ProGNNosis

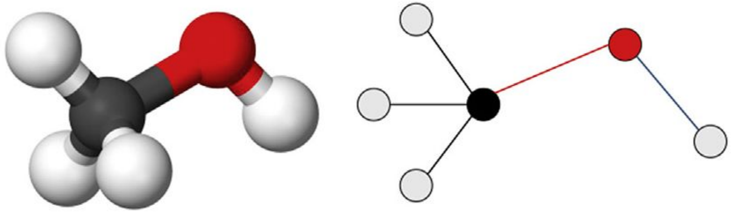
A Data-driven Model to Predict GNN Computation Time Using Graph Metrics

Authors: Axel Wassington and Sergi Abadal
Universitat Politècnica de Catalunya

UPC IDEAI
Barcelona Neural Networking Center

Motivation

GNNs are used to model **Diverse Problems**



Biology/Chemistry

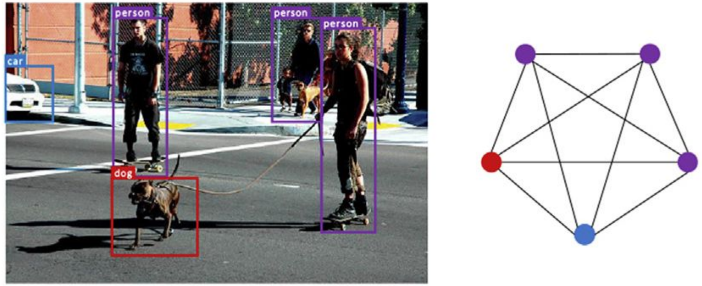
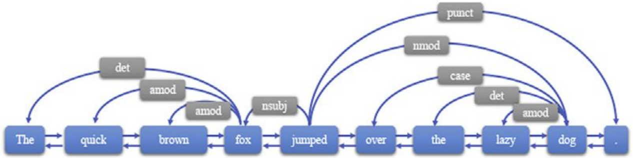
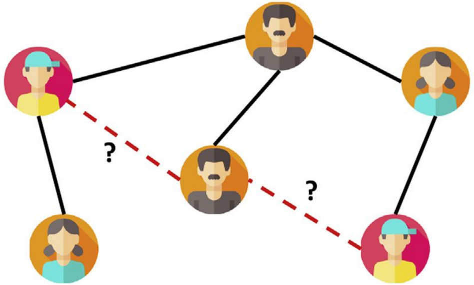


Image Comprehension



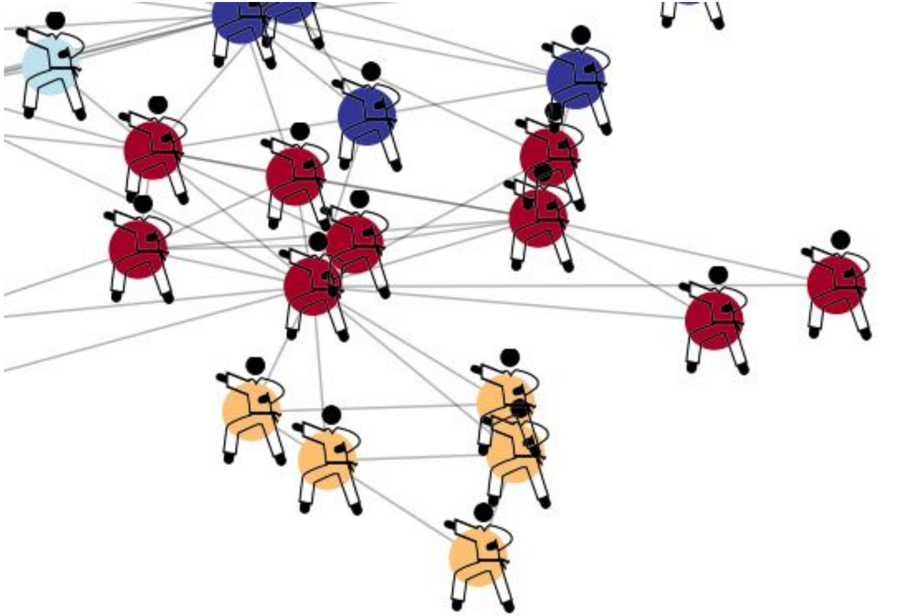
Sentence comprehension



Social relationships

Zhou, Jie, et al. "Graph neural networks: A review of methods and applications." *AI Open* 1 (2020): 57-81.

Each knowledge field features specific **Graph Metrics**

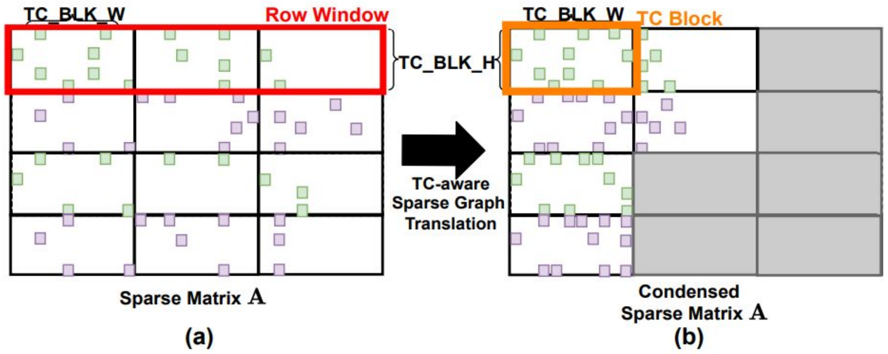


Social: High clustering coefficient

City roads: Low clustering coefficient

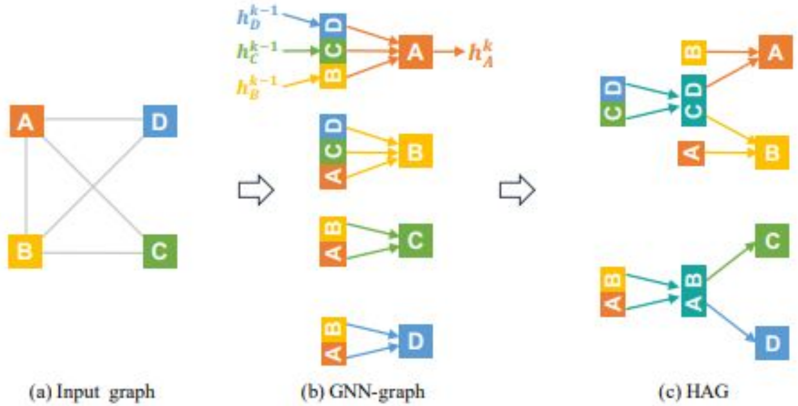
GNN Accelerators take advantage of different graph metrics

TC-GNN - Reorders to have dense blocks



Wang, Yuke, Boyuan Feng, and Yufei Ding. "TC-GNN: Accelerating Sparse Graph Neural Network Computation Via Dense Tensor Core on GPUs." *arXiv preprint arXiv:2112.02052* (2021).

