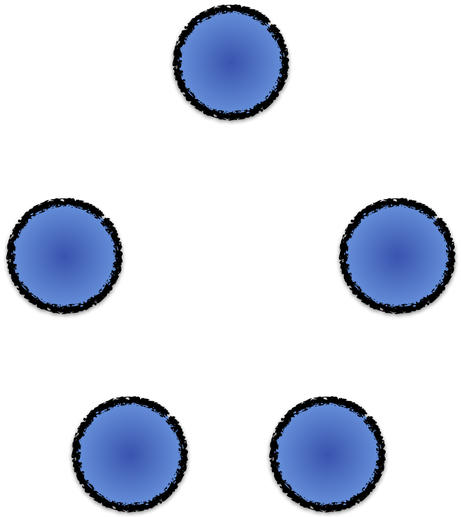
Lecture 3: Spatial Graph Convolution and Performance on Simple Random Data, Part 2

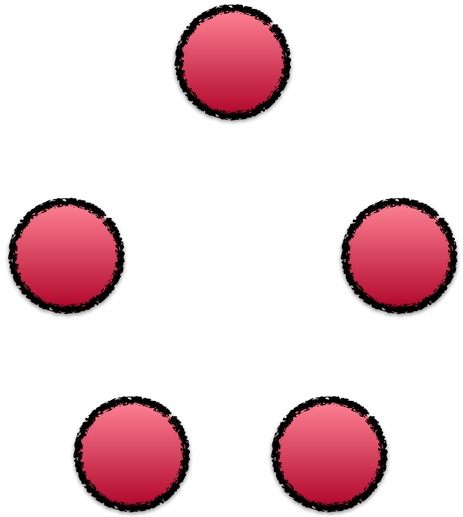
Kimon Fountoulakis

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Revision from previous lecture

