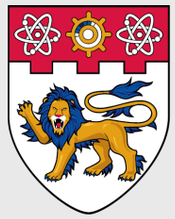
On the Experimental transferability of Spectral Graph Convolutional Networks

Master's project presentation
6/ 7/ 2020

Axel Nilsson



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

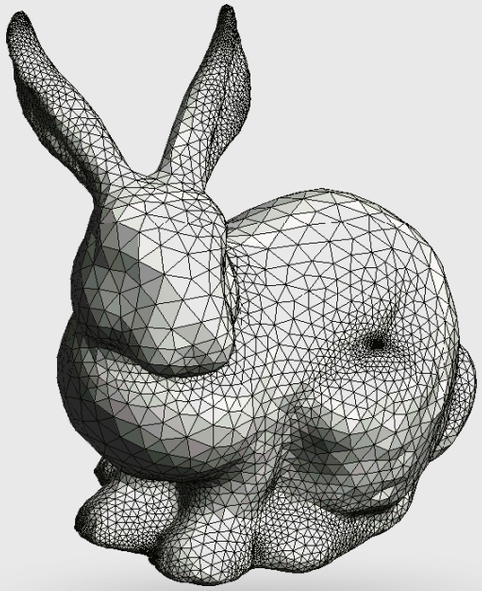
1. **Introduction**
   - Spectral graph convolutional networks
   - ChebNet

