7.88hr	105552	57.130±6.539	57.404±6.597	116.00	219.51s/6.52hr
	3	502872	85.341±0.207	85.270±0.198	81.75	424.23s/9.56hr	507252	55.489±7.863	55.736±8.024	66.00	319.98s/5.79hr
	8	581716	Diverged	Diverged	Diverged	Diverged	586788	Diverged	Diverged	Diverged	Diverged

GRAPH CLASSIFICATION

Model	L	MNIST #Param	MNIST Test Acc.±s.d.	MNIST Train Acc.±s.d.	MNIST #Epoch	MNIST Epoch/Total	CIFAR10 #Param	CIFAR10 Test Acc.±s.d.	CIFAR10 Train Acc.±s.d.	CIFAR10 #Epoch	CIFAR10 Epoch/Total
MLP	4	104044	95.340±0.138	97.432±0.470	232.25	22.74s/1.48hr	104380	56.340±0.181	65.113±1.685	185.25	29.48s/1.53hr
GCN	4	101365	90.705±0.218	97.196±0.223	127.50	83.41s/2.99hr	101657	55.710±0.381	69.523±1.948	142.50	109.70s/4.39hr
GraphSage	4	104337	**97.312±0.097**	100.000±0.000	98.25	113.12s/3.13hr	104517	65.767±0.308	99.719±0.062	93.50	124.61s/5.29hr
MoNet	4	104049	90.805±0.032	96.609±0.440	146.25	93.19s/3.82hr	104229	54.655±0.518	65.911±2.515	141.50	97.13s/3.85hr
GAT	4	110400	95.535±0.205	99.994±0.008	104.75	42.26s/1.25hr	110704	64.223±0.455	89.114±0.499	103.75	55.27s/1.62hr
GatedGCN	4	104217	**97.340±0.143**	100.000±0.000	96.25	128.79s/3.50hr	104357	**67.312±0.311**	94.553±1.018	97.00	154.15s/4.22hr
GIN	4	105434	96.485±0.252	100.000±0.000	128.00	39.22s/1.41hr	105654	55.255±1.527	79.412±9.700	141.50	52.12s/2.07hr
RingGNN	2	105398	11.350±0.000	11.235±0.000	14.00	2945.69s/12.77hr	105165	19.300±16.108	19.556±16.397	13.50	3112.96s/13.00hr
	2	505182	91.860±0.449	92.169±0.505	16.25	2575.99s/12.63hr	504949	39.165±17.114	40.209±17.790	13.75	2998.24s/12.60hr
	8	506357	Diverged	Diverged	Diverged	Diverged	510439	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	108024	95.075±0.961	95.830±1.338	27.75	1523.20s/12.40hr	108516	59.175±1.593	63.751±2.697	28.50	1506.29s/12.60hr
	3	501690	95.002±0.419	95.692±0.677	26.25	1608.73s/12.42hr	502770	58.043±2.512	61.574±3.575	20.00	2091.22s/12.55hr
	8	500816	Diverged	Diverged	Diverged	Diverged	501584	Diverged	Diverged	Diverged	Diverged

LINK PREDICTION

Model	L	TSP #Param	TSP Test F1±s.d.	TSP Train F1±s.d.	TSP #Epoch	TSP Epoch/Total	COLLAB #Param ($L=3$)	COLLAB Test Hits±s.d.	COLLAB Train Hits±s.d.	COLLAB #Epoch	COLLAB Epoch/Total
MLP	4	96956	0.544±0.001	0.544±0.001	164.25	50.15s/2.31hr	39441	20.350±2.168	29.807±3.360	147.50	2.09s/0.09hr
GCN	4	95702	0.630±0.001	0.631±0.001	261.00	152.89s/11.15hr	40479	50.422±1.131	92.112±0.991	122.50	351.05s/12.04hr
GraphSage	4	99263	0.665±0.003	0.669±0.003	266.00	157.26s/11.68hr	39856	51.618±0.690	99.949±0.052	152.75	277.93s/11.87hr
MoNet	4	99007	0.641±0.002	0.643±0.002	282.00	84.46s/6.65hr	39751	36.144±2.191	61.156±3.973	167.50	26.69s/1.26hr
GAT	4	96182	0.671±0.002	0.673±0.002	328.25	68.23s/6.25hr	42637	51.501±0.962	97.851±1.114	157.00	18.12s/0.80hr
GatedGCN	4	97858	0.791±0.003	0.793±0.003	159.00	218.20s/9.72hr	40965	52.635±1.168	96.103±1.876	95.00	453.47s/12.09hr
GatedGCN-PE							41889	**52.849±1.145**	96.165±0.453	94.75	452.75s/12.08hr
GatedGCN-E	4	97858	0.808±0.003	0.811±0.003	197.00	218.51s/12.04hr	40965	49.212±1.560	88.747±1.058	95.00	451.21s/12.03hr
GatedGCN-E	16	500770	**0.838±0.002**	0.850±0.001	53.00	807.23s/12.17hr					
GIN	4	99002	0.656±0.003	0.660±0.003	273.50	72.73s/5.56hr	39544	41.730±2.284	70.555±4.444	140.25	8.66s/0.34hr
RingGNN	2	106862	0.643±0.024	0.644±0.024	2.00	17850.52s/17.19hr	-	OOM			
	2	507938	0.704±0.003	0.705±0.003	3.00	12835.53s/16.08hr	-	OOM			
	8	506564	Diverged	Diverged	Diverged	Diverged	-	OOM			
3WLGNN	3	106366	0.694±0.073	0.695±0.073	2.00	17468.81s/16.59hr	-	OOM			
	3	506681	0.288±0.311	0.290±0.312	2.00	17190.17s/16.51hr	-	OOM			
	8	508832	OOM	OOM	OOM	OOM	-	OOM			

RingGNN and 3WLGNN rely on dense tensors which leads to **OOM** on both GPU and CPU memory.

k-NN Heuristic Matrix Fact.	0	k =2	Test F1: 0.693	-			60546561	44.206±0.452	100.000±0.000	254.33	2.66s/0.21hr