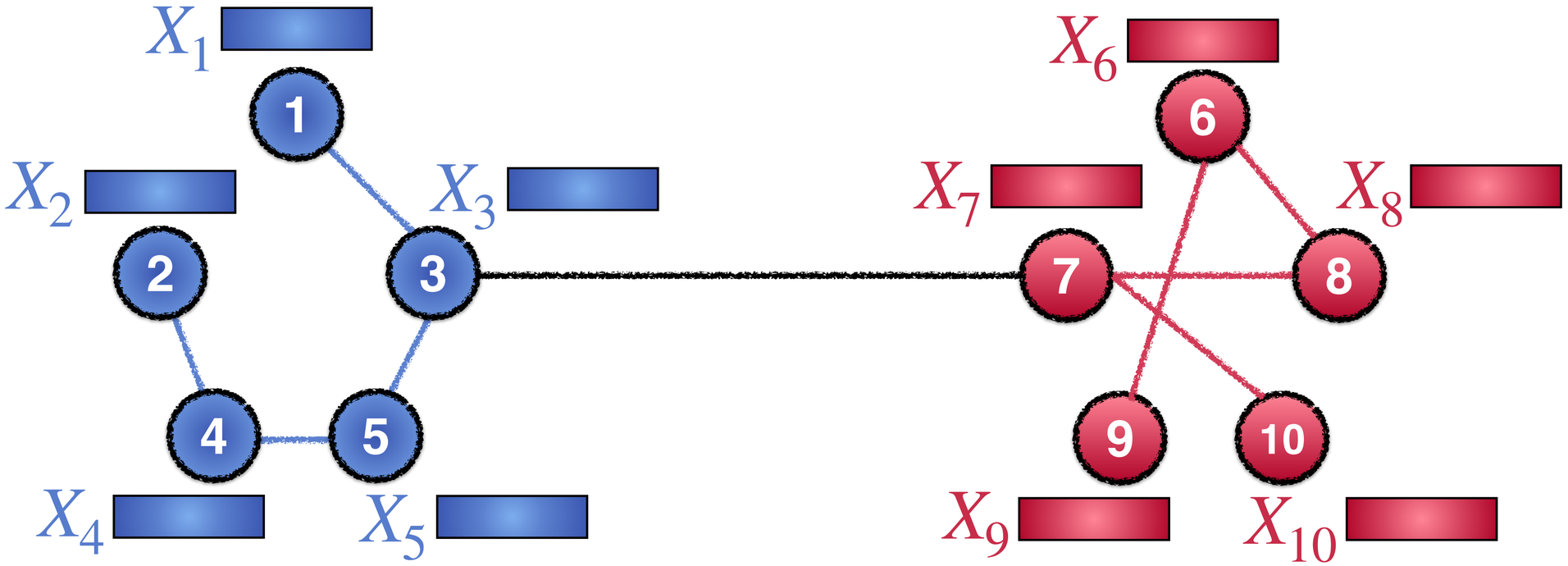
Graph data



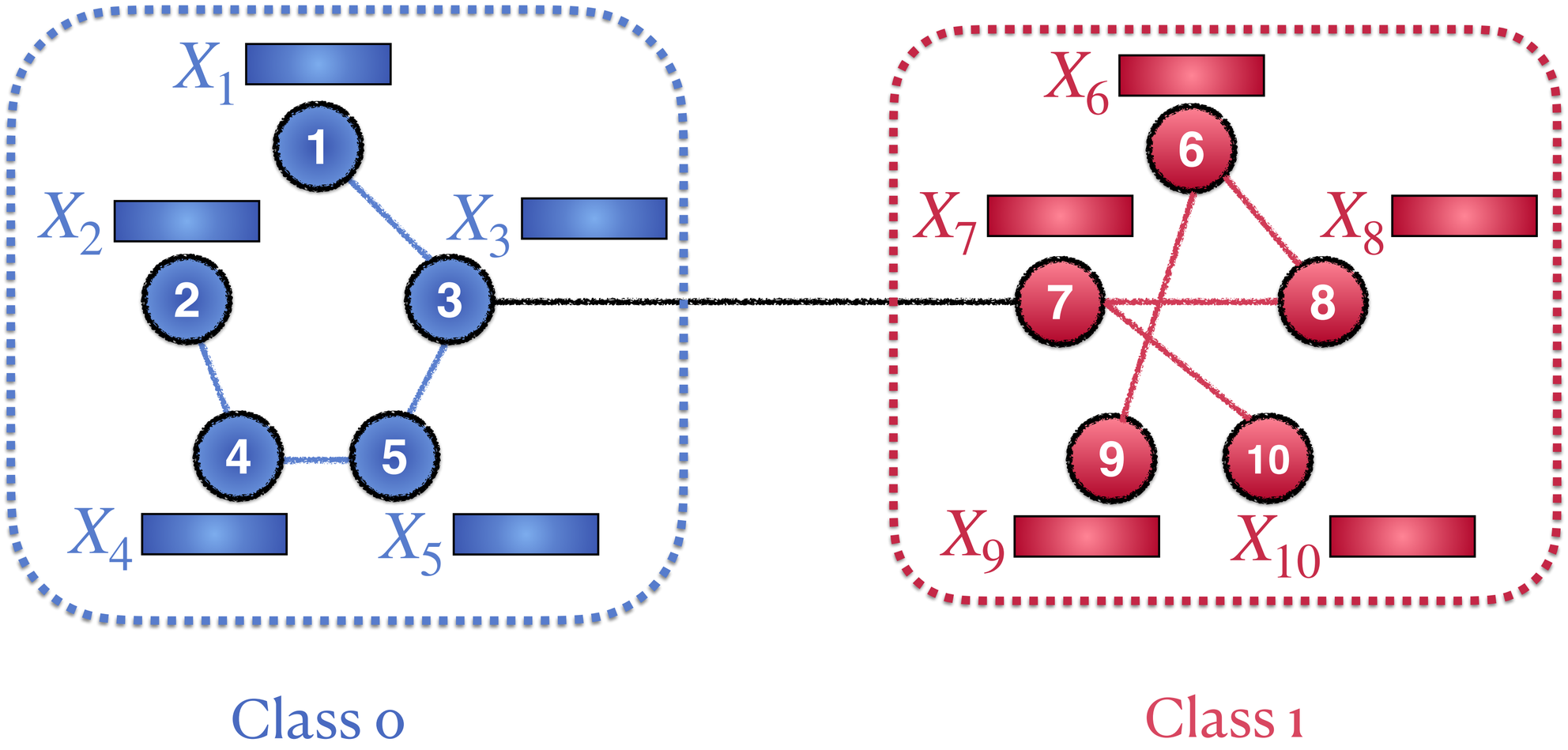


Graph data

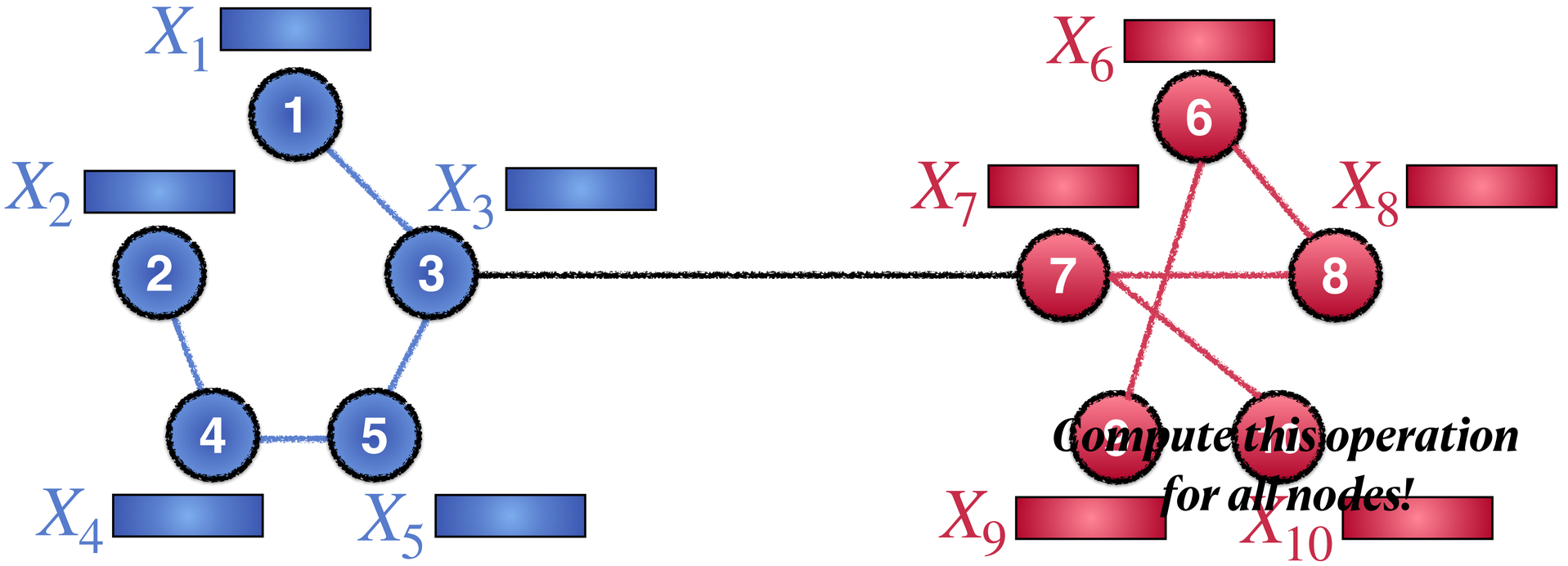
Graph data



Node classification



Graph convolution operation



$$X_3' = \frac{1}{4}(\quad + \quad + \quad + \quad)$$

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Graph convolution is a linear operator

$$X' := D^{-1} A \, X$$

Convolved data — Degree matrix — Adjacency matrix — input

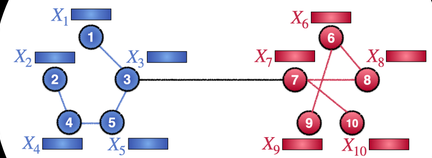
- A component of A is equal to 1 if two nodes are connected with an edge.
- D is a diagonal matrix where each component is equal to the number of neighbours of a node.

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Graph neural network

Single Layer

$$\text{Activation}\left(\text{Linear}\left(\text{Graph Convolution}\left(\sigma\left(D^{-1}AXW\right)\right)\right)\right)$$



Multi-layer networks: pass the output of the previous layer as input to the next one.

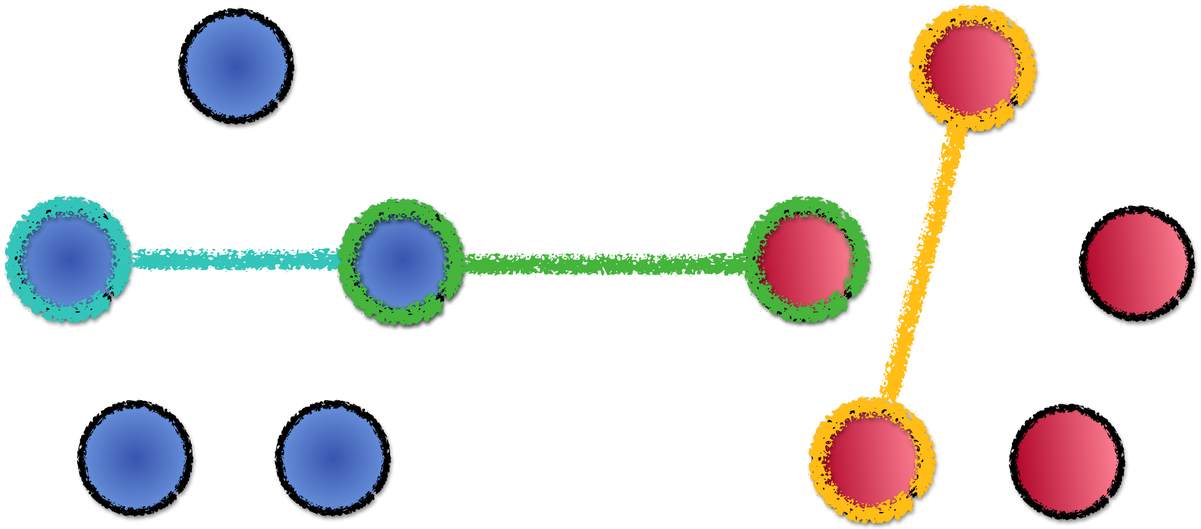
Research questions

What does graph convolution do to the new means and variance of the data?

Does graph convolution improve performance?

What about out-of-distribution performance?