2. **Benchmarking**
   - Benchmarking GNNs
   - OGB

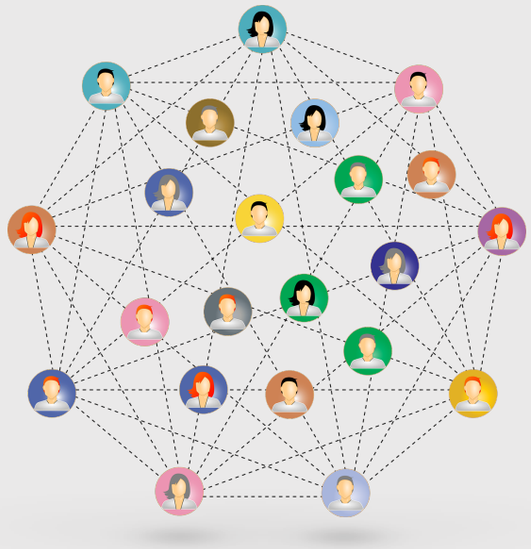
3. **Structural edge dropout**

4. **Questions (20 minutes)**

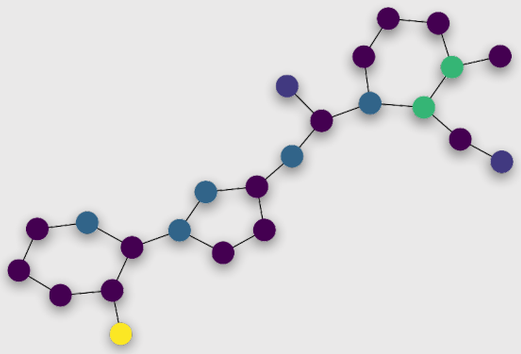
1. Introduction - Graphs



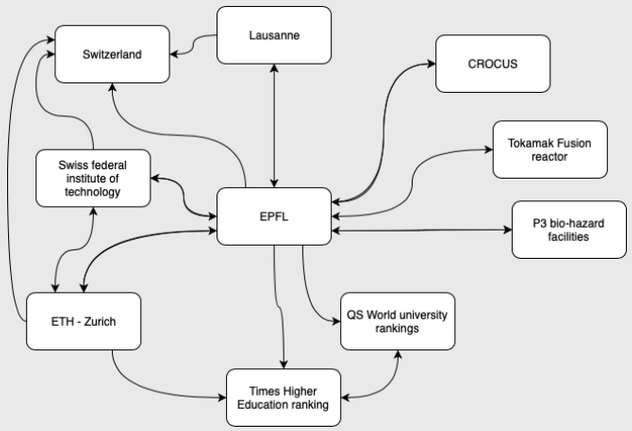
Meshes



Social networks



Molecules



Worldwide Web

G - Graph
N - Set of nodes
E - set of edges

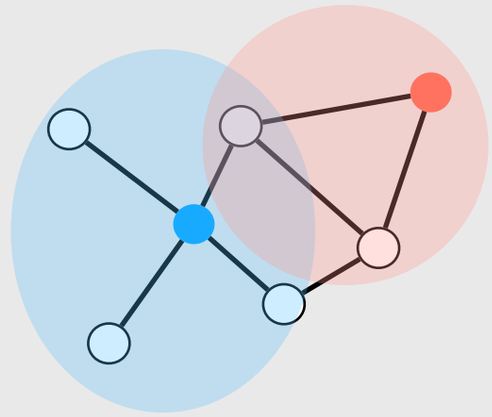
***A* - Adjacency matrix**
***D* - Degree matrix**

h **- Node features**
e **- Edge features**
g **- Graph features**

Graph Convolutional Networks (GCNs)

Convolutional neural networks do not translate well to graphs:

- No ordering of nodes
- No orientation
- Varying neighbourhood sizes



The Laplacian operator:

$$\Delta_u = D - A$$
$$\Delta_n = I_n - D^{\frac{1}{2}} A D^{\frac{1}{2}}$$

$$\Delta = \Phi^T \Lambda \Phi$$

Spectral decomposition:
n eigenvalues λ and eigenvectors Φ

Vanilla spectral GCN:

$$h^{\ell+1} = \xi\left(\Phi\hat{\theta}(\Lambda)\Phi^\top h^\ell\right)$$
$$= \xi\left(\hat{\theta}(\Delta)h^\ell\right)$$

h - node feature
xi - Non-linear activation function
theta - matrix of learnable weights
phi - eigenvectors of the laplacian

ChebNet: a fast spectral GCN

Re-normalised Laplacian:

$$\tilde{\Delta} = 2\lambda_{max}^{-1}\Delta_n - I$$

Re-scales the eigenvalues to [-1,1]

Chebyschev Polynoms:

$$\begin{cases} T_0 = h \\ T_1 = \tilde{\Delta}T_0 \\ T_{n\geq 2} = 2\tilde{\Delta}T_{n-1} - T_{n-2} \end{cases}$$

Recursively computes a basis

Learned filters:

$$g_\theta(\tilde{\Delta})h = \sum_{j=0}^{k} \theta_j T_j(\tilde{\Delta})$$

For the corresponding order *k*

- **O(1) parameter per layer**
- **Filters are localised**
- **No eigendecomposition**
- **Filters are basis dependent**

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. page 9. arXiv:1606.09375

A proof of transferability

The work of Levie et al. debunked the prejudices of the vanilla spectral GCNs

“If two graphs discretise the same continuous metric space, then a spectral GCN has approximately the same repercussion on both graphs.”

Spectral GCNs should work well on sets of graphs

Levie et al. - 2019 - Transferability of Spectral Graph Convolutional Networks

Objective

Give experimental proof of transferability of spectral GCNs on datasets with sets of graphs

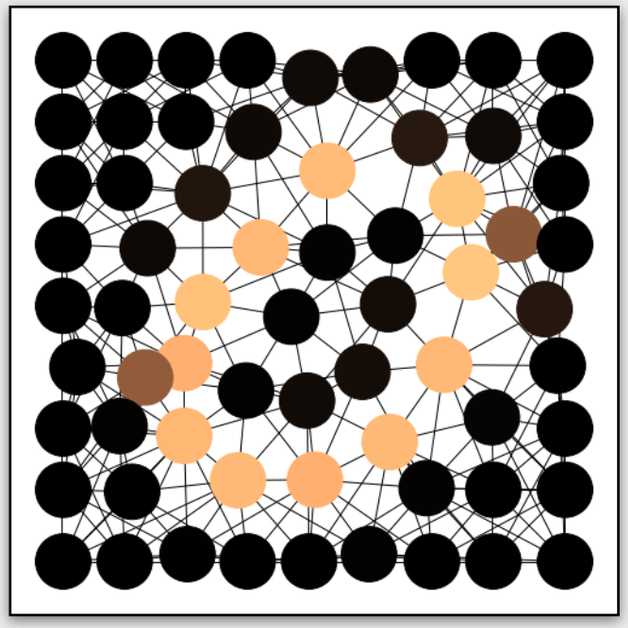
Try to improve the transferability of the spectral GCNs -> Structural Edge Dropout