GRAPH REGRESSION – ZINC

Model	L	#Param	Test MAE±s.d.	Train MAE±s.d.	#Epoch	Epoch/Total
MLP	4	108975	0.706±0.006	0.644±0.005	116.75	1.01s/0.03hr
GCN	4	103077	0.459±0.006	0.343±0.011	196.25	2.89s/0.16hr
	16	505079	0.367±0.011	0.128±0.019	197.00	12.78s/0.71hr
GraphSage	4	94977	0.468±0.003	0.251±0.004	147.25	3.74s/0.15hr
	16	505341	0.398±0.002	0.081±0.009	145.50	16.61s/0.68hr
MoNet	4	106002	0.397±0.010	0.318±0.016	188.25	1.97s/0.10hr
	16	504013	0.292±0.006	0.093±0.014	171.75	10.82s/0.52hr
GAT	4	102385	0.475±0.007	0.317±0.006	137.50	2.93s/0.11hr
	16	531345	0.384±0.007	0.067±0.004	144.00	12.98s/0.53hr
GatedGCN	4	105735	0.435±0.011	0.287±0.014	173.50	5.76s/0.28hr
GatedGCN-E	4	105875	0.375±0.003	0.236±0.007	194.75	5.37s/0.29hr
	16	504309	0.282±0.015	0.074±0.016	166.75	20.50s/0.96hr
GatedGCN-E-PE	16	505011	**0.214±0.013**	0.067±0.019	185.00	10.70s/0.56hr
GIN	4	103079	0.387±0.015	0.319±0.015	153.25	2.29s/0.10hr
	16	509549	0.526±0.051	0.444±0.039	147.00	10.22s/0.42hr
RingGNN	2	97978	0.512±0.023	0.383±0.020	90.25	327.65s/8.32hr
RingGNN-E	2	104403	0.363±0.026	0.243±0.025	95.00	366.29s/9.76hr
	2	527283	0.353±0.019	0.236±0.019	79.75	293.94s/6.63hr
	8	510305	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	102150	0.407±0.028	0.272±0.037	111.25	286.23s/8.88hr
3WLGNN-E	3	103098	0.256±0.054	0.140±0.044	117.25	334.69s/10.90hr
	3	507603	0.303±0.068	0.173±0.041	120.25	329.49s/11.08hr
	8	582824	0.303±0.057	0.246±0.043	52.50	811.27s/12.15hr

Evaluation Metrics: (higher is better, except for ZINC)

- CLUSTER, PATTERN use weighted accuracy w.r.t. the class sizes.
- MNIST, CIFAR10 use multi-label classification accuracy.
- TSP uses binary F1 score for the positive edges.
- COLLAB uses Hits@50 via the evaluator provided by OGB [?].
- ZINC uses mean absolute regression error.

Notation:

- Models with the suffix **-E** use input edge features to initialize edge representations (ZINC: bond type, TSP: Euclidean distance, COLLAB: collaboration frequency and year).
- Models with the suffix **-PE** use Laplacian Eigenvectors as node positional encodings, with dimension 8 for ZINC, 2 for PATTERN and 20 for others.
- Results denoted by **Diverged** indicate unstable and divergent runs across all 4 seeds and initial learning rate values $\{10^{-3}, 10^{-4}, 10^{-5}\}$.
- Results denoted by **OOM** indicate runs which throw out of memory errors on our hardware configuration.

Extended training:

- For TSP, RingGNN/3WLGNN with 100K parameters achieved 0.733±0.020 and 0.649±0.051 respectively after 48hr of training.

Benchmarking results

- Result #3 : Anisotropic mechanism improve (isotropic) MP-GNNs.
  - Sparse attention[1,2] and dense attention[3] are not injective (unlike f_{GIN}) but experiments showed that they are good at generalization.
  - From a representation view, softmax attention is flexible to represent max/mean/weighted mean w.r.t. contextual information.

NODE CLASSIFICATION

Model	L	PATTERN #Param	PATTERN Test Acc.±s.d.	PATTERN Train Acc.±s.d.	PATTERN #Epoch	PATTERN Epoch/Total	CLUSTER #Param	CLUSTER Test Acc.±s.d.	CLUSTER Train Acc.±s.d.	CLUSTER #Epoch	CLUSTER Epoch/Total
MLP	4	105263	50.519±0.000	50.487±0.014	42.25	8.95s/0.11hr	106015	20.973±0.004	20.938±0.002	42.25	5.83s/0.07hr
GCN	4	100923	63.880±0.074	65.126±0.135	105.00	118.85s/3.51hr	101655	53.445±2.029	54.041±2.197	70.00	65.72s/1.30hr
	16	500823	71.892±0.334	78.409±1.592	81.50	492.19s/11.31hr	501687	68.498±0.976	71.729±2.212	79.75	270.28s/6.08hr
GraphSage	4	101739	50.516±0.001	50.473±0.014	43.75	93.41s/1.17hr	102187	50.454±0.145	54.374±0.203	64.00	53.56s/0.97hr
	16	502842	50.492±0.001	50.487±0.005	46.50	391.19s/5.19hr	503350	63.844±0.110	86.710±0.167	57.75	225.61s/3.70hr
MoNet	4	103775	85.482±0.037	85.569±0.044	89.75	35.71s/0.90hr	104227	58.064±0.131	58.454±0.183	76.25	24.29s/0.52hr
	16	511487	85.582±0.038	85.720±0.068	81.75	68.49s/1.58hr	511999	66.407±0.540	67.727±0.649	77.75	47.82s/1.05hr
GAT	4	109936	75.824±1.823	77.883±1.632	96.00	20.92s/0.57hr	110700	57.732±0.323	58.331±0.342	67.25	14.17s/0.27hr
	16	526990	78.271±0.186	90.212±0.476	53.50	50.33s/0.77hr	527874	70.587±0.447	76.074±1.362	73.50	35.94s/0.75hr
GatedGCN	4	104003	84.480±0.122	84.474±0.155	78.75	139.01s/3.09hr	104355	60.404±0.419	61.618±0.536	94.50	79.97s/2.13hr
	16	502223	85.568±0.088	86.007±0.123	65.25	644.71s/11.91hr	502615	73.840±0.326	87.880±0.908	60.00	400.07s/6.81hr
GatedGCN-PE	16	502457	**86.508±0.085**	86.801±0.133	65.75	647.94s/12.08hr	504253	**76.082±0.196**	88.919±0.720	57.75	399.66s/6.58hr
GIN	4	100884	85.590±0.011	85.852±0.030	93.00	15.24s/0.40hr	103544	58.384±0.236	59.480±0.337	74.75	10.71s/0.23hr
	16	508574	85.387±0.136	85.664±0.116	86.75	25.14s/0.62hr	517570	64.716±1.553	65.973±1.816	80.75	20.67s/0.47hr
RingGNN	2	105206	86.245±0.013	86.118±0.034	75.00	573.37s/12.17hr	104746	42.418±20.063	42.520±20.212	74.50	428.24s/8.79hr
	2	504766	86.244±0.025	86.105±0.021	72.00	595.97s/12.15hr	524202	22.340±0.000	22.304±0.000	43.25	501.84s/6.22hr
	8	505749	Diverged	Diverged	Diverged	Diverged	514380	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	103572	85.661±0.353	85.608±0.337	95.00	304.79s/7.88hr	105552	57.130±6.539	57.404±6.597	116.00	219.51s/6.52hr
	3	502872	85.341±0.207	85.270±0.198	81.75	424.23s/9.56hr	507252	55.489±7.863	55.736±8.024	66.00	319.98s/5.79hr
	8	581716	Diverged	Diverged	Diverged	Diverged	586788	Diverged	Diverged	Diverged	Diverged