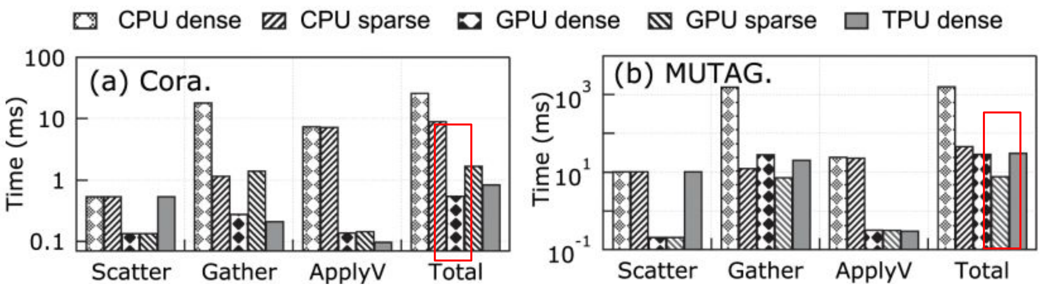
HAG - Does partial aggregation for shared neighbours



Jia, Zhihao, et al. "Redundancy-free computation for graph neural networks." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.

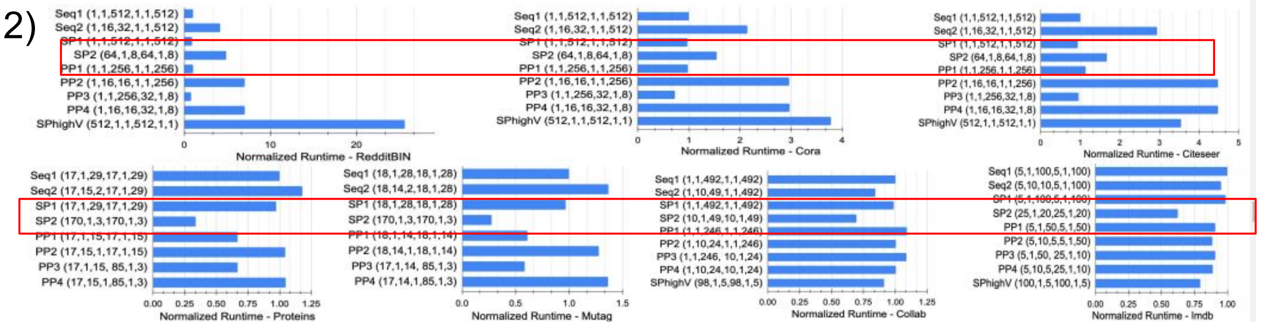
GNN accelerators are compared using **Small Sets of Graphs** (i.e. OGB)

1)



Zhang, Zhihui, et al. "Architectural implications of graph neural networks." *IEEE Computer architecture letters* 19.1 (2020): 59-62.

2)



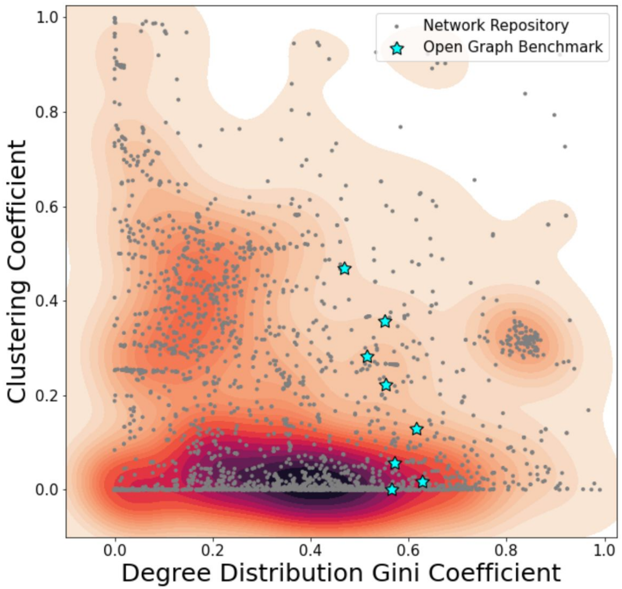
Garg, Raveesh, et al. "Understanding the Design Space of Sparse/Dense Multiphase Dataflows for Mapping Graph Neural Networks on Spatial Accelerators." Proc. IPDPS'22 (2022.)

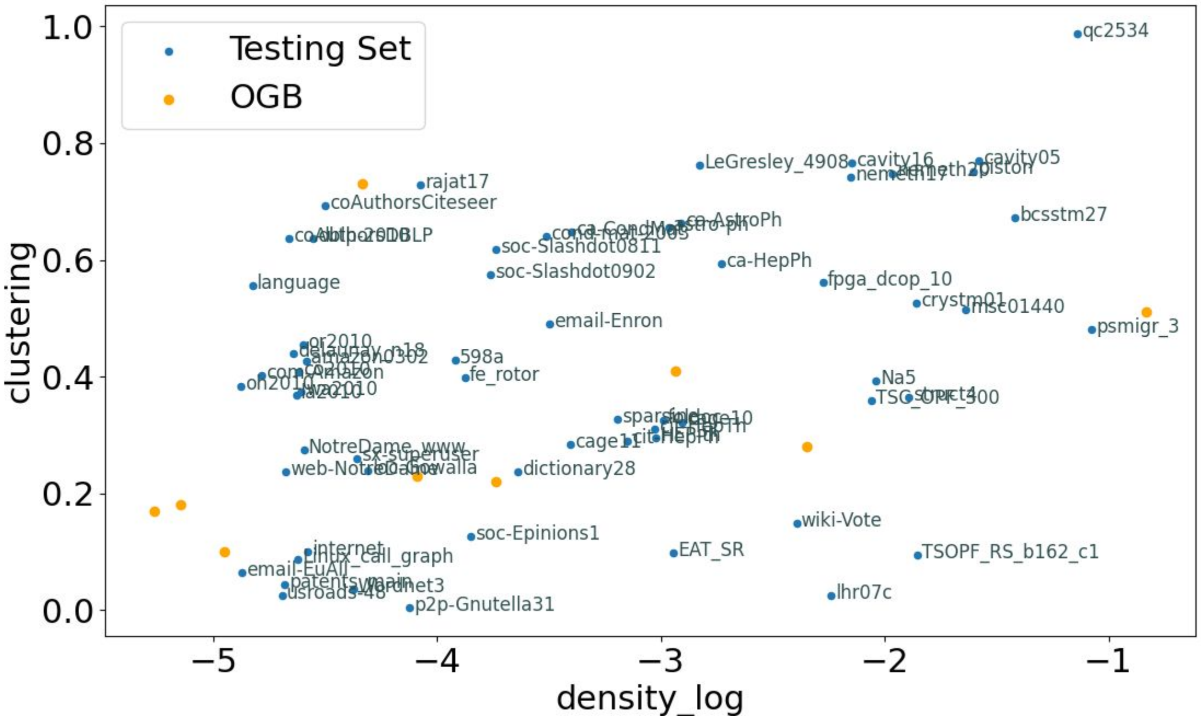
3)

Dataset	Method	DGL DB	DGL GS	PyG
Cora	GCN	4.19s	0.32s	**0.25s**
	GAT	6.31s	5.36s	**0.80s**
CiteSeer	GCN	3.78s	0.34s	**0.30s**
	GAT	5.61s	4.91s	**0.88s**
PubMed	GCN	12.91s	0.36s	**0.32s**
	GAT	18.69s	13.76s	**2.42s**
MUTAG	RGCN	18.81s	2.40s	**2.14s**

Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." Proc. ICLR Workshop Representation Learn. Graphs Manifolds(2019).