2. Benchmarking

- Several benchmarks aim at comparing GCNs
- Provides a series of different tasks with large datasets
- Framework giving training hyper parameters which ensures replicability
- None include spectral GCNs!

Graph Classification
MNIST & CIFAR10 Superpixels

Dataset	Model	# parameters	Accuracy $\lambda_{max} = 2$	epoch/total
MNIST	L=4	100001	96.2625 ± 0.106 %	46 s/ 0.81 hrs
	L=16	387365	96.3125 ± 0.338 %	95 s/ 1.69 hrs
CIFAR10	L=4	100155	62.2125 ± 0.453 %	60 s/ 0.95 hrs
	L=16	387519	64.4075 ± 0.548 %	100 s/ 1.7 hrs
MNIST	GCN	101365	90.7050 ± 0.218 %	
	GraphSage	104337	97.3400 ± 0.143 %	
CIFAR10	GraphSage	104517	65.7670 ± 0.308 %	
	GatedGCN	104357	67.3120 ± 0.311 %	

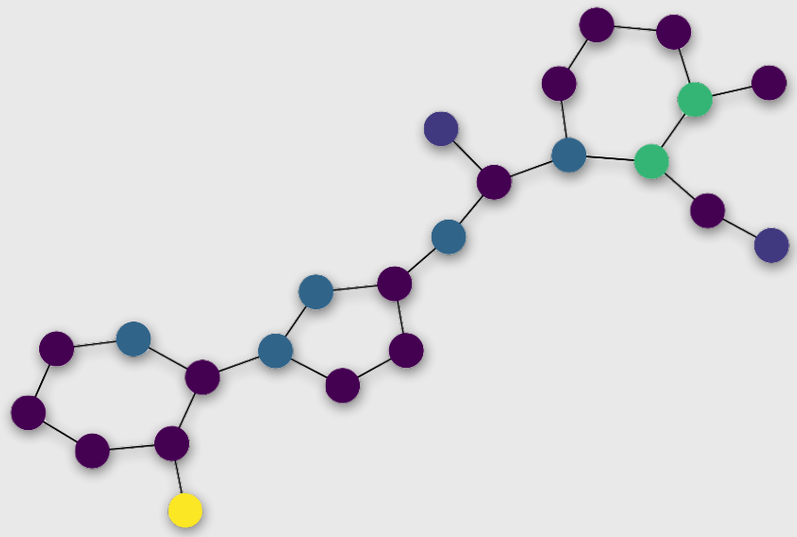


Data of the MNIST Superpixel dataset - label: 0

- **Task:** Graph classification on images to superpixel graphs with the SLIC transform.
- **Results:** Average performance on MNIST and CIFAR10 compared to similar models

Graph regression - ZINC

- **Task:** Graph regression, prediction of the solubility of each molecule
- **Result:** Best performance between models learning isotropic filters. Good performance overall.
- Questionable whether the train/val/test set are representative of any underlying space
- Unlikely that each molecule is a sample of a continuous space