GRAPH CLASSIFICATION

Model	L	MNIST #Param	MNIST Test Acc.±s.d.	MNIST Train Acc.±s.d.	MNIST #Epoch	MNIST Epoch/Total	CIFAR10 #Param	CIFAR10 Test Acc.±s.d.	CIFAR10 Train Acc.±s.d.	CIFAR10 #Epoch	CIFAR10 Epoch/Total
MLP	4	104044	95.340±0.138	97.432±0.470	232.25	22.74s/1.48hr	104380	56.340±0.181	65.113±1.685	185.25	29.48s/1.53hr
GCN	4	101365	90.705±0.218	97.196±0.223	127.50	83.41s/2.99hr	101657	55.710±0.381	69.523±1.948	142.50	109.70s/4.39hr
GraphSage	4	104337	**97.312±0.097**	100.000±0.000	98.25	113.12s/3.13hr	104517	65.767±0.308	99.719±0.062	93.50	124.61s/3.29hr
MoNet	4	104049	90.805±0.032	96.609±0.440	146.25	93.19s/3.82hr	104229	54.655±0.518	65.911±2.515	141.50	97.13s/3.85hr
GAT	4	110400	95.535±0.205	99.994±0.008	104.75	42.26s/1.25hr	110704	64.223±0.455	89.114±0.499	103.75	55.27s/1.62hr
GatedGCN	4	104217	**97.340±0.143**	100.000±0.000	96.25	128.79s/3.50hr	104357	**67.312±0.311**	94.553±1.018	97.00	154.15s/4.22hr
GIN	4	105434	96.485±0.252	100.000±0.000	128.00	39.22s/1.41hr	105654	55.255±1.527	79.412±9.700	141.50	52.12s/2.07hr
RingGNN	2	105398	11.350±0.000	11.235±0.000	14.00	2945.69s/12.77hr	105165	19.300±16.108	19.556±16.397	13.50	3112.96s/13.00hr
	2	505182	91.860±0.449	92.169±0.505	16.25	2575.99s/12.63hr	504949	39.165±17.114	40.209±17.790	13.75	2998.24s/12.60hr
	8	506357	Diverged	Diverged	Diverged	Diverged	510439	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	108024	95.075±0.961	95.830±1.338	27.75	1523.20s/12.40hr	108516	59.175±1.593	63.751±2.697	28.50	1506.29s/12.60hr
	3	501690	95.002±0.419	95.692±0.677	26.25	1608.73s/12.42hr	502770	58.043±2.512	61.574±3.575	20.00	2091.22s/12.55hr
	8	500816	Diverged	Diverged	Diverged	Diverged	501584	Diverged	Diverged	Diverged	Diverged

LINK PREDICTION

Model	L	TSP #Param	TSP Test F1±s.d.	TSP Train F1±s.d.	TSP #Epoch	TSP Epoch/Total	COLLAB #Param ($L=3$)	COLLAB Test Hits±s.d.	COLLAB Train Hits±s.d.	COLLAB #Epoch	COLLAB Epoch/Total
MLP	4	96956	0.544±0.001	0.544±0.001	164.25	50.15s/2.31hr	39441	20.350±2.168	29.807±3.360	147.50	2.09s/0.09hr
GCN	4	95702	0.630±0.001	0.631±0.001	261.00	152.89s/11.15hr	40479	50.422±1.131	92.112±0.991	122.50	351.05s/12.04hr
GraphSage	4	99263	0.665±0.003	0.669±0.003	266.00	157.26s/11.68hr	39856	51.618±0.690	99.949±0.052	152.75	277.93s/11.87hr
MoNet	4	99007	0.641±0.002	0.643±0.002	282.00	84.46s/6.65hr	39751	36.144±2.191	61.156±3.973	167.50	26.69s/1.26hr
GAT	4	96182	0.671±0.002	0.673±0.002	328.25	68.23s/6.25hr	42637	51.501±0.962	97.851±1.114	157.00	18.12s/0.80hr
GatedGCN	4	97858	0.791±0.003	0.793±0.003	159.00	218.20s/9.72hr	40965	52.635±1.168	96.103±1.876	95.00	453.47s/12.09hr
GatedGCN-PE			-	-	-	-	41889	**52.849±1.145**	96.165±0.453	94.75	452.75s/12.08hr
GatedGCN-E	4	97858	0.808±0.003	0.811±0.003	197.00	218.51s/12.04hr	40965	49.212±1.560	88.747±1.058	95.00	451.21s/12.03hr
GatedGCN-E	16	500770	**0.838±0.002**	0.850±0.001	53.00	807.23s/12.17hr			-		
GIN	4	99002	0.656±0.003	0.660±0.003	273.50	72.73s/5.56hr	39544	41.730±2.284	70.555±4.444	140.25	8.66s/0.34hr
RingGNN	2	106862	0.643±0.024	0.644±0.024	2.00	17850.52s/17.19hr	-	OOM			
	2	507938	0.704±0.003	0.705±0.003	3.00	12835.53s/16.08hr	-	OOM			
	8	506564	Diverged	Diverged	Diverged	Diverged	-	OOM			
3WLGNN	3	106366	0.694±0.073	0.695±0.073	2.00	17468.81s/16.59hr	-	OOM			
	3	506681	0.288±0.311	0.290±0.312	2.00	17190.17s/16.51hr	-	OOM			
	8	508832	OOM	OOM	OOM	OOM	-	OOM			
k-NN Heuristic Matrix Fact.	0	k =2	Test F1: 0.693	-			60546561	44.206±0.452	100.000±0.000	254.33	2.66s/0.21hr

RingGNN and 3WLGNN rely on dense tensors which leads to **OOM** on both GPU and CPU memory.

GRAPH REGRESSION – ZINC

Model	L	#Param	Test MAE±s.d.	Train MAE±s.d.	#Epoch	Epoch/Total
MLP	4	108975	0.706±0.006	0.644±0.005	116.75	1.01s/0.03hr
GCN	4	103077	0.459±0.006	0.343±0.011	196.25	2.89s/0.16hr
	16	505079	0.367±0.011	0.128±0.019	197.00	12.78s/0.71hr
GraphSage	4	94977	0.468±0.003	0.251±0.004	147.25	3.74s/0.15hr
	16	505341	0.398±0.002	0.081±0.009	145.50	16.61s/0.68hr
MoNet	4	106002	0.397±0.010	0.318±0.016	188.25	1.97s/0.10hr
	16	504013	0.292±0.006	0.093±0.014	171.75	10.82s/0.52hr
GAT	4	102385	0.475±0.007	0.317±0.006	137.50	2.93s/0.11hr
	16	531345	0.384±0.007	0.067±0.004	144.00	12.98s/0.53hr
GatedGCN	4	105735	0.435±0.011	0.287±0.014	173.50	5.76s/0.28hr
GatedGCN-E	4	105875	0.375±0.003	0.236±0.007	194.75	5.37s/0.29hr
	16	504309	0.282±0.015	0.074±0.016	166.75	20.50s/0.96hr
GatedGCN-E-PE	16	505011	**0.214±0.013**	0.067±0.019	185.00	10.70s/0.56hr
GIN	4	103079	0.387±0.015	0.319±0.015	153.25	2.29s/0.10hr
	16	509549	0.526±0.051	0.444±0.039	147.00	10.22s/0.42hr
RingGNN	2	97978	0.512±0.023	0.383±0.020	90.25	327.65s/8.32hr
RingGNN-E	2	104403	0.363±0.026	0.243±0.025	95.00	366.29s/9.76hr
	2	527283	0.353±0.019	0.236±0.019	79.75	293.94s/6.63hr
	8	510305	Diverged	Diverged	Diverged	Diverged
3WLGNN	3	102150	0.407±0.028	0.272±0.037	111.25	286.23s/8.88hr
3WLGNN-E	3	103098	0.256±0.054	0.140±0.044	117.25	334.69s/10.90hr
	3	507603	0.303±0.068	0.173±0.041	120.25	329.49s/11.08hr
	8	582824	0.303±0.057	0.246±0.043	52.50	811.27s/12.15hr

