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

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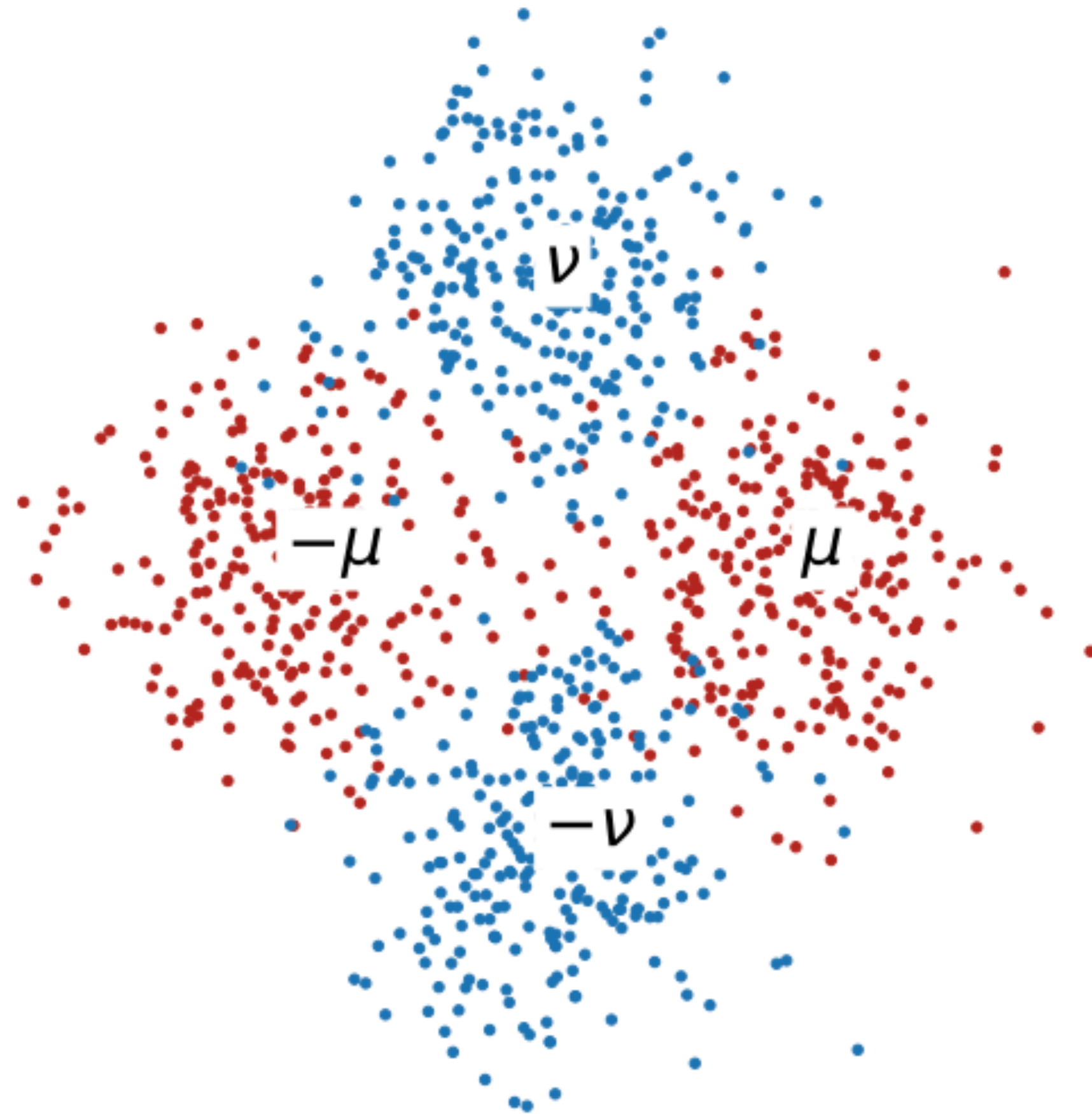
UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