Data of the ZINC, node colours are related to atom type - label: -0.2070

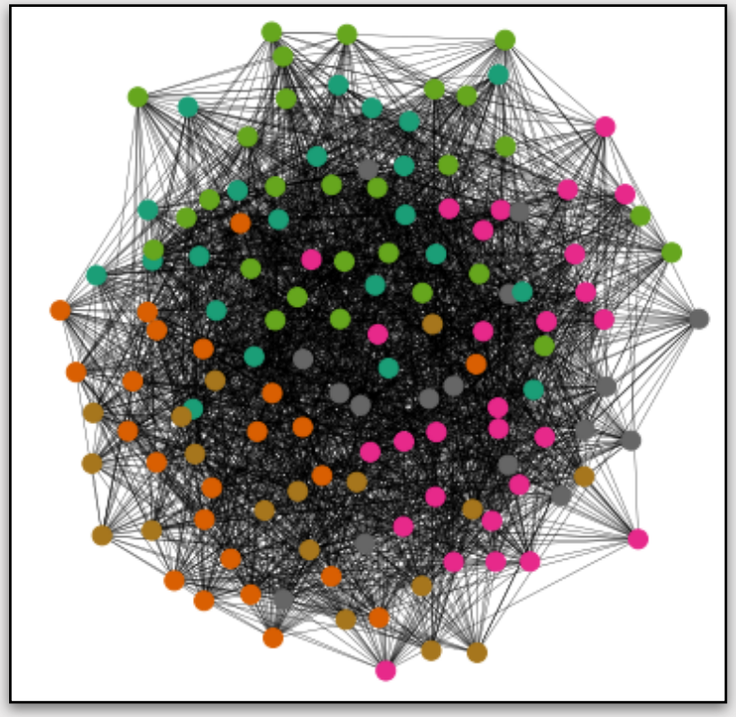
Dataset	Model	# parameters	Accuracy	Accuracy $\lambda_{max} = 2$	epoch/total
ZINC	L=4	101230	**0.3304 ± 0.0210**	0.3408 ± 0.041	135 s/ 3.5 hrs
	L=4 w rsd	101230	0.4099 ± 0.0048		171 s/ 3.2 hrs
	L=16	374710	**0.2680 ± 0.0184**		39 s/ 1.24 hrs
	L=16 w rsd	374710	0.2834 ± 0.0066		38 s/ 1.10 hrs
ZINC	GatedGCN	105735	0.4350 ± 0.011		
	GatedGCN-E-PE	505011	0.2140 ± 0.006		

Node classification - SBM

- **Task:** Predict the node label between six communities of various sizes with a probability *p* of being connected to other nodes of the community and *q* to others

- **Result:** Very good performance

- All graphs describe a non-euclidian continuous underlying manifold



Data of the SBM Cluster dataset. The colour of the nodes represent their labels.

Dataset		# parameters	Accuracy	Accuracy $\lambda_{max} = 2$	epoch/total
CLUSTER	L=4	102745	**72.8968 ± 0.197**	72.4338 ± 0.213	103 s/ 2.1 hrs
	L=4 w rsd	102745	72.7414 ± 0.211	**73.0887 ± 0.295**	103 s/ 2.1 hrs
	L=16 w rsd	399055	**74.5450 ± 0.306**		115 s/ 1.7 hrs
Gated-GCN	L = 4	104355	60.404 ± 0.419		
	L= 16	502615	73.840 ± 0.326		