Evaluation Metrics: (higher is better, except for ZINC)
- CLUSTER, PATTERN use weighted accuracy w.r.t. the class sizes.
- MNIST, CIFAR10 use multi-label classification accuracy.
- TSP uses binary F1 score for the positive edges.
- COLLAB uses Hits@50 via the evaluator provided by OGB [?].
- ZINC uses mean absolute regression error.

Notation:
- Models with the suffix **-E** use input edge features to initialize edge representations (ZINC: bond type, TSP: Euclidean distance, COLLAB: collaboration frequency and year).
- Models with the suffix **-PE** use Laplacian Eigenvectors as node positional encodings, with dimension 8 for ZINC, 2 for PATTERN and 20 for others.
- Results denoted by **Diverged** indicate unstable and divergent runs across all 4 seeds and initial learning rate values $\{10^{-3}, 10^{-4}, 10^{-5}\}$.
- Results denoted by **OOM** indicate runs which throw out of memory errors on our hardware configuration.

Extended training:
- For TSP, RingGNN/3WLGNN with 100K parameters achieved 0.733±0.020 and 0.649±0.051 respectively after 48hr of training.

[1] Bahdanau, Cho, Bengio, Neural machine translation by jointly learning to align and translate, 2015
[2] Velickovic, Cucurull, Casanova, Romero, Lio, Bengio, Graph attention networks, 2017
[3] Bresson, Laurent, Residual gated graph convnets, 2017

Benchmarking results

- Result #4 : Explicit edge representation enhance link prediction task. We study three variants
  - No edge feature/no edge representation : Isotropic models (like vanilla GCNs[1])

$$h_i^{\ell+1} = \sigma\Big(\sum_{j \in \mathcal{N}_i} W^\ell h_j^\ell \Big)$$

  - Edge feature/no edge representation : Anisotropic models (like GAT[2])

$$h_i^{\ell+1} = \sigma\Big(\sum_{j \in \mathcal{N}_i} f_{V^\ell}(h_i^\ell, h_j^\ell) \cdot W^\ell h_j^\ell \Big)$$

  - Edge feature and edge representation : Anisotropic models (like GatedGCNs[3])

$$h_i^{\ell+1} = \sigma\Big(\sum_{j \in \mathcal{N}_i} e_{ij}^\ell \cdot W^\ell h_j^\ell \Big), \; e_{ij}^{\ell+1} = f_{V^\ell}\big(h_i^\ell, h_j^\ell, e_{ij}^\ell\big),$$

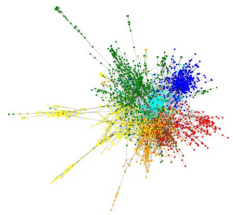
	Model	E.Feat.	E.Repr.	L	#Param	Test Acc.±s.d.	Train Acc.±s.d.	#Epochs	Epoch/Total
TSP	GatedGCN	x	x	4	99026	0.646±0.002	0.648±0.002	197.50	150.83s/8.34hr
		✓	x	4	98174	0.757±0.009	0.760±0.009	218.25	197.80s/12.06hr
		✓	✓	4	97858	0.791±0.003	0.793±0.003	159.00	218.20s/9.72hr
	GatedGCN-E	✓	✓	4	97858	**0.808±0.003**	0.811±0.003	197.00	218.51s/12.04hr
	GAT	x	x	4	95462	0.643±0.001	0.644±0.001	132.75	325.22s/12.10hr
		✓	x	4	96182	0.671±0.002	0.673±0.002	328.25	68.23s/6.25hr
		✓	✓	4	96762	0.748±0.022	0.749±0.022	93.00	462.22s/12.10hr
	GAT-E	✓	✓	4	96762	0.782±0.006	0.783±0.006	98.00	438.37s/12.11hr
COLLAB	GatedGCN	x	x	3	26593	35.989±1.549	60.586±4.251	148.00	263.62s/10.90h
		✓	x	3	26715	50.668±0.291	96.128±0.576	172.00	384.39s/18.44hr
		✓	✓	3	27055	**51.537±1.038**	96.524±1.704	188.67	376.67s/19.85hr
	GatedGCN-E	✓	✓	3	27055	47.212±2.016	85.801±0.984	156.67	377.04s/16.49hr
	GAT	x	x	3	28201	41.141±0.701	70.344±1.837	153.50	371.50s/15.97hr
		✓	x	3	28561	50.662±0.687	96.085±0.499	174.50	403.52s/19.69hr
		✓	✓	3	26676	49.674±0.105	92.665±0.719	201.00	349.19s/19.59hr
	GAT-E	✓	✓	3	26676	44.989±1.395	82.230±4.941	120.67	328.29s/11.10hr

[1] Kipf, Welling, Semi-supervised classification with graph convolutional networks, 2017
[2] Velickovic, Cucurull, Casanova, Romero, Lio, Bengio, Graph attention networks, 2017
[3] Bresson, Laurent, Residual gated graph convnets, 2017

Outline

- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- **Graph datasets**
- Conclusion

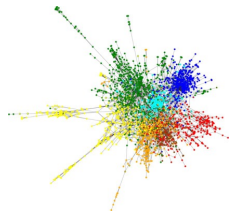
Small real graphs



- Small-scale datasets with small-size graphs
  - Cora[1] : A single graph with 2.7K nodes/5.4K edges, real graph (scientific publications), no feature, node-level classification task (7 classes)
  - Citeseer[2] : A single graph with 3.3K nodes/4.7K edges, real graph (scientific publications), no feature, node-level classification task (6 classes)
  - TU[3] : a collection of real (mostly) small-scale datasets
    - TU-ENZYMES : 600 real molecular graphs with 32 nodes/62 edges in average, 18 node features, graph-level classification task (6 classes)
    - TU-PROTEINS : 1.1K real molecular graphs with 39 nodes/72 edges in average, 29 node features, graph-level classification task (2 classes)
    - TU-DD : 1.1K real molecular graphs with 284 nodes/715 edges in average, no feature, graph-level classification task (2 classes)