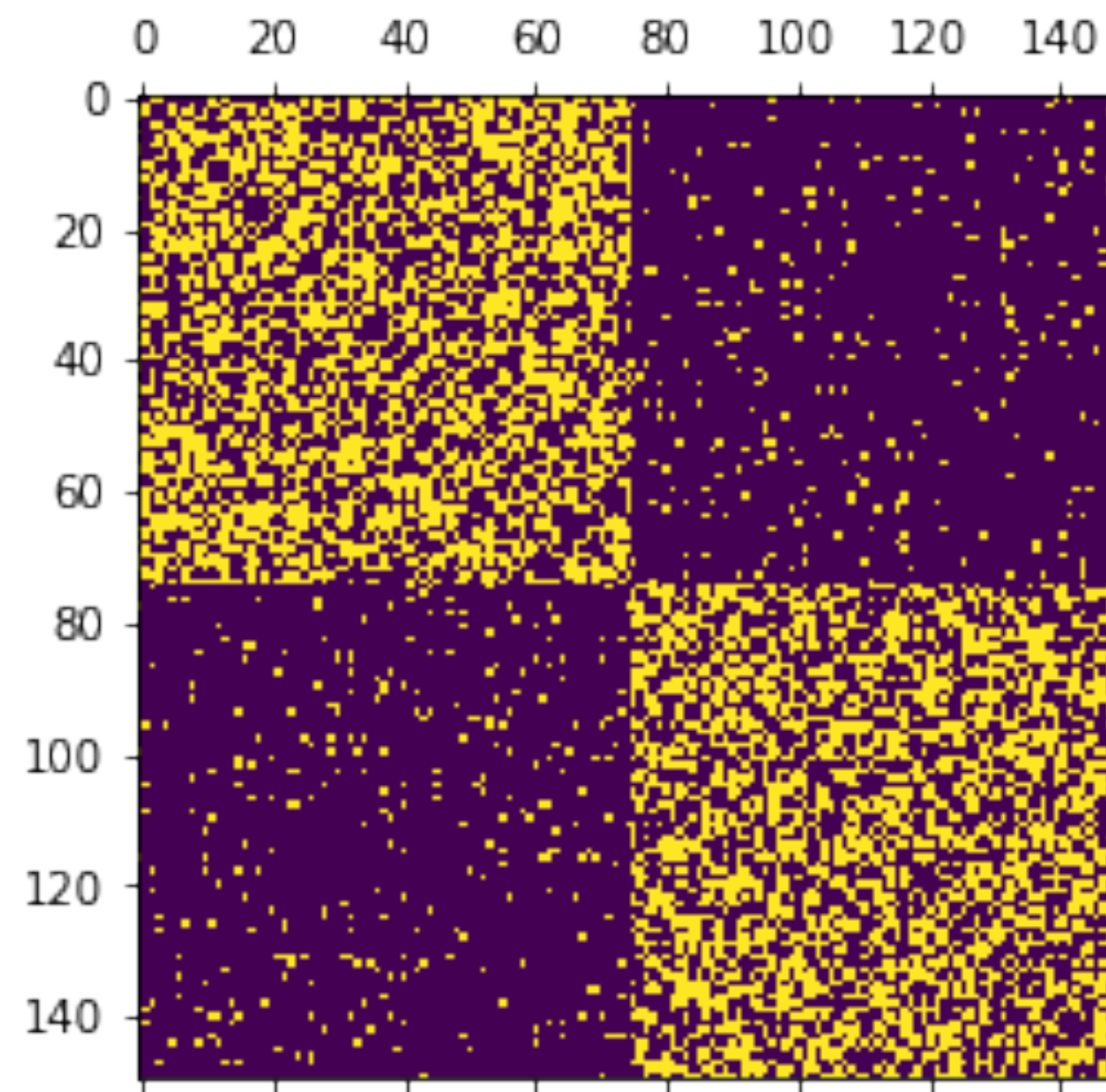# Multi-layer Graph Convolution Networks

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

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

layer 1

layer 2

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 $p = \Omega\left(\dfrac{\log^2 n}{n}\right)$

-

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

-

-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 $p = \Omega\left(\dfrac{\log^2 n}{n}\right)$

-

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

-

-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 $p = \Omega\left(\dfrac{\log^2 n}{n}\right)$

-

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

-

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

-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, $p, q = \Omega \left(\dfrac{\log n}{\sqrt{n}} \right)$