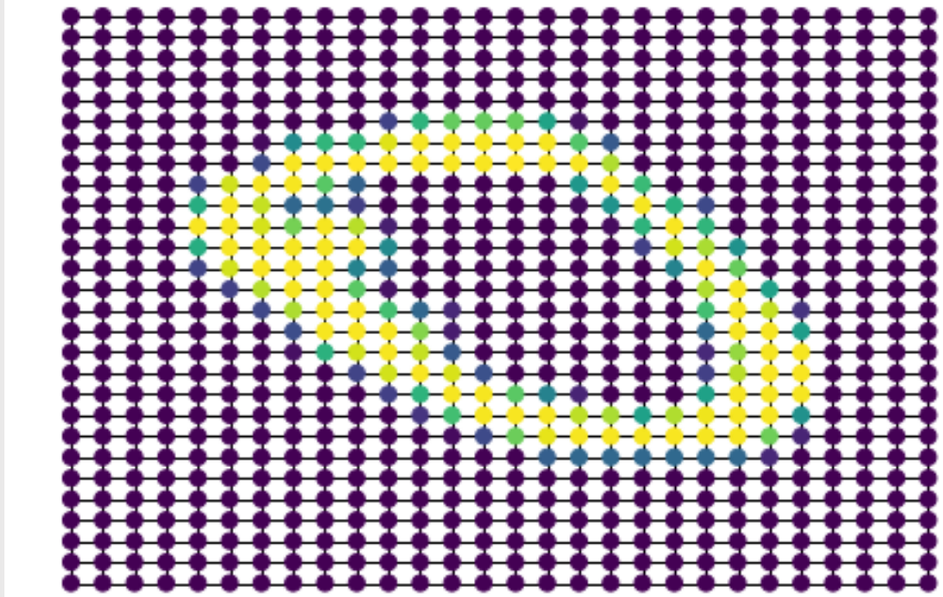
OGB: Result Summary

- **Task:** Graph regression, prediction of the proprieties of each molecule
- **Result:** Above average performance overall. Good performances with regard to classical models GCN and GIN on both tasks
  - Splitting in Test/train/val is more equitable than ZINC
  - Relatively better performance for the larger dataset
  - New models have been added to the leaderboard since the report that show greater performance

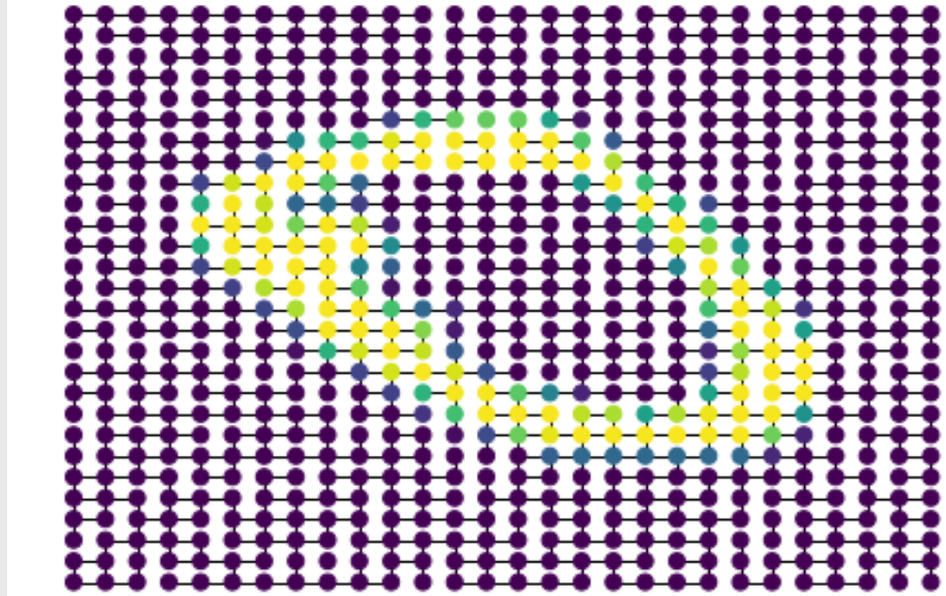
Dataset	# graphs	Metric	Accuracy (%)		
			Best Train	Val	Test
MOL-HIV	41'127	ROC-AUC	0.9992	0.8490	**0.7631± 0.0127**
MOL-PCBA	437'929	PRC-AUC	0.5417	0.2387	**0.2317 ± 0.0036**