[1] McCallum, Nigam, Rennie, Seymore, Automating the construction of internet portals with machine learning, 2000
Link to dataset: https://docs.dgl.ai/generated/dgl.data.CoraGraphDataset.html
[2] Getoor, Link-based classification, 2005
Link to dataset: https://docs.dgl.ai/en/0.8.x/generated/dgl.data.CiteseerGraphDataset.html
[3] Morris et-al, TUDataset: A collection of benchmark datasets for learning with graphs, 2020
Link to datasets: https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

Small real graphs



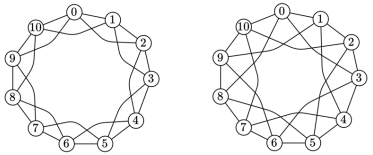
- Please, do not use small datasets as results are not consistent due to high variance[1] :

	Model	L	#Param	seed 1				seed 2			
				Test Acc.±s.d.	Train Acc.±s.d.	#Epoch	Epoch/Total	Test Acc.±s.d.	Train Acc.±s.d.	#Epoch	Epoch/Total
ENZYMES	MLP	4	101481	55.833±3.516	93.062±7.551	332.30	0.18s/0.17hr	53.833±4.717	87.854±10.765	327.80	0.19s/0.18hr
	vanilla GCN	4	103407	**65.833±4.610**	97.688±3.064	343.00	0.69s/0.67hr	64.833±7.089	93.042±4.982	334.30	0.74s/0.70hr
	GraphSage	4	105595	65.000±4.944	100.000±0.000	294.20	1.62s/1.34hr	**68.167±5.449**	100.000±0.000	287.30	1.76s/1.42hr
	MoNet	4	105307	63.000±8.090	95.229±5.864	333.70	0.53s/0.49hr	62.167±4.833	93.562±5.897	324.40	0.68s/0.62hr
	GAT	4	101274	**68.500±5.241**	100.000±0.000	299.30	0.70s/0.59hr	**68.500±4.622**	100.000±0.000	309.10	0.76s/0.66hr
	GatedGCN	4	103409	**65.667±4.899**	99.979±0.062	316.80	2.31s/2.05hr	**70.000±4.944**	99.979±0.062	313.20	2.63s/2.30hr
	GIN	4	104864	65.333±6.823	100.000±0.000	402.10	0.53s/0.61hr	67.667±5.831	100.000±0.000	404.90	0.60s/0.68hr
	RingGNN	2	103538	18.667±1.795	20.104±2.166	337.30	7.12s/6.71hr	45.333±4.522	56.792±6.081	497.50	8.05s/11.16hr
	3WLGNN	3	104658	61.000±6.799	98.875±1.571	381.80	9.22s/9.83hr	57.667±9.522	96.729±5.525	336.50	11.80s/11.09hr
DD	MLP	4	100447	72.239±3.854	73.816±1.015	371.80	6.36s/6.61hr	**72.408±3.449**	73.880±0.623	349.60	1.13s/1.11hr
	vanilla GCN	4	102293	72.758±4.083	100.000±0.000	266.70	3.56s/2.66hr	**73.168±5.000**	100.000±0.000	270.20	3.81s/2.88hr
	GraphSage	4	102577	**73.433±3.429**	100.000±0.000	267.20	11.50s/8.59hr	71.900±3.647	100.000±0.000	265.50	6.60s/4.90hr
	MoNet	4	102305	71.736±3.365	81.003±2.593	252.60	3.30s/2.34hr	71.479±2.167	81.268±2.295	253.50	2.83s/2.01hr
	GAT	4	100132	**75.900±3.824**	95.851±2.575	201.30	6.31s/3.56hr	**74.198±3.076**	96.964±1.544	220.10	2.84s/1.75hr
	GatedGCN	4	104165	**72.918±2.090**	82.796±2.242	300.70	12.05s/10.13hr	71.983±3.644	83.243±3.716	323.60	8.78s/7.93hr
	GIN	4	103046	71.910±3.873	99.851±0.136	275.70	5.28s/4.08hr	70.883±2.702	99.883±0.088	276.90	2.31s/1.79hr
	RingGNN	2	109857	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	3WLGNN	3	104124	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
PROTEINS	MLP	4	100643	75.644±2.681	79.847±1.551	244.20	0.42s/0.29hr	75.823±2.915	79.442±1.443	241.20	0.35s/0.24hr
	vanilla GCN	4	104865	76.098±2.406	81.387±2.451	350.90	1.55s/1.53hr	**75.912±3.064**	82.140±2.706	349.60	1.46s/1.42hr
	GraphSage	4	101928	75.289±2.419	85.827±0.839	245.40	3.36s/2.30hr	75.559±1.907	85.118±1.171	244.40	3.44s/2.35hr
	MoNet	4	103858	**76.452±2.898**	78.206±0.548	306.80	1.23s/1.06hr	**76.453±2.892**	78.273±0.695	289.50	1.26s/1.03hr
	GAT	4	102710	**76.277±2.410**	83.186±2.000	344.60	1.47s/1.42hr	75.557±3.443	84.253±2.348	335.10	1.51s/1.41hr
	GatedGCN	4	104855	**76.363±2.904**	79.431±0.695	293.80	5.03s/4.13hr	**76.721±3.106**	78.689±0.692	272.80	4.78s/3.64hr
	GIN	4	103854	74.117±3.357	75.351±1.267	420.90	1.02s/1.20hr	71.241±4.921	71.373±2.835	362.00	1.04s/1.06hr
	RingGNN	2	109036	67.564±7.551	67.607±4.401	150.40	28.61s/12.08hr	56.063±6.301	59.289±5.560	222.70	19.08s/11.88hr
	3WLGNN	3	105366	61.712±4.859	62.427±4.548	211.40	12.82s/7.58hr	64.682±5.877	65.034±5.253	200.40	13.05s/7.32hr

Table 16: Performance on the TU datasets with 10-fold cross validation (higher is better). Two runs of all the experiments using the same hyperparameters but different random seeds are shown separately to note the differences in ranking and variation for reproducibility. The top 3 performance scores are highlighted as **First**, **Second**, **Third**.