... but OGB is affected by **Selection Bias**





Palowitch, John, et al. "GraphWorld: Fake Graphs Bring Real Insights for GNNs." *arXiv preprint arXiv:2203.00112* (2022).

How do we know which is the best acceleration for a specific graph?

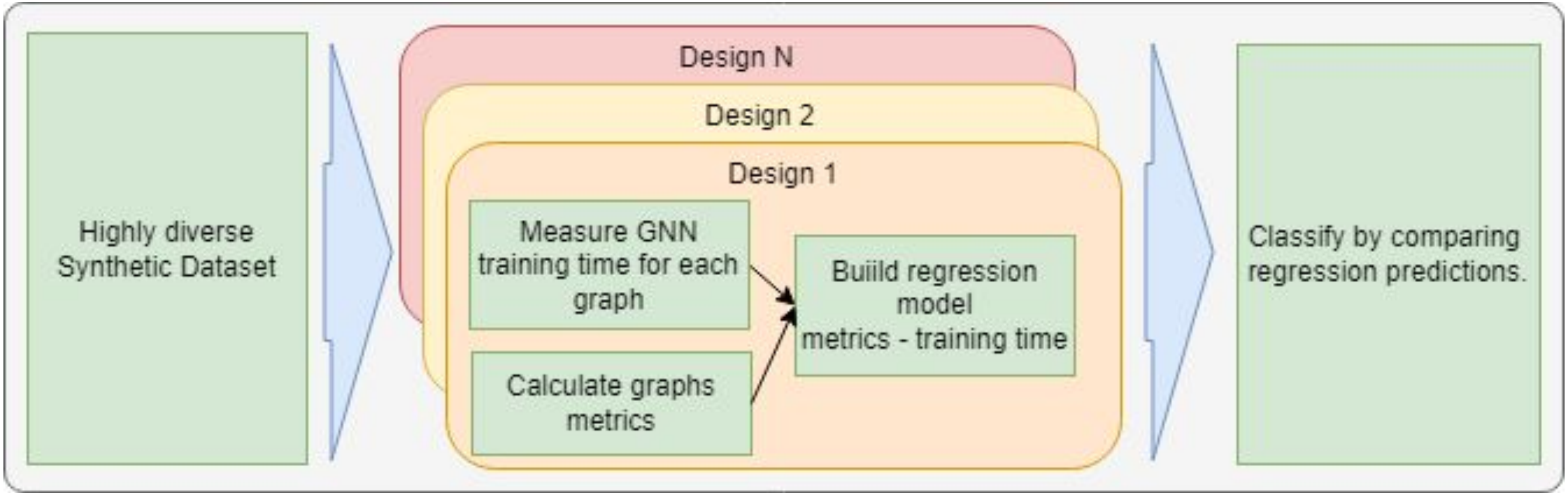
A **Naive solution** would be..

.. to try different accelerators and see which works best

- Could be a good option if the number of accelerators is low
- May need to transform the graph into different formats
- May need specialized hardware

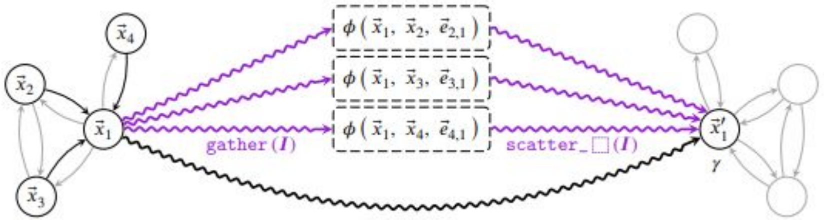
Or... We could try to predict the time each accelerator would take and make an informed decision.

With **ProGNNosis** we present a new way to automatically select the best accelerator for a given graph



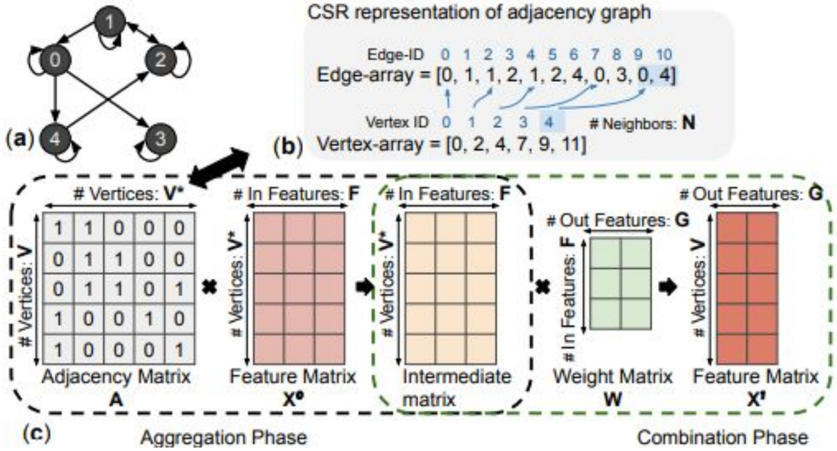
Through this **Use Case** we illustrate the utility of ProGNNosis

Design 1: Message passing (EDGE_LIST)