Random graph: stochastic block model



Class 0 Class 1

Draw an edge in Class 0 with probability p

Draw an edge between Class 0 and Class 1 with probability q

Draw an edge in Class 1 with probability p

$SBM(p, q)$

STOCHASTIC BLOCKMODELS: FIRST STEPS *

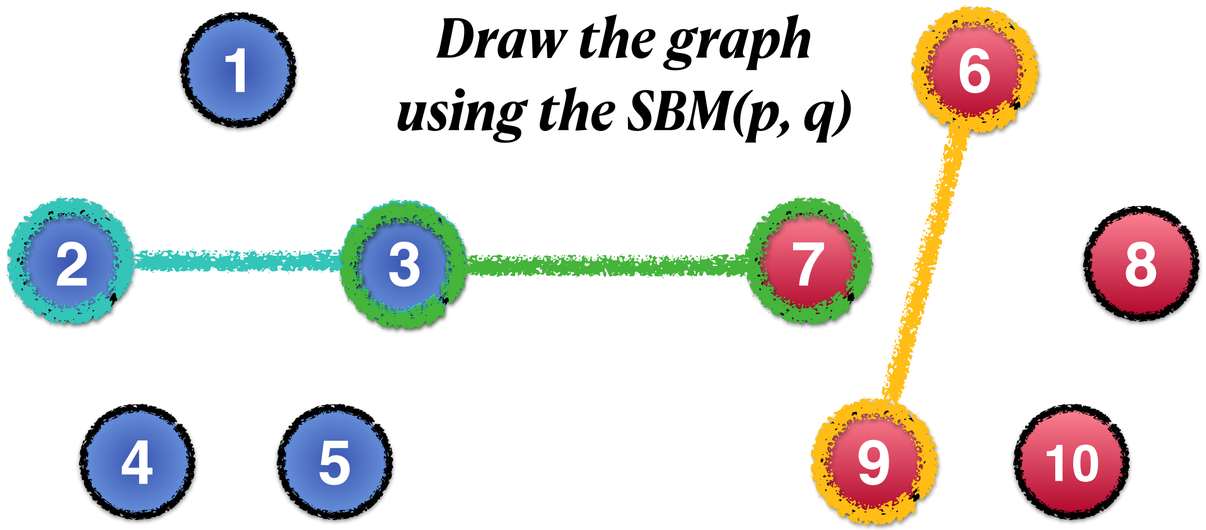
Paul W. HOLLAND

Educational Testing Service **

Kathryn Blackmond LASKEY and Samuel LEINHARDT

Carnegie-Mellon University †

Random graph + random features



Draw the graph using the SBM(p, q)

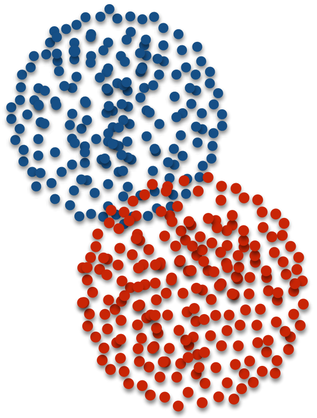
Class 0

Class 1

Draw features for each node using the Gaussian Mixture Model

$$X_i \sim \mathcal{N}(\mu, \sigma^2 I) \text{ if node } i \in \text{Class 0}$$

$$X_i \sim \mathcal{N}(\nu, \sigma^2 I) \text{ if node } i \in \text{Class 1}$$



Contextual Stochastic Block Models

Part of Advances in Neural Information Processing Systems 31 (NeurIPS 2018)

Bibtex | Metadata | Paper | Reviews | Supplemental

Authors

Yash Deshpande, Subhabrata Sen, Andrea Montanari, Elchanan Mossel

Department of Mathematics, Massachusetts Institute of Technology
Departments of Electrical Engineering and Statistics, Stanford University
Department of Mathematics, Massachusetts Institute of Technology
Department of Mathematics, Massachusetts Institute of Technology

Important quantities

For the graph

$$\frac{\mathbb{E}[|A|]}{p+q} = \frac{\text{expected number of neighbours}}{\sigma}$$

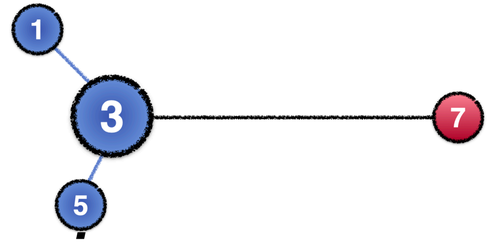
For the features

$$\frac{\|\mu - \nu\|}{\sigma}$$

“relative signal strength”

Convolution changes the mean and variance

After **1 graph convolution** the means move closer by a factor $\frac{|p-q|}{p+q}$.



But the variance is reduced by a factor $\mathcal{O}(\mathbb{E}[|\mathcal{N}|])$.

Before $\|\mu_1 - \mu_2\|$ $|p-q|$ $\|\mu_1' - \nu\|$

The variance is reduced by a factor larger than than the factor for the means.

$$X_3' = \frac{1}{4}(X_1 + X_3 + X_5 + X_7)$$

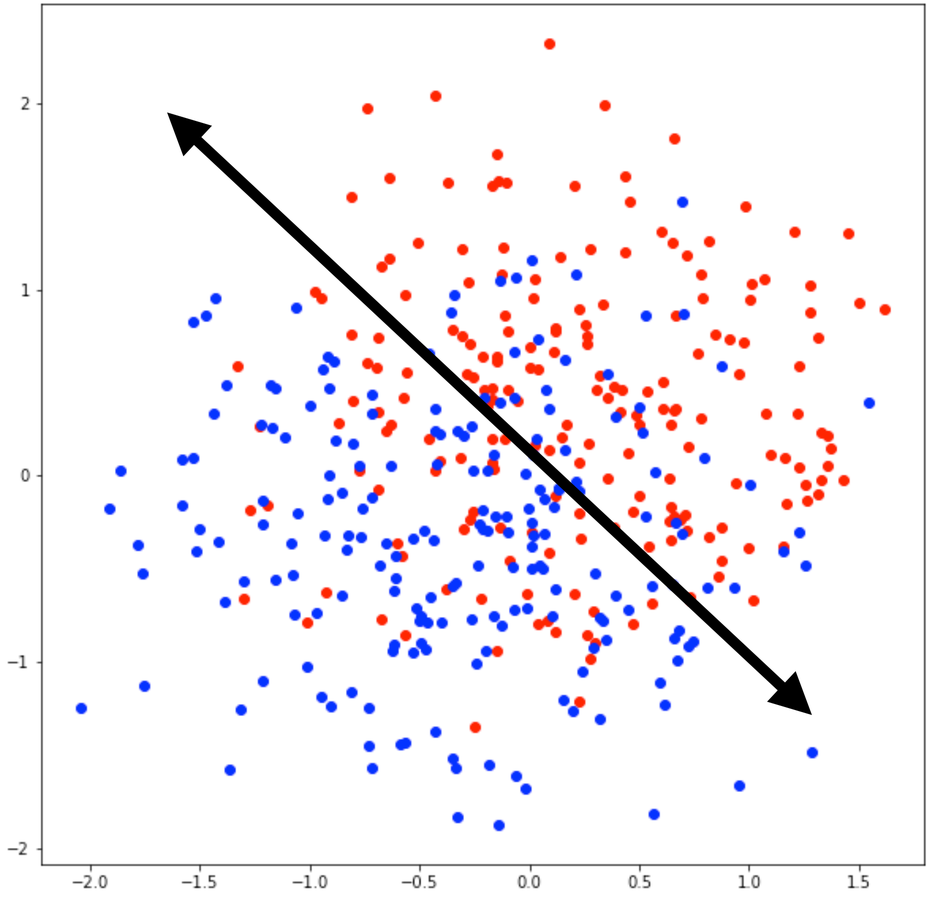
What about classification performance?