link : https://ogb.stanford.edu/docs/leader_graphprop/

3. Structural Edge Dropout



MNIST image on a 4 - NN Lattice

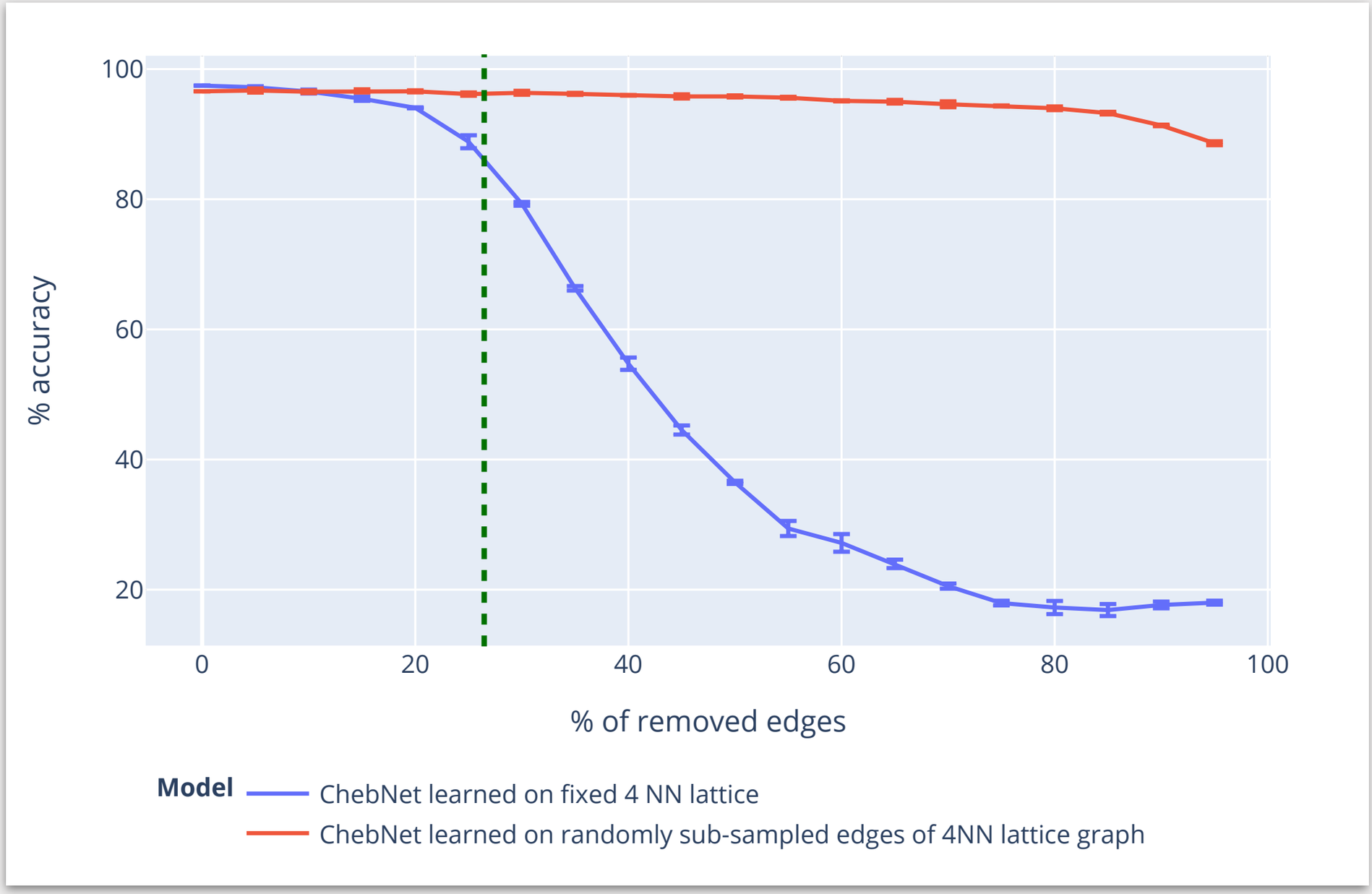


MNIST image on a 4 - NN with structural edge dropout

Structural augmentation are particular to graphs

Cut a random set of edges at a variable rate between 0 and *r* % of all the edges for every graph during the training

Structural edge dropout



- **The node features are not changed, only the graph is**
- **Shows improvement on transferability outside the region of training**