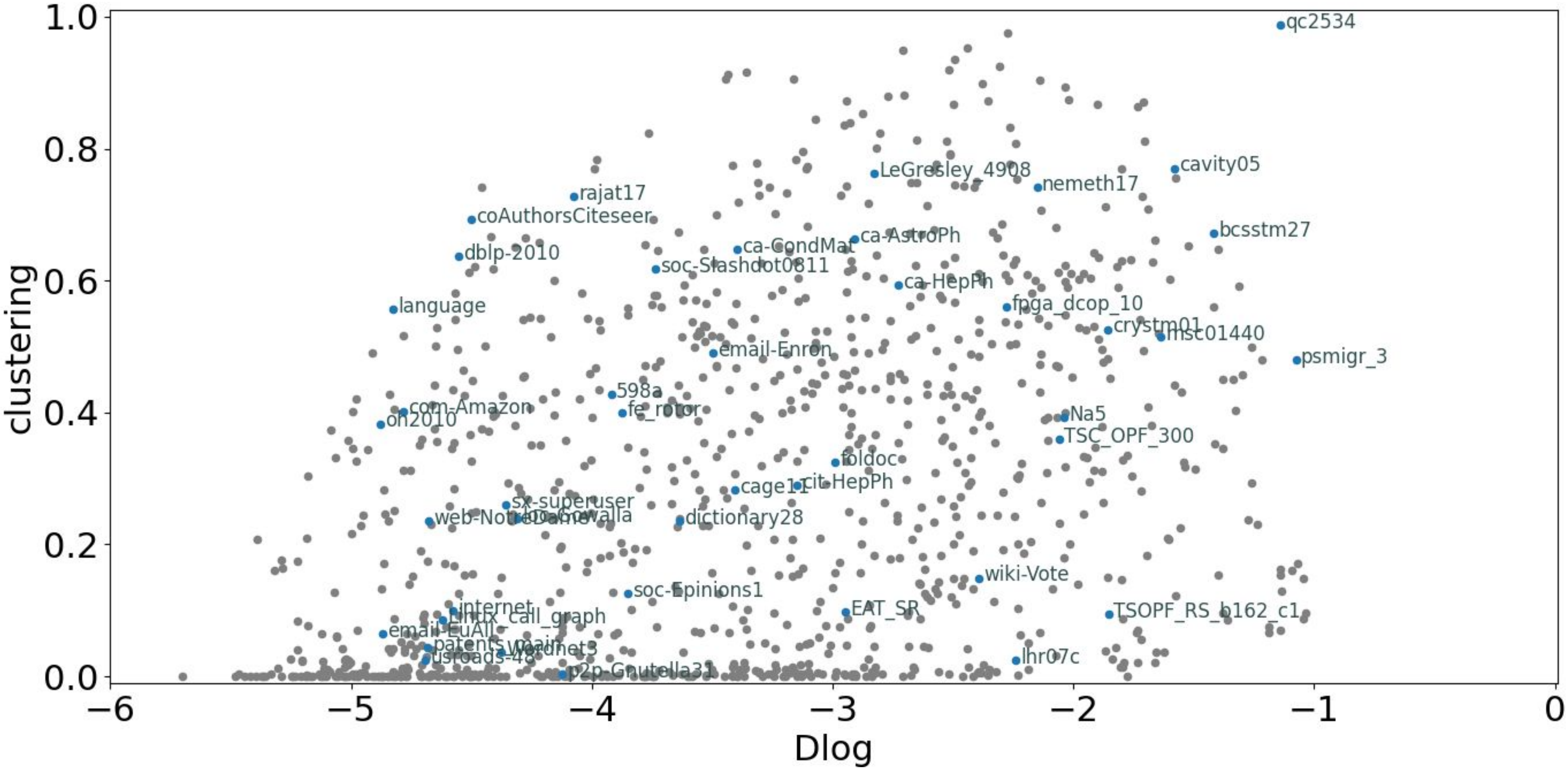
Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." Proc. ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds (2019).

Design 2: Sparse Matrix Multiplication (SPARSE)



Garg, Raveesh, et al. "Understanding the Design Space of Sparse/Dense Multiphase Dataflows for Mapping Graph Neural Networks on Spatial Accelerators." Proc. IPDPS'22 (2022.)

First we create a **Diverse dataset**



Wassington, Axel, and Sergi Abadal. "Bias Reduction via Cooperative Bargaining in Synthetic Graph Dataset Generation." *arXiv preprint arXiv:2205.13901* (2022).

We take **Measurements** for both the synthetic and real graphs

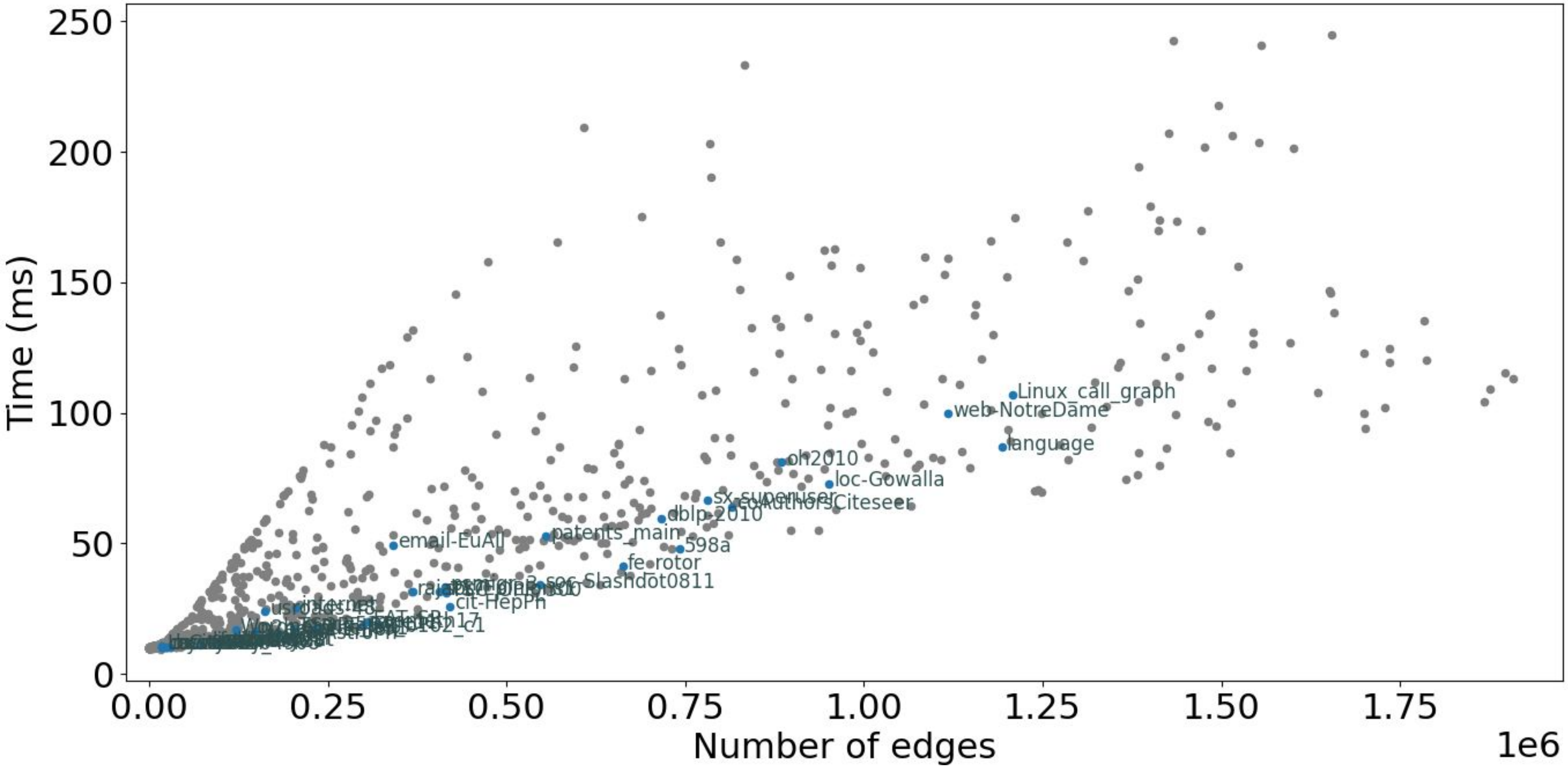
Graph metrics:

- Networkx

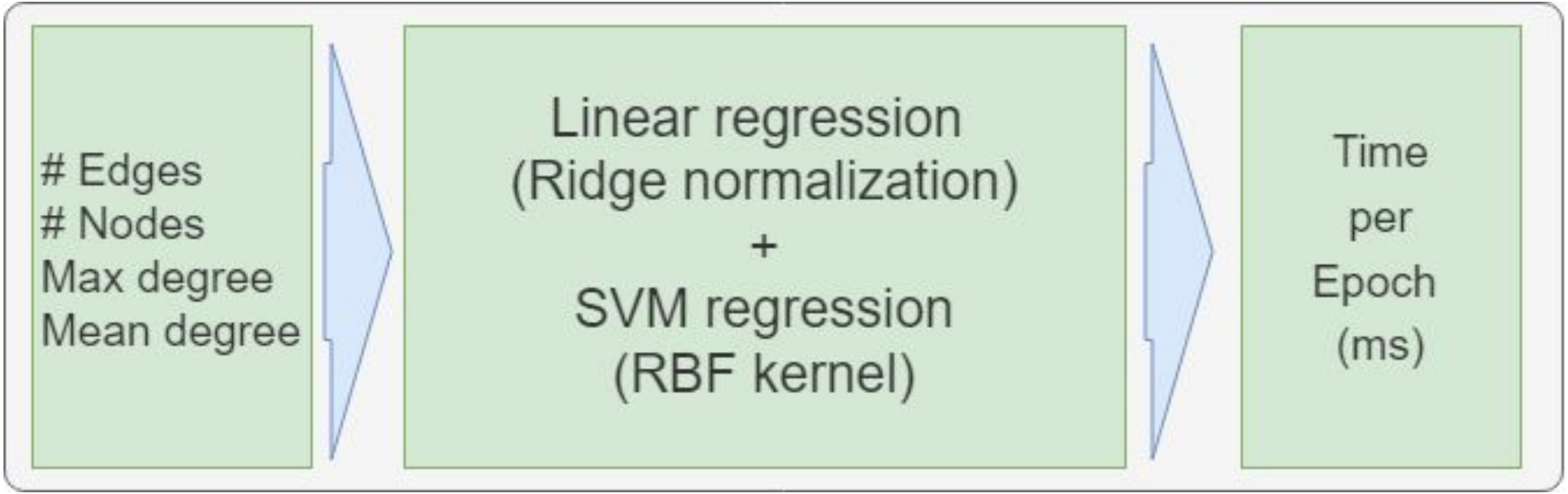
Training time:

- Hardware
  - CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
  - GPU: GeForce GTX 980 Ti
  - RAM: 15 GB
- Software
  - PyG with CUDA version 10.1 and Torch version 1.10.2
- GNN Parameters
  - Type: node classification
  - Feature vector size: 32
  - Hidden layer of size 32
  - Number of layers: 3
  - Number of classes: 2.

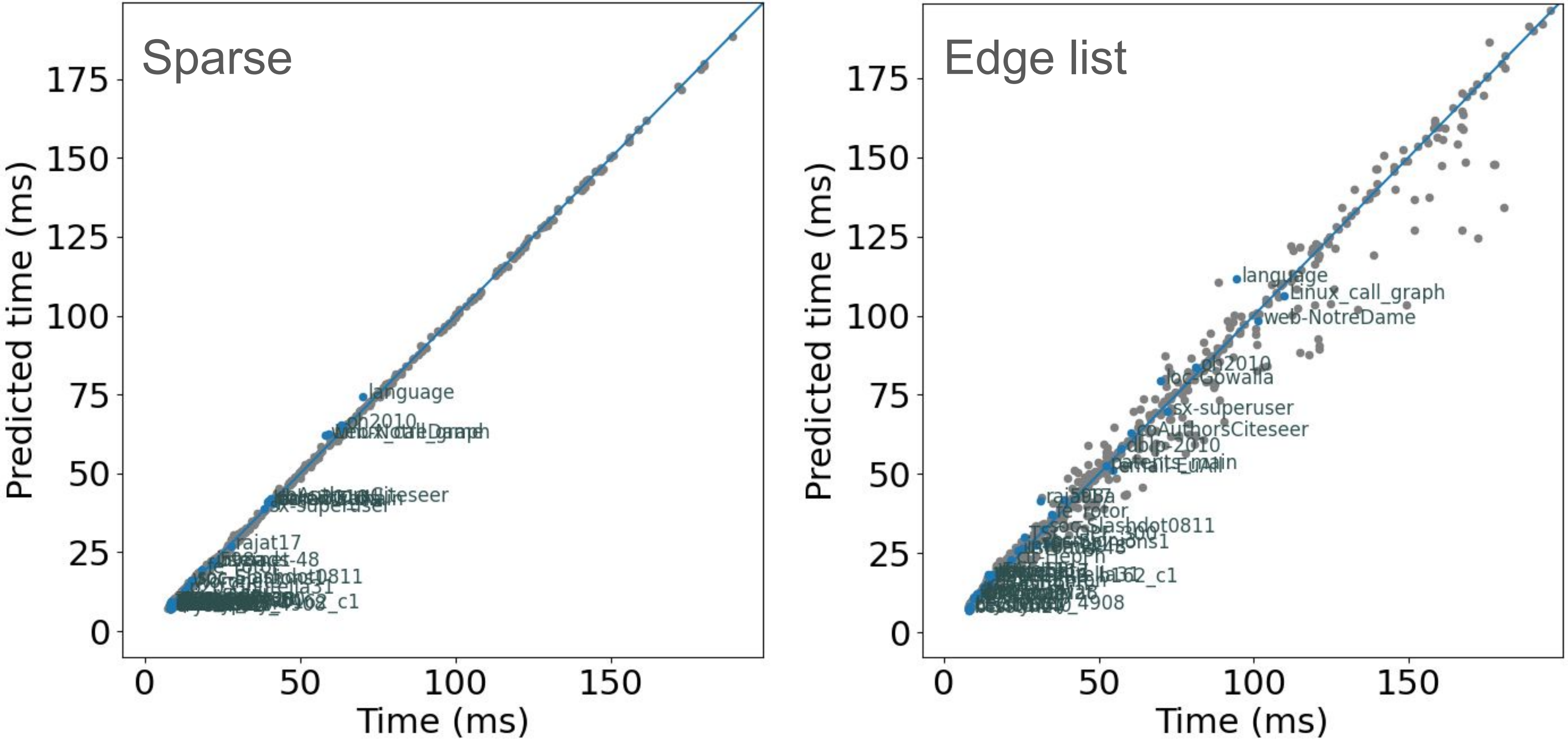
We use **Data Visualization** to understand the problem