Without the graph, *no hyperplane* can separate the data if

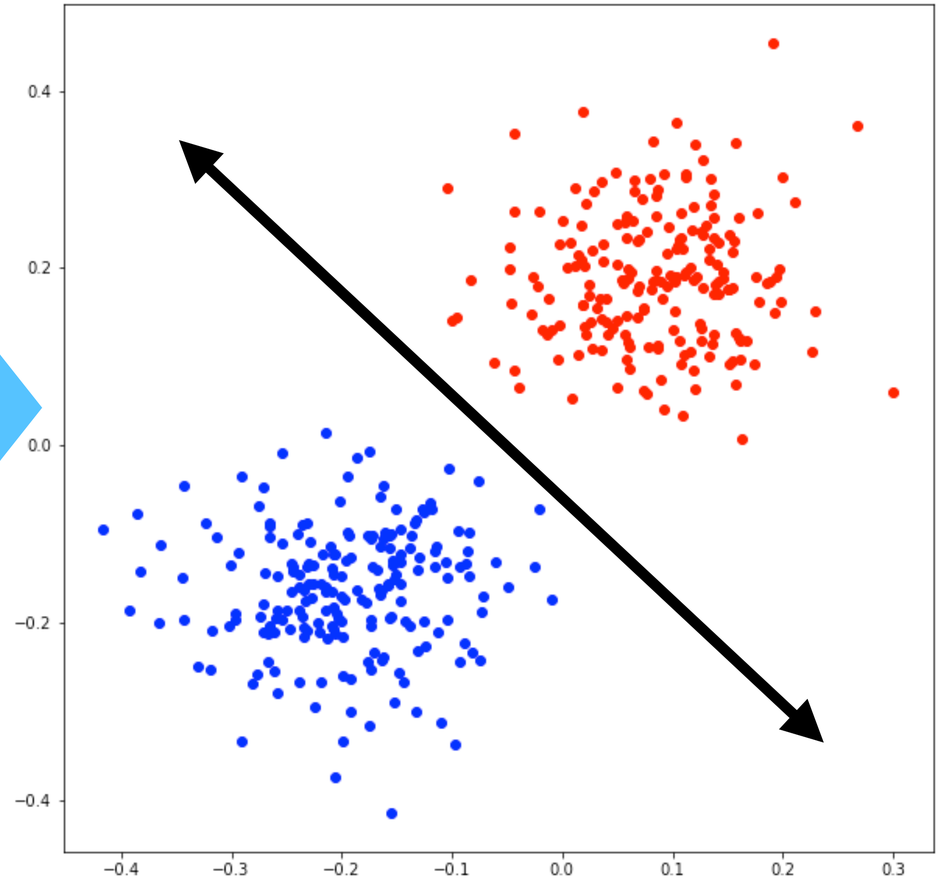
$$\frac{\|\mu - \nu\|}{\sigma} = \mathcal{O}(\sqrt{\log n})$$

With the graph convolution the threshold improves to

$$\frac{\|\mu - \nu\|}{\sigma} = \mathcal{O}\left(\frac{\sqrt{\log n}}{\sqrt{\mathbb{E}[|\mathcal{N}|]}}\right)$$



Graph Convolution



Original data

After graph convolution

The train and test data become linearly separable!

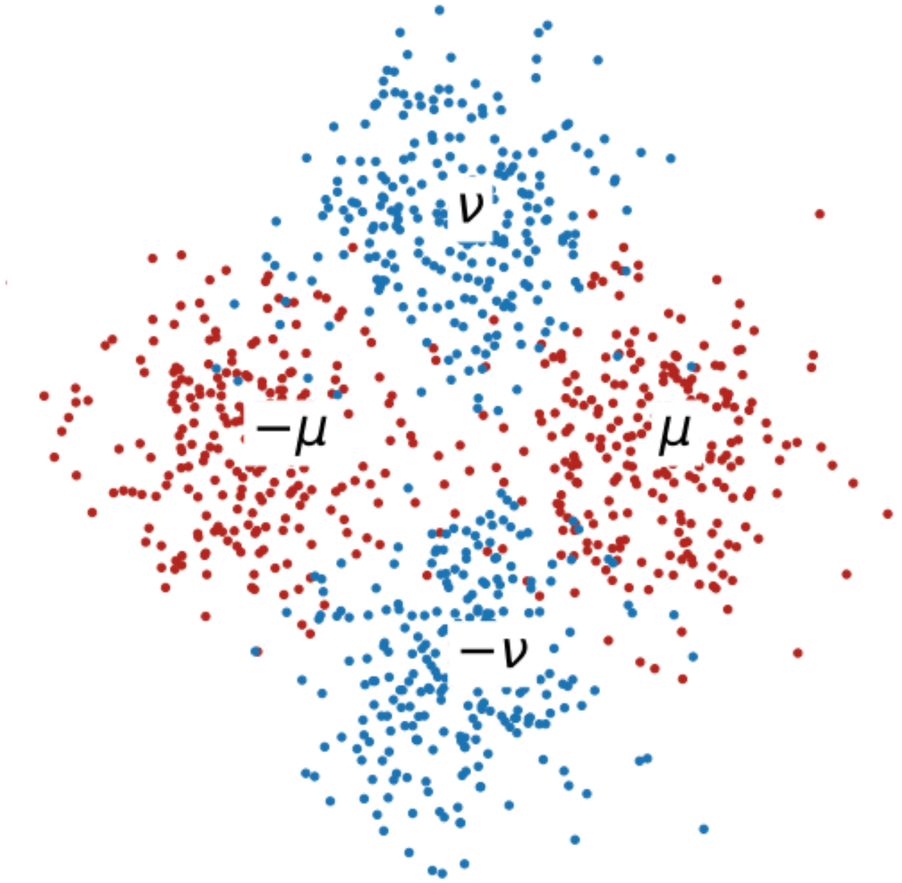
What about out-of-distribution generalization?

Graph convolution makes the test data separable with high probability even if the test distribution is an SBM with different parameters p, q.

We only need large enough $\frac{|p-q|}{p+q}$

Multi-layer Graph Convolution Networks

Data model for features



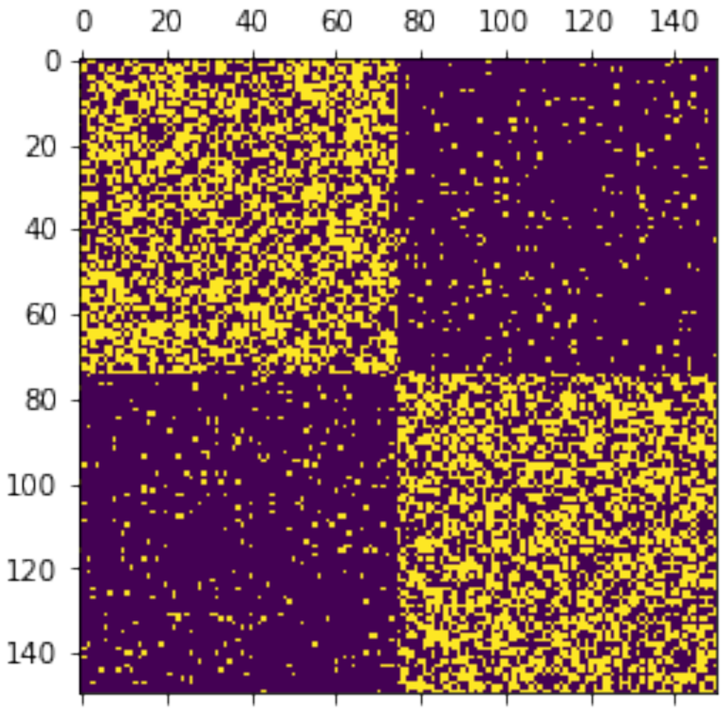
- 4 Gaussians with means $\mu, -\mu, \nu, -\nu$ and standard deviation σ