[1] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020

Artificial graphs

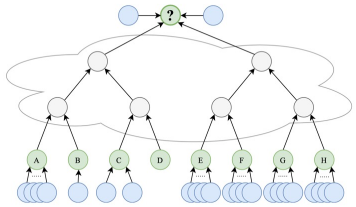


- Small-scale datasets with small-size graphs for isomorphism
  - CSL[1] : 150 artificial graphs with 41 nodes/164 edges in average, no feature, graph-level classification task (10 classes)
  - EXP[2] : 600 artificial graphs with 44 nodes/110 edges in average, no feature, graph-level classification task (2 classes)
  - SR25[3] : 15 artificial graphs with 25 nodes/300 edges in average, no feature, graph-level classification task (15 classes)
- Medium-scale datasets with medium-size SBM* graphs for pattern matching and clustering
  - PATTERN[4] : 14k artificial graphs with 117 nodes/4749 edges in average, 1 node feature, node-level classification task (2 classes)
  - CLUSTER[4] : 12k artificial graphs with 117 nodes/4301 edges in average, 1 node feature, node-level classification task (6 classes)

[1] Murphy et-al, Relational pooling for graph representations, 2019
Link to dataset: https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/gnn_benchmark_dataset.html
[2] Abboud et-al, The surprising power of graph neural networks with random node initialization, 2020
Link to dataset: https://github.com/XiaoxinHe/Graph-MLPMixer/blob/b24ae88cd5af68b06a0ccdc8866f3fbb4f432550/core/data_utils/exp.py
[3] Balcilar et-al, Breaking the limits of message passing graph neural networks, 2021
Link to dataset: https://github.com/XiaoxinHe/Graph-MLPMixer/blob/b24ae88cd5af68b06a0ccdc8866f3fbb4f432550/core/data_utils/sr25.py
[4] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020
Link to datasets: https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/gnn_benchmark_dataset.html

* SBM = Stochastic Block Models

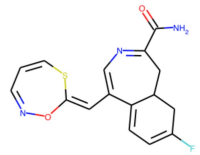
Artificial graphs



- Medium-scale datasets with medium-size graphs for graph theory
  - Cycles[1] : 20k artificial graphs with 48 nodes/87 edges in average, no feature, graph-level classification task (2 classes)
  - GraphTheoryProp[2] : 7k artificial graphs with 18 nodes/95 edges in average, no feature, collection of 6 graph-level and node-level tasks
  - TSP[3] : 12k artificial random graphs with 275 nodes/6894 edges in average, 3 node features, 1 edge feature, link-level classification task (2 classes)
- Medium-scale datasets with medium-size graphs for over-squashing
  - TreeNeighbourMatch[4] : up to 640k artificial tree-graphs with 511 nodes/510 edges in average, no feature, node-level classification task (256 classes)
- Large-scale datasets with small-size graphs
  - GraphWorld[5] : up to 1M SBM artificial graphs, up to 512 nodes, collection of graph-level, link-level node-level tasks

[1] Loukas, What graph neural networks cannot learn: depth vs width, 2020
Link to dataset: https://github.com/graphdeeplearning/benchmarking-gnns/blob/master/data/cycles.py
[2] Corso et-al, Principal neighbourhood aggregation for graph nets, 2020
Link to dataset: https://github.com/graphdeeplearning/benchmarking-gnns/blob/master/data/graphtheoryprop.py
[3] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020
Link to dataset: https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/gnn_benchmark_dataset.html
[4] Alon, Yahav, On the bottleneck of graph neural networks and its practical implications, 2020
Link to dataset: https://github.com/XiaoxinHe/Graph-MLPMixer/blob/b24ae88cd5af68b06a0ccdc8866f3fbb4f432550/core/data_utils/tree_dataset.py
[5] Palowitch et-al, Graphworld: Fake graphs bring real insights for gnns, 2022
Link to dataset: https://github.com/google-research/graphworld

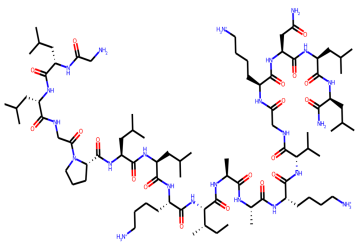
Molecular graphs



- Medium-scale datasets with small-size graphs
  - QM9[1] : 130k real molecular graphs with 9 nodes/19 edges in average, 9 node features, 2 edge features, collection of 19 graph-level regression tasks
  - ZINC[2] : 12k (and 250k) real molecular graphs with 23 nodes/24 edges in average, 1 node feature, 1 edge feature, graph-level regression task (synthetic target)
  - AQSOL[2] : 9.8k real molecular graphs with 17 nodes/35 edges in average, 1 node feature, 1 edge feature, graph-level regression task (real target)
  - OGBG-MolTOX21[3] : 7.8k real molecular graphs with 18 nodes/38 edges in average, 9 node feature, 3 edge feature, collection of 12 graph-level classification tasks (2 classes)
  - OGBG-MolHIV[3] : 41k real molecular graphs with 25 nodes/27 edges in average, 9 node feature, 3 edge feature, graph-level classification task (2 classes)
  - OGBG-PPA[3] : 158k real molecular graphs with 243 nodes/2266 edges in average, 9 node feature, 3 edge feature, graph-level classification task (37 classes)
  - OGBG-MolPCBA[3] : 437k real molecular graphs with 26 nodes/28 edges in average, 9 node feature, 3 edge feature, collection of 128 graph-level classification tasks (2 classes)

[1] Wu et-al, MoleculeNet: a benchmark for molecular machine learning, 2018
Link to dataset: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.QM9.html#torch_geometric.datasets.QM9
[2] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020
Link to ZINC dataset: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.ZINC.html#torch_geometric.datasets.ZINC
Link to AQSQL dataset: https://github.com/graphdeeplearning/benchmarking-gnns/blob/master/data/molecules/generate_AqSol_raw.ipynb
[3] Hu et-al, Open graph benchmark: Datasets for machine learning on graphs, 2020
Link to datasets and loaders: https://ogb.stanford.edu/docs/graphprop/#ogbg-mol

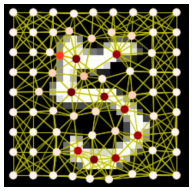
Molecular graphs



- Medium-scale datasets with medium-size graphs
  - Peptides-func[1] : 15k real molecular graphs with 150 nodes/307 edges in average, 9 node features, 3 edge feature, graph-level classification task (10 classes)
  - Peptides-struct[1] : 15k real molecular graphs with 150 nodes/307 edges in average, 9 node features, 3 edge feature, collection of 11 graph-level regression tasks
- Large-scale datasets with small-size graphs
  - OGBG-PCQM4Mv2[2] : 3.3M real molecular graphs with 14 nodes/14 edges in average, 9 node features, 3 edge features, graph-level regression task

[1] Dwivedi et-al, Long range graph benchmark, 2022
Link to Peptides-func dataset: https://github.com/XiaoxinHe/Graph-MLPMixer/blob/b24ae88cd5af68b06a0ccdc8866f3fbb4f432550/core/data_utils/peptides_functional.py
Link to Peptides-struct dataset: https://github.com/XiaoxinHe/Graph-MLPMixer/blob/b24ae88cd5af68b06a0ccdc8866f3fbb4f432550/core/data_utils/peptides_structural.py
[2] Wu et-al, MoleculeNet: a benchmark for molecular machine learning, 2018
Link to dataset: https://ogb.stanford.edu/docs/lsc/pcqm4mv2