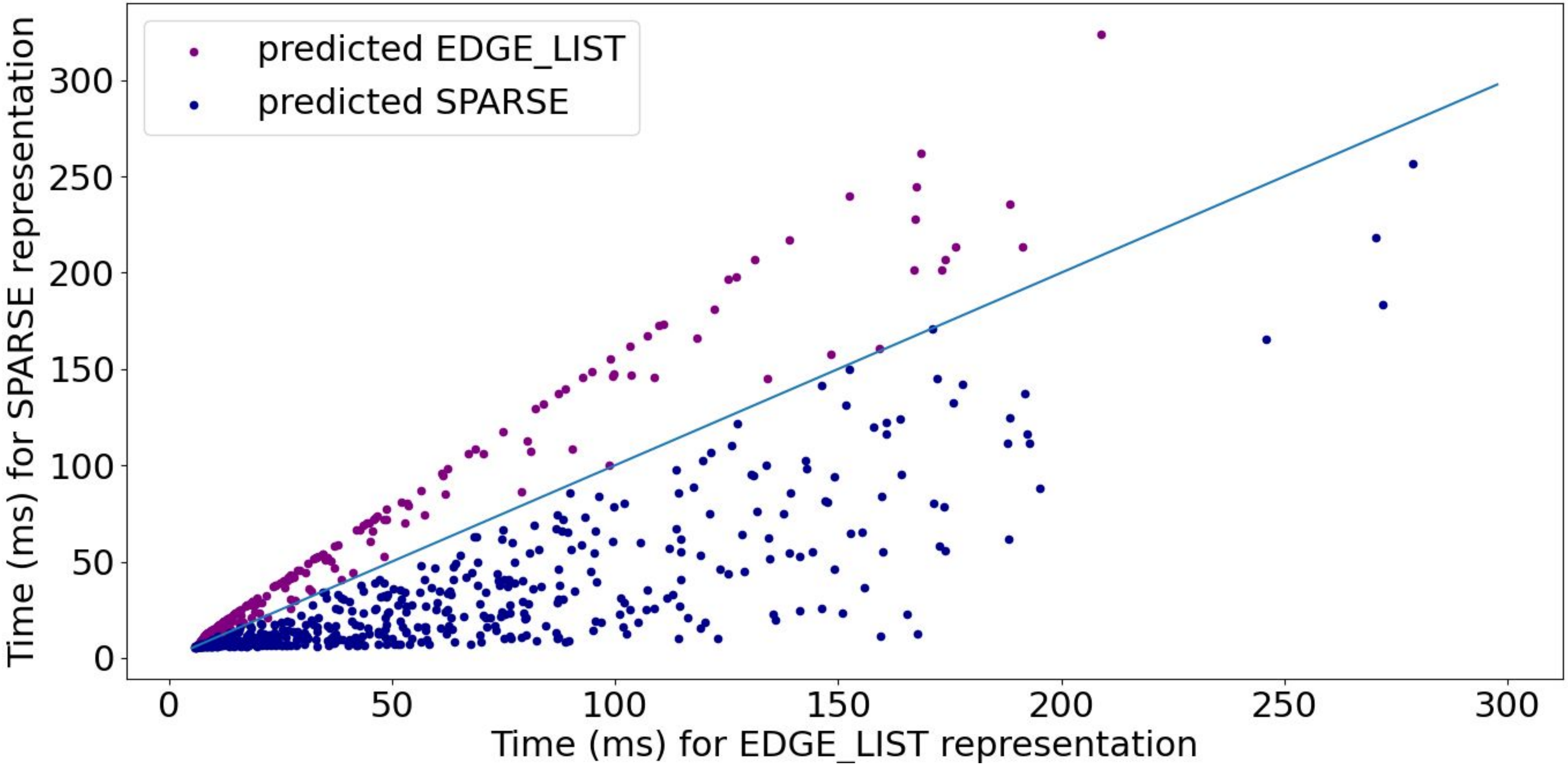
Based on what we learned we build a **Regression Model**



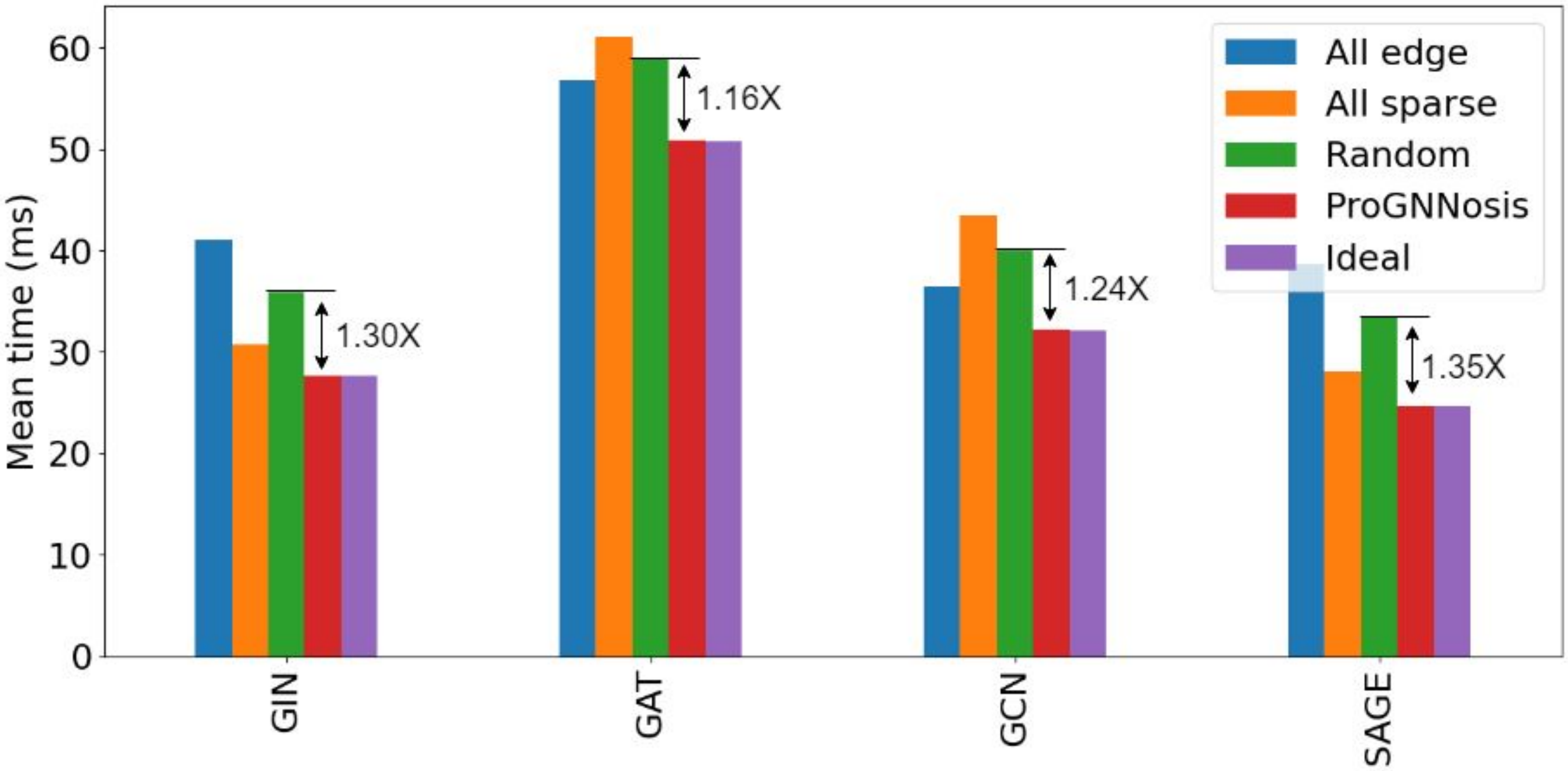
We obtain good **Regression Results** using the model