Data model for the graph

$$A \sim SBM(p, q)$$

$$\mathbb{P}(A_{ij} = 1) = \begin{cases} p & \text{if } i, j \text{ are in the same class} \\ q & \text{otherwise} \end{cases}$$

$A =$



Vanilla Graph Convolution Network (GCN): multiple layers

Example: 3-layer GCN

$$X' := \sigma_3(D^{-1}A\ \sigma_2(D^{-1}A\ \underbrace{\sigma_1(D^{-1}AXW_1)}_{\text{layer 1}}\ W_2)\ W_3)$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{layer 2}}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{layer 3}}$$

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): multiple layers

- If the intra-edge probability is $p = \Omega\left(\frac{\log^2 n}{n}\right)$
- and the inter-edge probability $q = \Omega\left(\frac{\log^2 n}{n}\right)$

- This mean that the expected degree $E[D] = \Omega(\log^2 n)$

Vanilla Graph Convolution Network (GCN): multiple layers

- If the intra-edge probability is $p = \Omega\left(\frac{\log^2 n}{n}\right)$

- and the inter-edge probability $q = \Omega\left(\frac{\log^2 n}{n}\right)$

-And we also have degree concentration, i.e., with very high probability all nodes have degree around its expectation $E[D]$

Vanilla Graph Convolution Network (GCN): multiple layers

- If the intra-edge probability is $p = \Omega\left(\frac{\log^2 n}{n}\right)$

- and the inter-edge probability $q = \Omega\left(\frac{\log^2 n}{n}\right)$

-and the distance between the means satisfies

$$\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{(E[D])^{1/4}}\right)$$

Vanilla Graph Convolution Network (GCN): multiple layers

-If the degree assumption is true and the distance between the means satisfies $\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{(E[D])^{1/4}}\right)$

-Then there exist 2 and 3 layer graph convolution networks that perfectly classify the nodes.

Vanilla Graph Convolution Network (GCN): multiple layers

-How do GCNs compare to the optimal Bayes classifier that does not use the graph data?

-The Bayes classifier achieves perfect classification when

$$\|\mu - \nu\| = \Omega(\sigma\sqrt{\log n})$$

-The GCN achieves perfect classification when

$$\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{(E[D])^{1/4}}\right)$$

Vanilla Graph Convolution Network (GCN): multiple layers

- GCN improves the perfect separability threshold by $(E[D])^{1/4}$, which can be quite large!

- That's because $E[D] = \Omega(\log^2 n)$ and $E[D] \leq \Theta(n)$

Performance of GCN for denser graphs

- If we assume a denser graph, $p, q = \Omega\left(\frac{\log n}{\sqrt{n}}\right)$

- which implies that $E[D] = \Omega\left(\sqrt{n}\log n\right)$

-This is denser than before since previously we assumed that $E[D] = \Omega(\log^2 n)$

Performance of GCN for denser graphs

-So if we assume a denser graph
-and if the distance between the means is

$$\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{n^{1/4}}\right)$$

-Then there are 2 and 3 layer GCN that can perfectly classify the data with high probability.

Performance of GCN for denser graphs

-The base classifier achieves perfect classification when

$$\|\mu - \nu\| = \Omega(\sigma\sqrt{\log n})$$

-The GCN on sparse graphs achieves perfect classification when

$$\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{(E[D])^{1/4}}\right)$$

-The GCN on dense graphs achieves perfect classification when

$$\|\mu - \nu\| = \Omega\left(\frac{\sigma\sqrt{\log n}}{n^{1/4}}\right)$$

How to place the graph convolutions?

It does not matter how you place the graph convolutions.

3 Layers - **1**

1 Conv. at 3rd layer

How to place the graph convolutions?

It does not matter how you place the graph convolutions.

3 Layers - **1**,**1**

1 Conv. at 3rd layer

1 Conv. at 2nd layer

How to place the graph convolutions?

It does not matter how you place the graph convolutions.

3 Layers - **1**,**1**,**0**

1 Conv. at 3rd layer

1 Conv. at 2nd layer

0 Conv. at 1st layer

Placement of convolutions does not affect performance

2-layers, 1 convolution at the last layer

$$X' := \sigma_2(D^{-1}A\sigma_1(XW_1)W_2)$$

2-layers, 2 convolutions at the last layer

$$X' := \sigma_2(D^{-1}A)^2\sigma_1(XW_1)W_2)$$

3-layers, 2 convolutions at the last layer

$$X' := \sigma_3(D^{-1}A)^2\sigma_2(\sigma_1(XW_1)W_2)W_3)$$

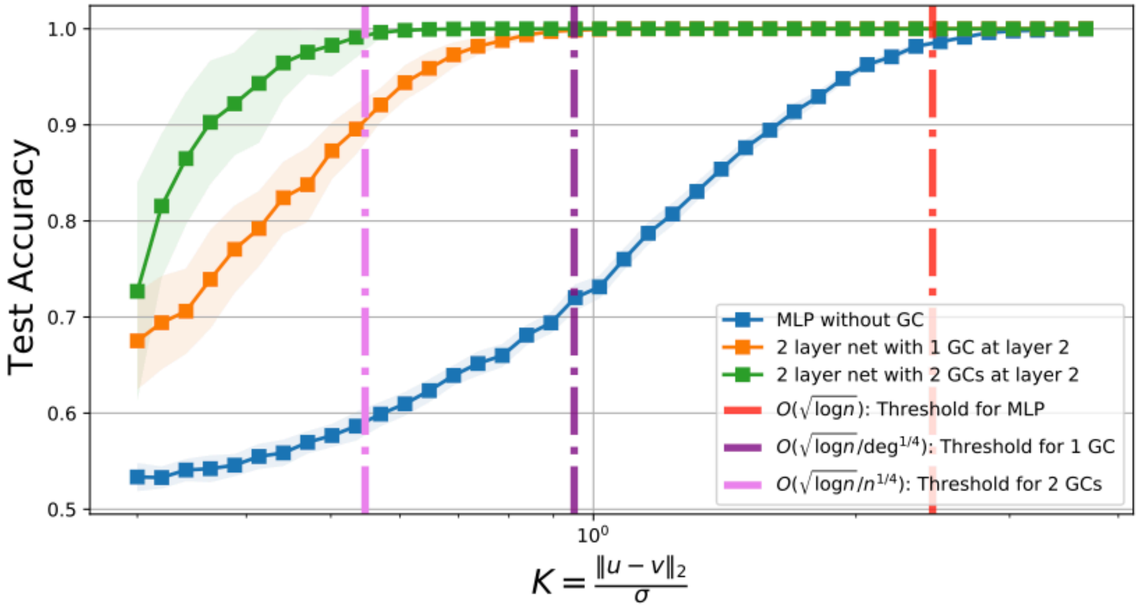
3-layers, 2 convolutions at the 2nd and 3rd layers