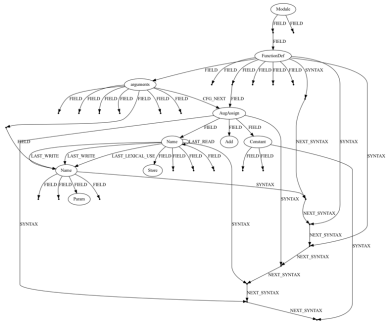
Hybrid real-artificial graphs



- Medium-scale datasets with small-size graphs for computer vision
  - MNIST[1] : 70k superpixel graphs with 70 nodes/684 edges in average, 3 node features, 1 edge feature, graph-level classification task (10 classes)
  - CIFAR[1] : 60k superpixel graphs with 117 nodes/1129 edges in average, 5 node features, 1 edge feature, graph-level classification task (10 classes)
- Medium-scale datasets with medium-size graphs for computer vision
  - PascalVOC-SP[2] : 11k superpixel graphs with 479 nodes/2710 edges in average, 14 node features, 1 edge feature, node-level classification task (20 classes)
  - COCO-SP[2] : 123k superpixel graphs with 476 nodes/2693 edges in average, 14 node features, 1 edge feature, node-level classification task (80 classes)

[1] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020
Link to datasets: https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/gnn_benchmark_dataset.html
[2] Dwivedi et-al, Long range graph benchmark, 2022
Link to datasets: https://github.com/vijaydwivedi75/lrgb

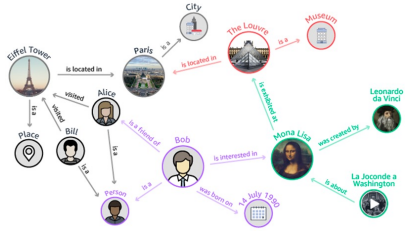
Hybrid real-artificial graphs



- Medium-scale datasets with small-size graphs for codes
  - MalNet-Tiny[1] : 5k malicious software function call graphs with 1410 nodes/2859 edges in average, no node feature, no edge feature, graph-level classification task (5 classes)
  - OGBG-code2[2] : 452k abstract syntax graphs of method definitions with 125 nodes/124 edges in average, 4 node features, no edge feature, graph-level classification task (sequence of 5 tokens)
- Large-scale datasets with large-size graphs for codes
  - MalNet[1] : 1.2M malicious software function call graphs with 15k nodes/35k edges in average, 5 node features, no edge feature, two graph-level classification tasks (47 classes and 696 classes)

[1] Freitas et-al, A large-scale database for graph representation learning, 2021
Link to dataset: https://mal-net.org
[2] Hu et-al, Open graph benchmark: Datasets for machine learning on graphs, 2020
Link to dataset: https://ogb.stanford.edu/docs/graphprop/#ogbg-code2

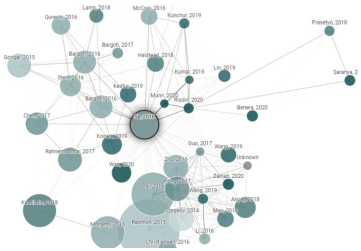
Knowledge graphs



- Single medium-scale graph
  - WikiCS[1] : a real knowledge graph with 11.7k nodes/216k edges, 300 node feature, no edge feature, node-level classification task (10 classes)
  - FB15k-237[2] : a real knowledge graph with 14.5k nodes(entities)/310k edges(triplets), 1 node feature (14.5k entity types), 1 edge feature (237 relation types), link-level ranking task
  - WN18[2] : a real knowledge graph with 40.9k nodes(entities)/142k edges(triplets), 1 node feature (40.9k entity types), 1 edge feature (18 relation types), link-level ranking task
  - OGBL-biokg[3] : a real knowledge graph with 93k nodes(entities)/5M edges(triplets), 1 node feature (5 entity types), 1 edge feature (51 relation types), link-level ranking task
- Single large-scale graph
  - OGBL-ppa[3] : a real knowledge graph with 576k nodes(entities)/30M edges(triplets), 1 node feature (58 entity types), no edge feature, link-level ranking task
  - OGBL-wikikg2[3] : a real knowledge graph with 2.5M nodes(entities)/17M edges(triplets), 1 node feature (2.5M entity types), 1 edge feature (535 relation types), link-level ranking task

[1] Mernyei, Cangea, Wiki-cs: A wikipedia-based benchmark for graph neural networks, 2020
Link to dataset: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.WikiCS.html#torch_geometric.datasets.WikiCS
[2] Bordes et-al, Translating embeddings for modeling multi-relational data, 2013
Link to FB15k-237 dataset: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.FB15k_237.html#torch_geometric.datasets.FB15k_237
Link to WN18 dataset: https://docs.dgl.ai/en/0.9.x/generated/dgl.data.WN18Dataset.html
[3] Hu et-al, Open graph benchmark: Datasets for machine learning on graphs, 2020
Link to dataset: https://ogb.stanford.edu/docs/linkprop

Citation graphs

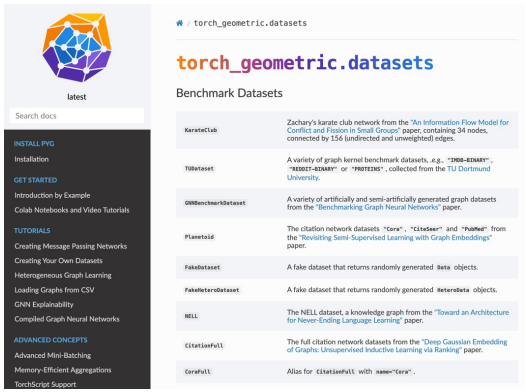


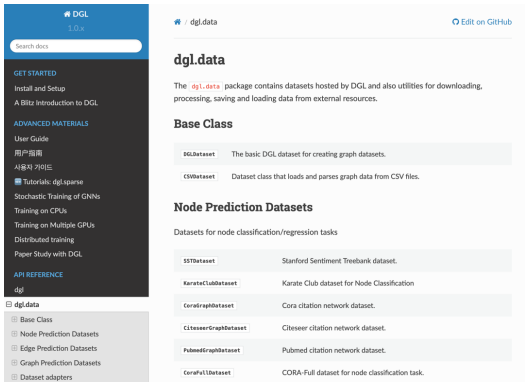
- Single medium-scale graph
  - OGBN-products[1] : a real citation graph with 2.5M nodes/61M edges, 100 node features, no edge feature, node-level classification task (47 classes)
  - OGBN-proteins[1] : a real citation graph with 132k nodes/39M edges, 1 node feature, 8 edge features, collection of 112 node-level classification tasks (2 classes)
  - OGBN-arxiv[1] : a real citation graph with 169k nodes/1.1M edges, 128 node features, no edge feature, node-level classification task (40 classes)
- Single large-scale graph
  - OGBN-papers100M[1] : a real citation graph with 111M nodes/1.6G edges, 128 node features, no edge feature, node-level classification task (172 classes)