We **Classify** the graphs comparing the regression results



By applying the proposed approach we obtain **Speedups**



Future directions

- Add other features to regression (as feature vector size) and stop considering them as fixed values.
- Extrapolate results to other hardware
- Extrapolate results to other accelerators (software and hardware)

ProGNNosis

A Data-driven Model to Predict GNN Computation Time Using Graph Metrics

Authors: Axel Wassington and Sergi Abadal
Universitat Politècnica de Catalunya

UPC IDEAI

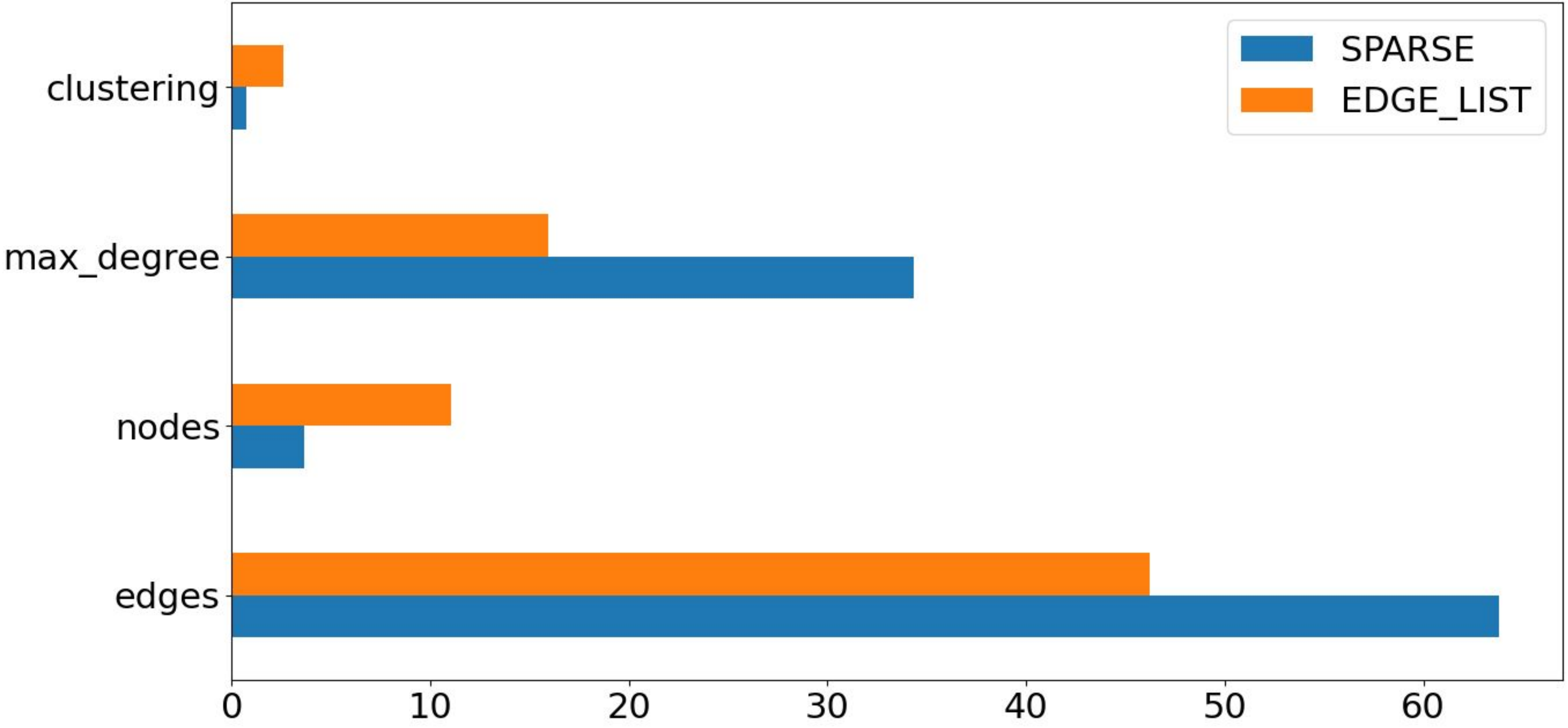
Barcelona Neural Networking Center

Backup slides

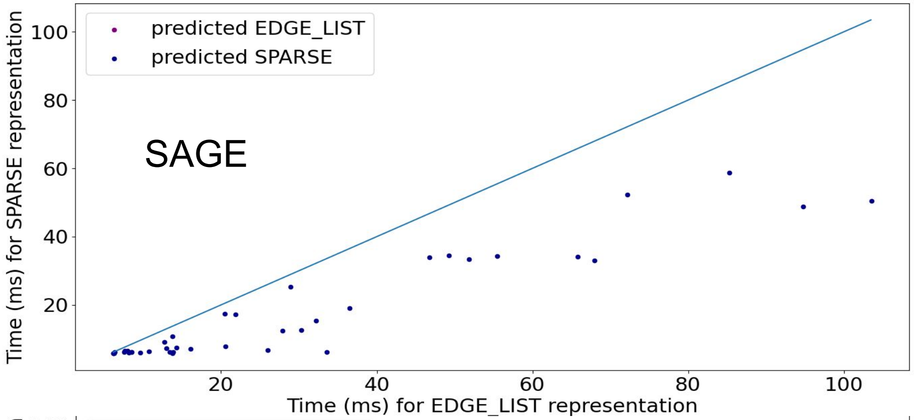
Classification Results

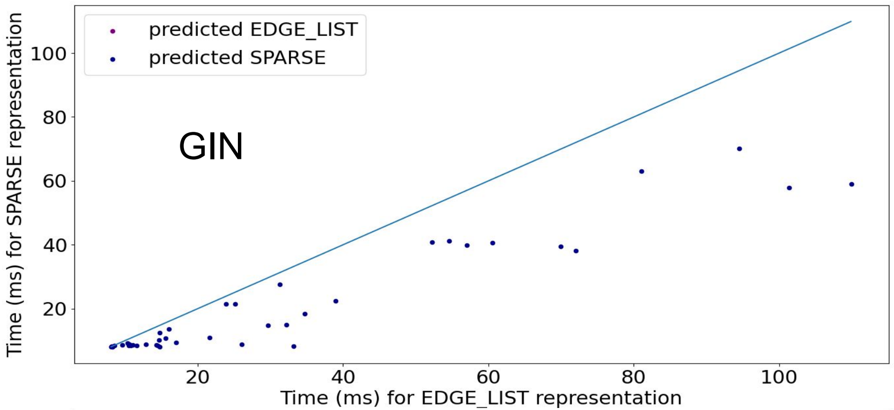
Model	Training dataset		Testing dataset	
	accuracy	**speedup**	**accuracy**	**speedup**
GCN	0.96	1.24	0.90	1.07
GIN	0.93	1.30	0.95	1.26
GAT	0.97	1.16	0.90	1.05
SAGE	0.95	1.35	0.98	1.35
Mean	0.95	1.26	0.93	1.18