$$X' := \sigma_3(D^{-1}A\sigma_2(D^{-1}A\sigma_1(XW_1)W_2)W_3)$$

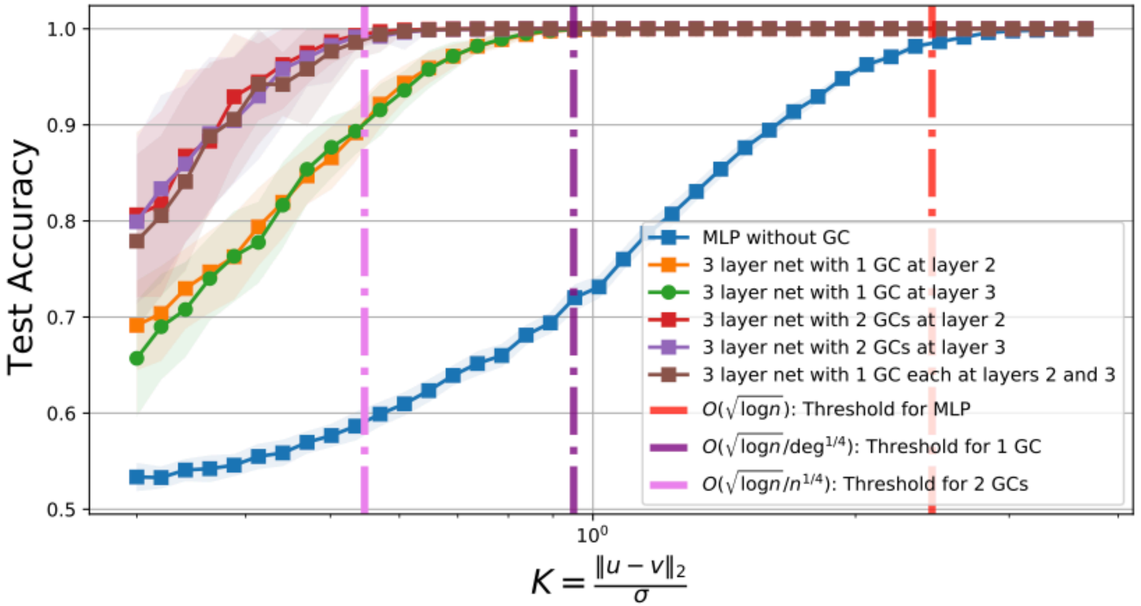
3-layers, 2 convolutions at the 2nd layer

$$X' := \sigma_3(\sigma_2((D^{-1}A)^2\sigma_1(XW_1)W_2)W_3)$$

Placement of convolutions does not affect performance



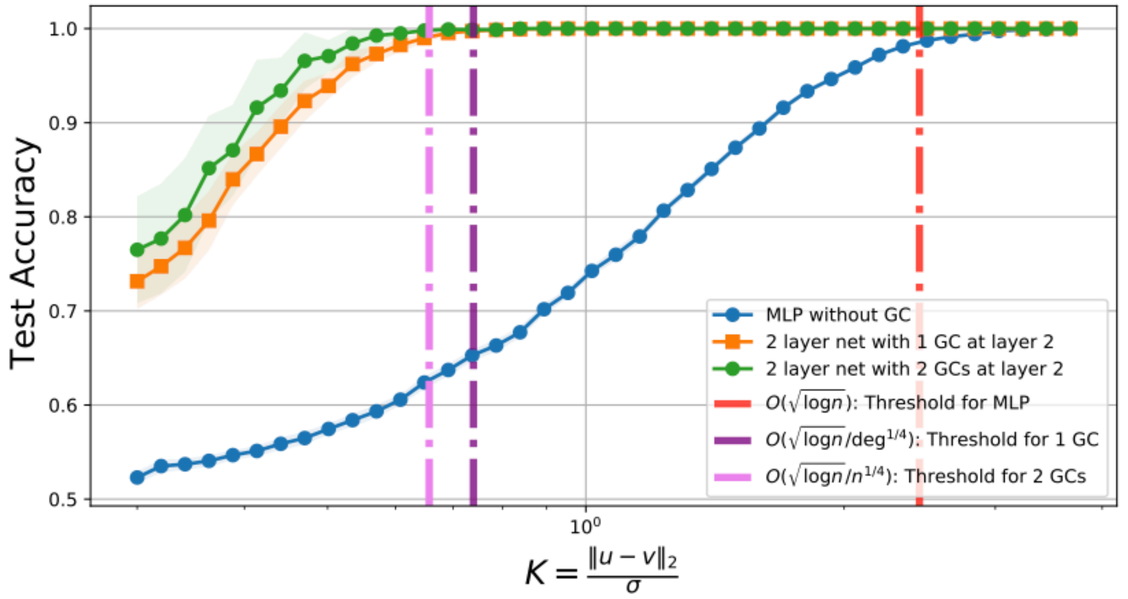
(a) Two-layer networks with $(p, q) = (0.2, 0.02)$.



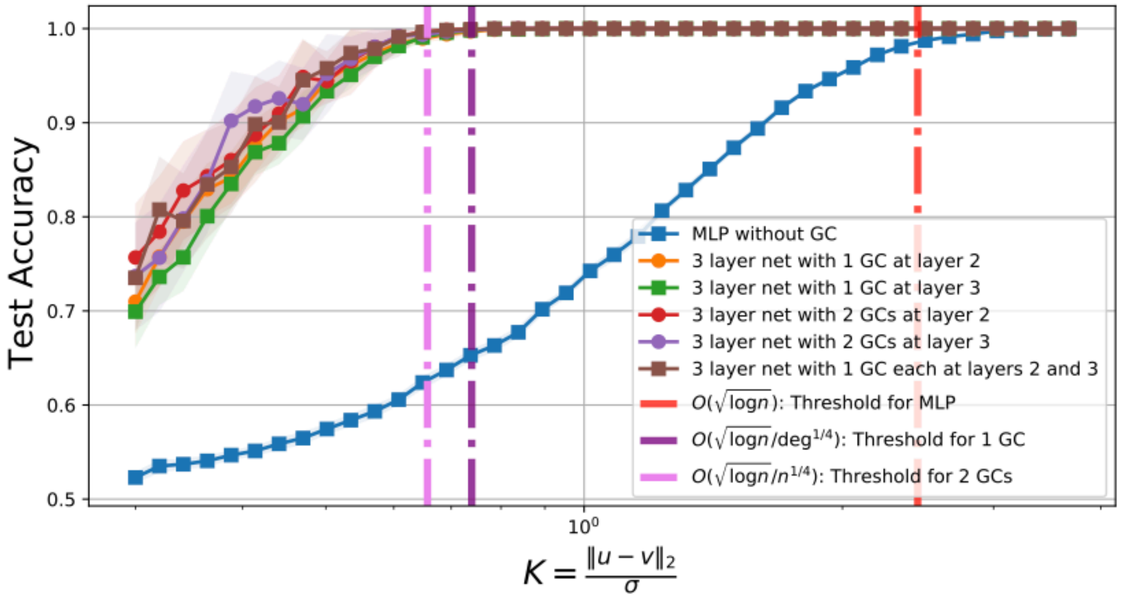
(b) Three-layer networks with $(p, q) = (0.2, 0.02)$.

A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance



(c) Two-layer networks with $(p, q) = (0.5, 0.1)$.