Structural edge dropout - on the benchmarking tasks

Dataset \ Edge rm. rate	0% (reference)	15%	30%
MNIST SUPERPIXEL	96.2625 ± 0.1056	96.6050 ± 0.1933	**96.6475 ± 0.1466**
CIFAR10 SUPERPIXEL	62.2125 ± 0.4526	65.9650 ± 0.6810	**66.3875 ± 0.8126**
SBM CLUSTER	72.7414 ± 0.2110	**73.1050 ± 0.1369**	72.5186 ± 0.3642

- **The performance of the ChebNet is improved in every case.**
- **Most significantly in the case of the CIFAR dataset**
- **Does not work for ZINC -> limitation of the technique**

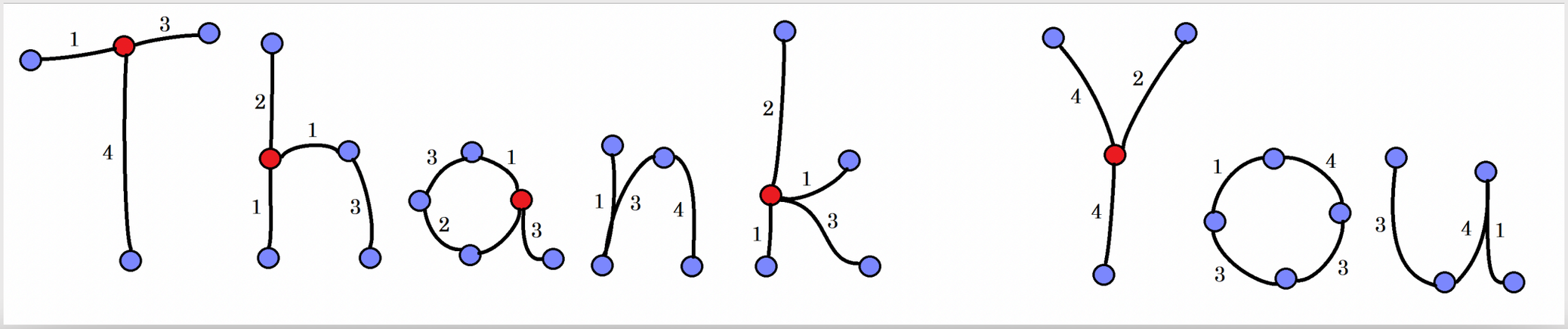
Conclusion

- The ChebNet **provide state of the art performance on *ZINC* and *CLUSTER*** of the ‘benchmarking-GNNs’ and good performances for two of OGB’s datasets

- Supports experimentally the argument that spectral GCNs have good performance and transferability

- Structural edge dropout can not only increase the performance of a spectral GCN but also its transferability

4. Questions
&



Benchmarking-GNNs: Result Summary

Dataset		# parameters	Accuracy	Acc. $\lambda_{max} = 2$	epoch/total
MNIST	L=4	100001		96.2625 ± 0.106 %	46 s/ 0.81 hrs
	L=16	387365		96.3125 ± 0.338 %	95 s/ 1.69 hrs
CIFAR10	L=4	100155		62.2125 ± 0.453 %	60 s/ 0.95 hrs
	L=16	387519		64.4075 ± 0.548 %	100 s/ 1.7 hrs
ZINC	L=4	101230	**0.3304** $\pm$ **0.0210**	0.3408 ± 0.041	135 s/ 3.5 hrs
	L=4 w rsd	101230	0.4099 ± 0.0048		171 s/ 3.2 hrs
	L=16	374710	**0.2680** $\pm$ **0.0184**		39 s/ 1.24 hrs
	L=16 w rsd	374710	0.2834 ± 0.0066		38 s/ 1.10 hrs
CLUSTER	L=4	102745	**72.8968** $\pm$ **0.197**	72.4338 ± 0.213	103 s/ 2.1 hrs
	L=4 w rsd	102745	72.7414 ± 0.211	**73.0887** $\pm$ **0.295**	103 s/ 2.1 hrs
	L=16 w rsd	399055	**74.5450** $\pm$ **0.306**		115 s/ 1.7 hrs