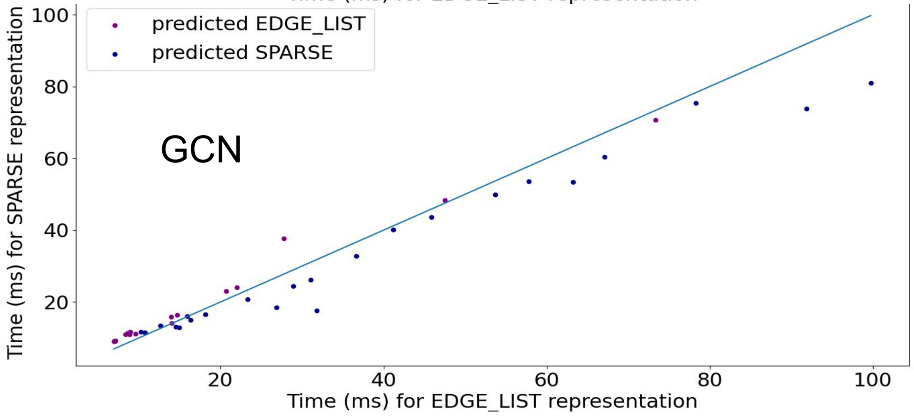
Impact factor

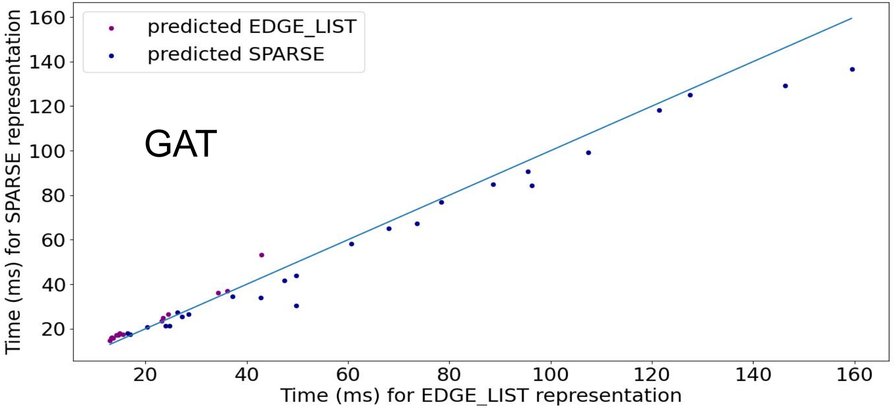


Classification testing set

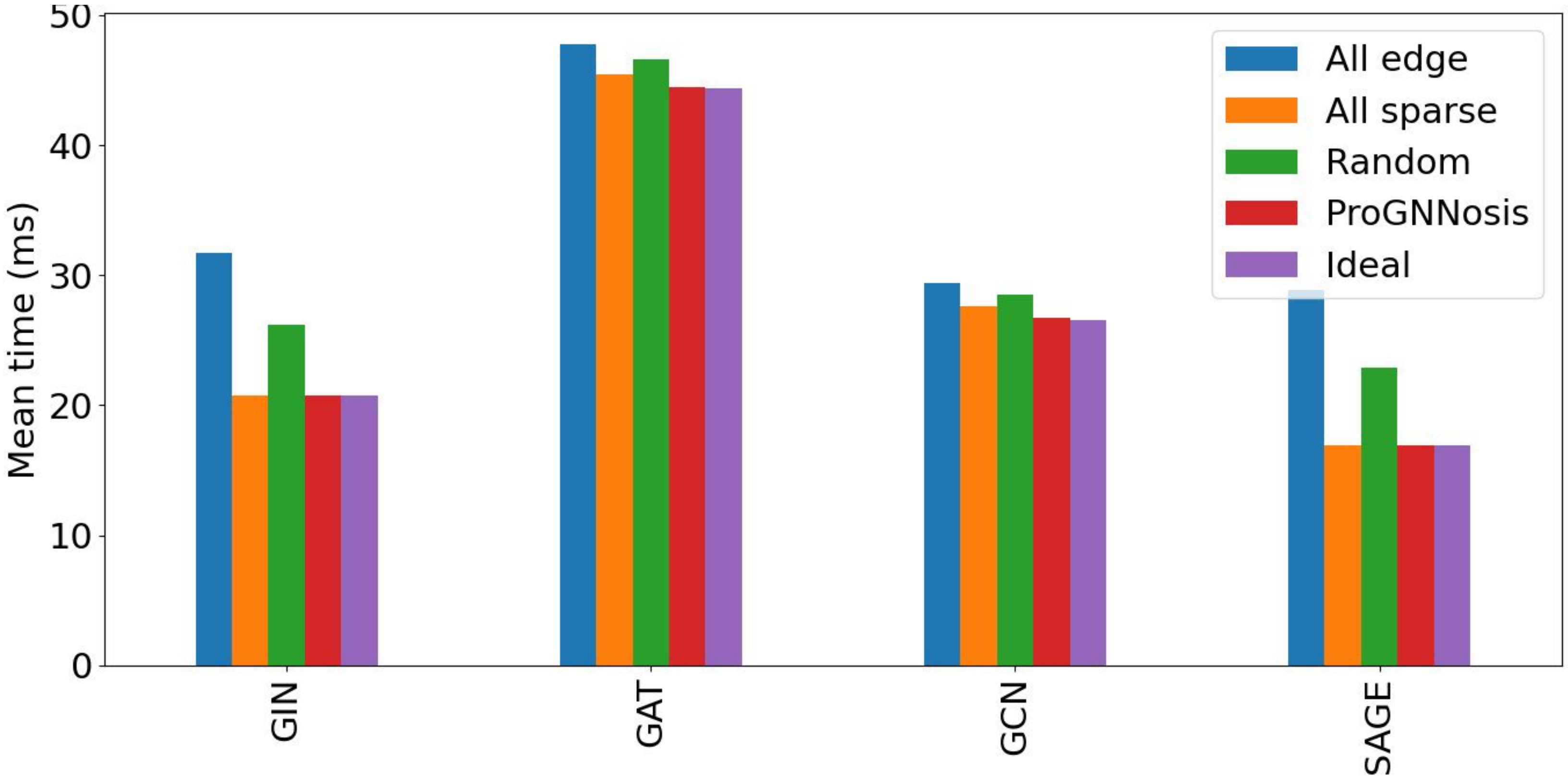








Times testing set



Metrics - Classification