(d) Three-layer networks with $(p, q) = (0.5, 0.1)$.

A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance

OGB products dataset

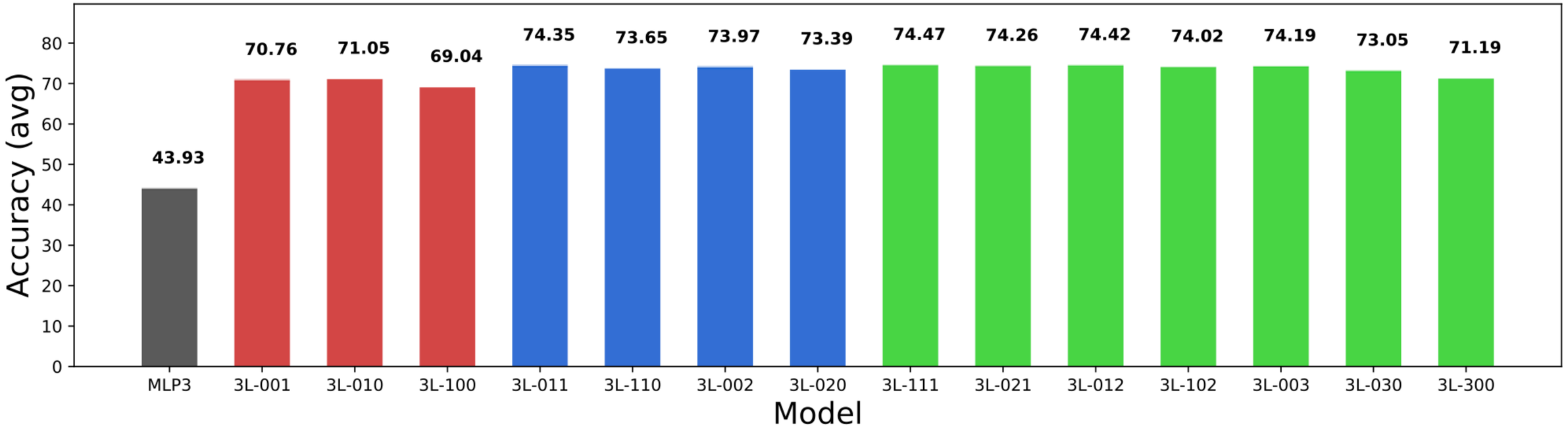
Graph: The `ogbn-products` dataset is an undirected and unweighted graph, representing an Amazon product co-purchasing network [1]. Nodes represent products sold in Amazon, and edges between two products indicate that the products are purchased together. We follow [2] to process node features and target categories. Specifically, node features are generated by extracting bag-of-words features from the product descriptions followed by a Principal Component Analysis to reduce the dimension to 100.

Prediction task: The task is to predict the category of a product in a multi-class classification setup, where the 47 top-level categories are used for target labels.

OPEN GRAPH BENCHMARK

Placement of convolutions does not affect performance

OGB products



- Multi-class dataset with 2,449,029 nodes, 61,859,140 edges

A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance

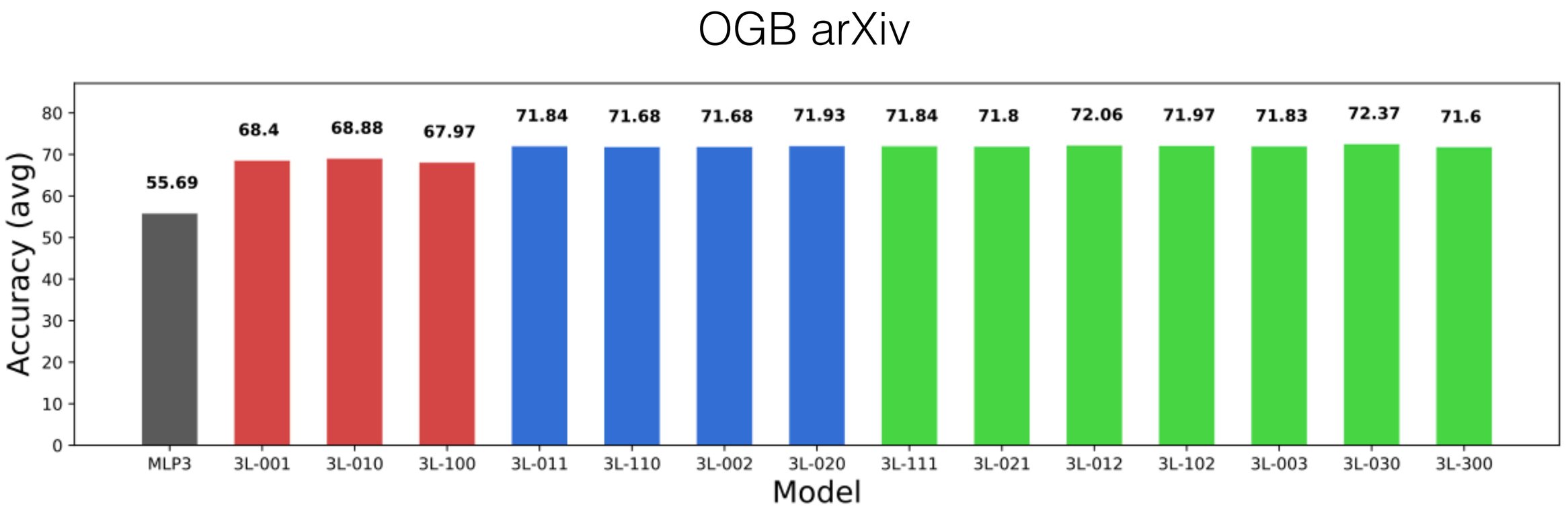
OGB arXiv dataset

Graph: The `ogbn-arxiv` dataset is a directed graph, representing the citation network between all Computer Science (CS) arXiv papers indexed by MAG [1]. Each node is an arXiv paper and each directed edge indicates that one paper cites another one. Each paper comes with a 128-dimensional feature vector obtained by averaging the embeddings of words in its title and abstract. The embeddings of individual words are computed by running the skip-gram model [2] over the MAG corpus. We also provide the mapping from MAG paper IDs into the raw texts of titles and abstracts here. In addition, all papers are also associated with the year that the corresponding paper was published.

Prediction task: The task is to predict the 40 subject areas of arXiv CS papers, e.g., cs.AI, cs.LG, and cs.OS, which are manually determined (i.e., labeled) by the paper's authors and arXiv moderators. With the volume of scientific publications doubling every 12 years over the past century, it is practically important to automatically classify each publication's areas and topics. Formally, the task is to predict the primary categories of the arXiv papers, which is formulated as a 40-class classification problem.