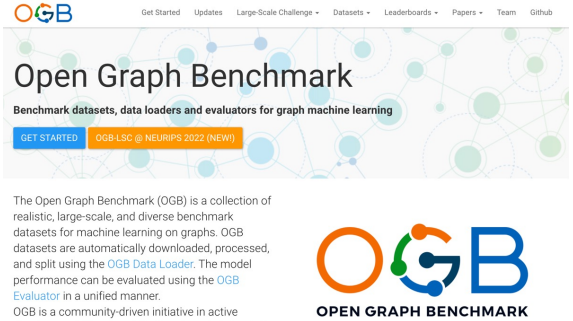
[1] Hu et-al, Open graph benchmark: Datasets for machine learning on graphs, 2020
Link to dataset: https://ogb.stanford.edu/docs/nodeprop

Additional graph datasets

- DGL datasets
  - https://docs.dgl.ai/api/python/dgl.data.html
- PyG dataset
  - https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html
- OGB datasets
  - https://ogb.stanford.edu

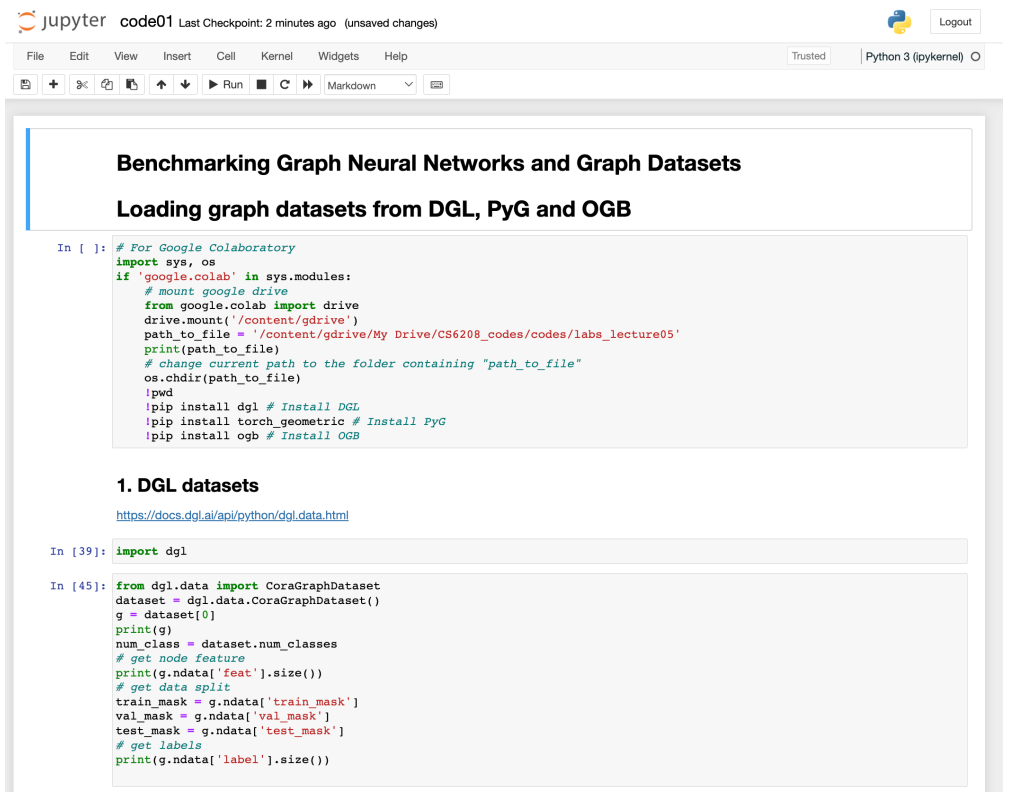






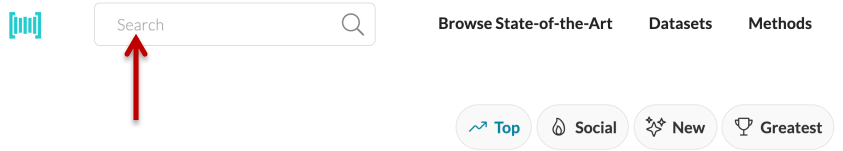
Lab 1: Loading graph datasets

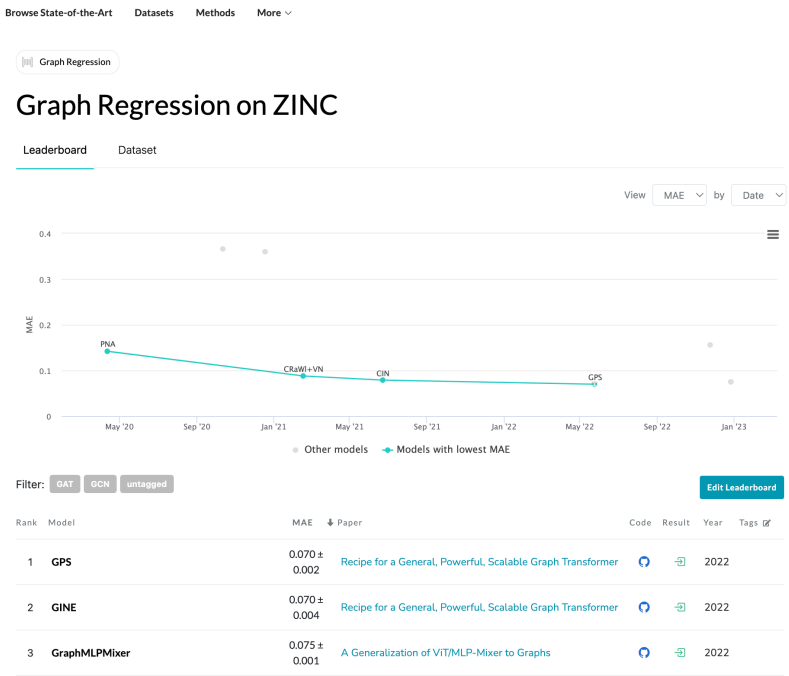
- Run code01.ipynb



Tracking SOTA per dataset

- Website https://paperswithcode.com
  - For example, ZINC graph molecular dataset





https://paperswithcode.com/dataset/zinc

Outline

- Emerging field
- Tracking progress
- Benchmarking GNNs
  - Architectures
  - Datasets
  - Code infrastructure
  - Experimental setting
  - Results and insights
- Graph datasets
- **Conclusion**

Conclusion

- Take-home messages from the benchmark[1] :
  - MP-GNNs outperformed WL-GNNs on the 8 datasets used in the benchmark.
  - MP-GNNs benefit from graph sparsity, and universal building blocks w/ batch normalization and residual connection.
  - Anisotropic mechanism improves (isotropic) MP-GNNs.
  - Explicit edge representation enhances link prediction.
- Benchmarking bridges the gap between theory and practical performance.
  - However, building good datasets is challenging (size, choice of node/edge features, noise, representativeness, GPU runtime, etc)
  - There is also a limit to learn from a dataset s.a. MNIST and ImageNet. ImageNet has provided many new architectures and insights. But over time, we "overfit" the test accuracy of these datasets with tricks and they do not contribute to new ideas.

[1] Dwivedi, Joshi, Laurent, Bengio, Bresson, Benchmarking graph neural networks, 2020

Questions?