OPEN GRAPH BENCHMARK

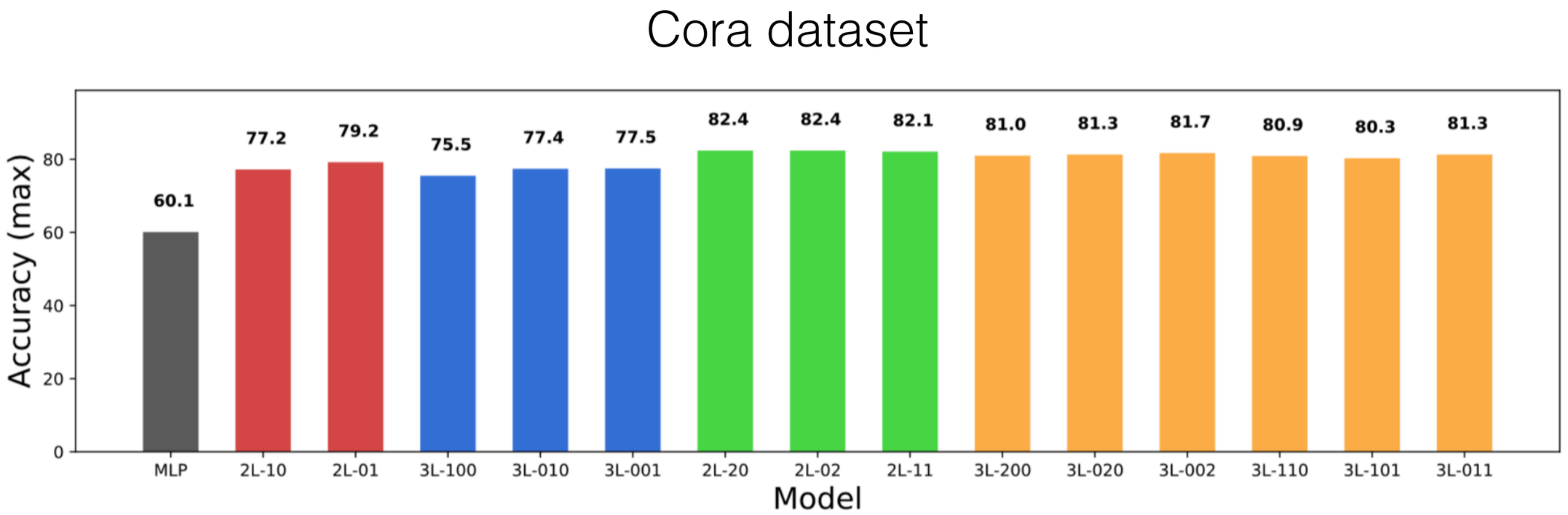
Placement of convolutions does not affect performance



- Multi-class dataset with169,343 nodes, 1,166,243 edges

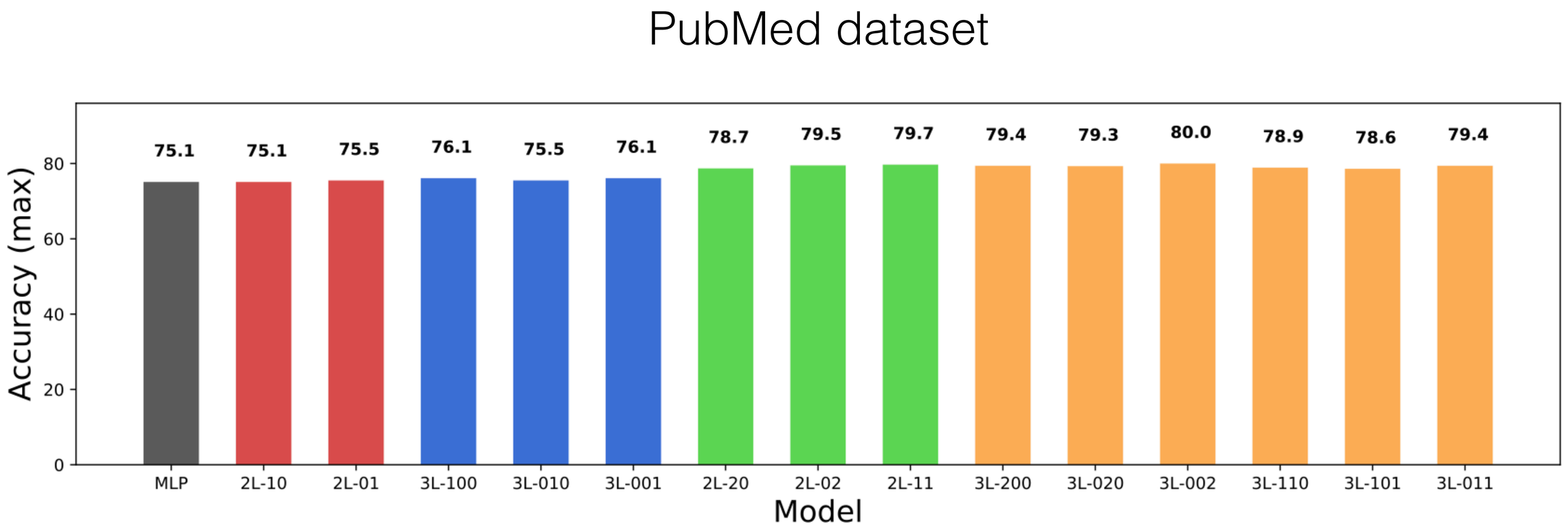
A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance



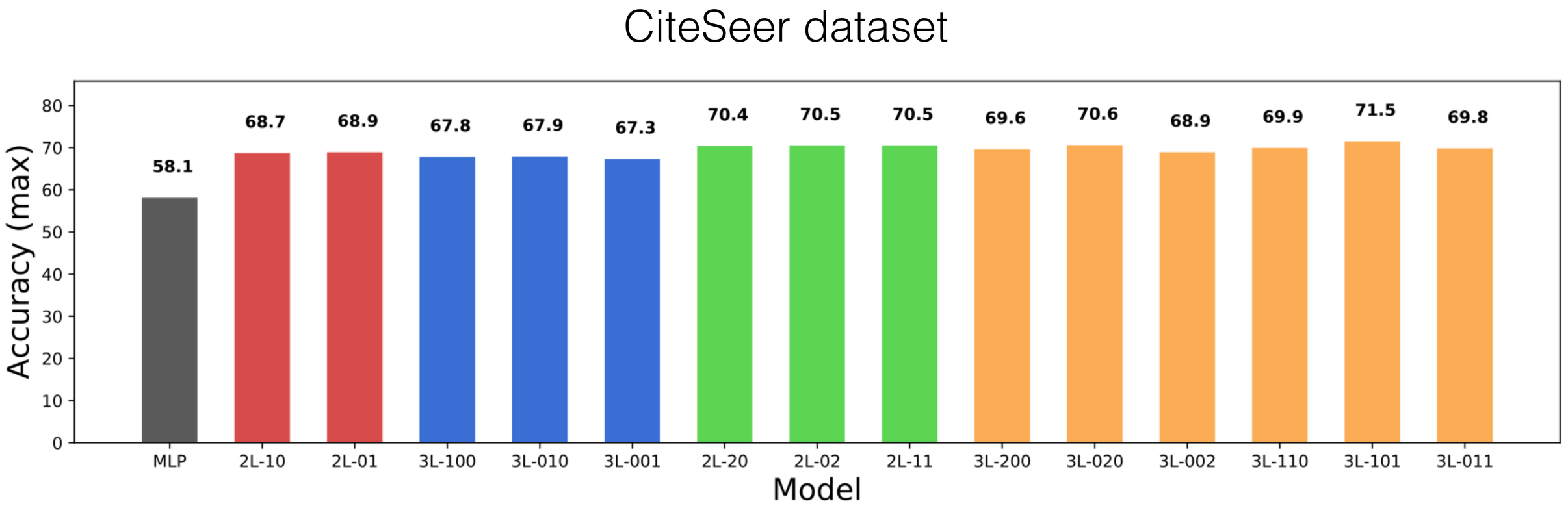
A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance



A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)

Placement of convolutions does not affect performance



A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)