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

Placement of convolutions does not affect performance

2-layer, 1 convolution per layer

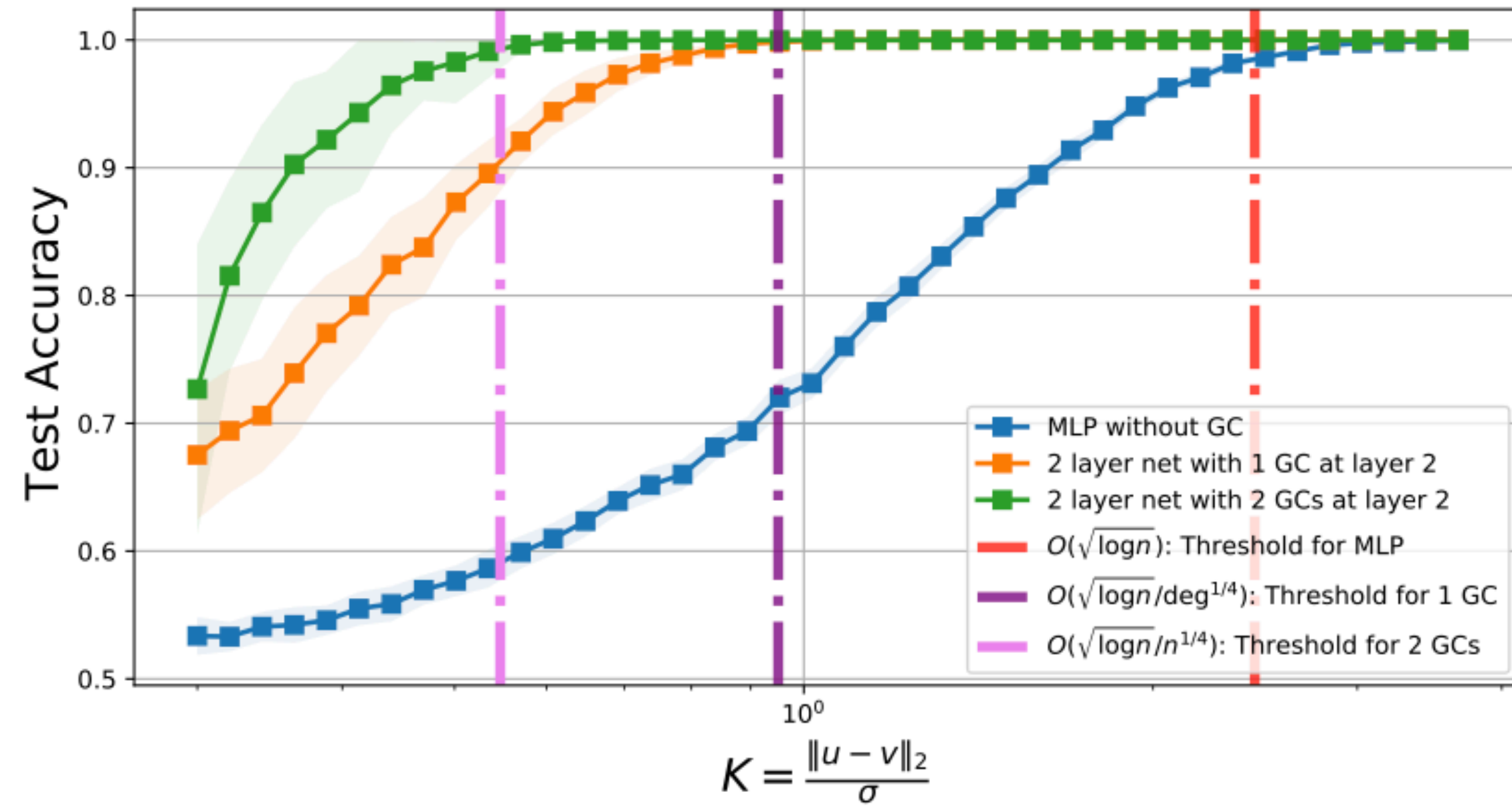$$X' := \sigma_2(D^{-1}A\sigma_1(D^{-1}AXW_1)W_2)$$

2-layer, 2 convolutions at the first layer

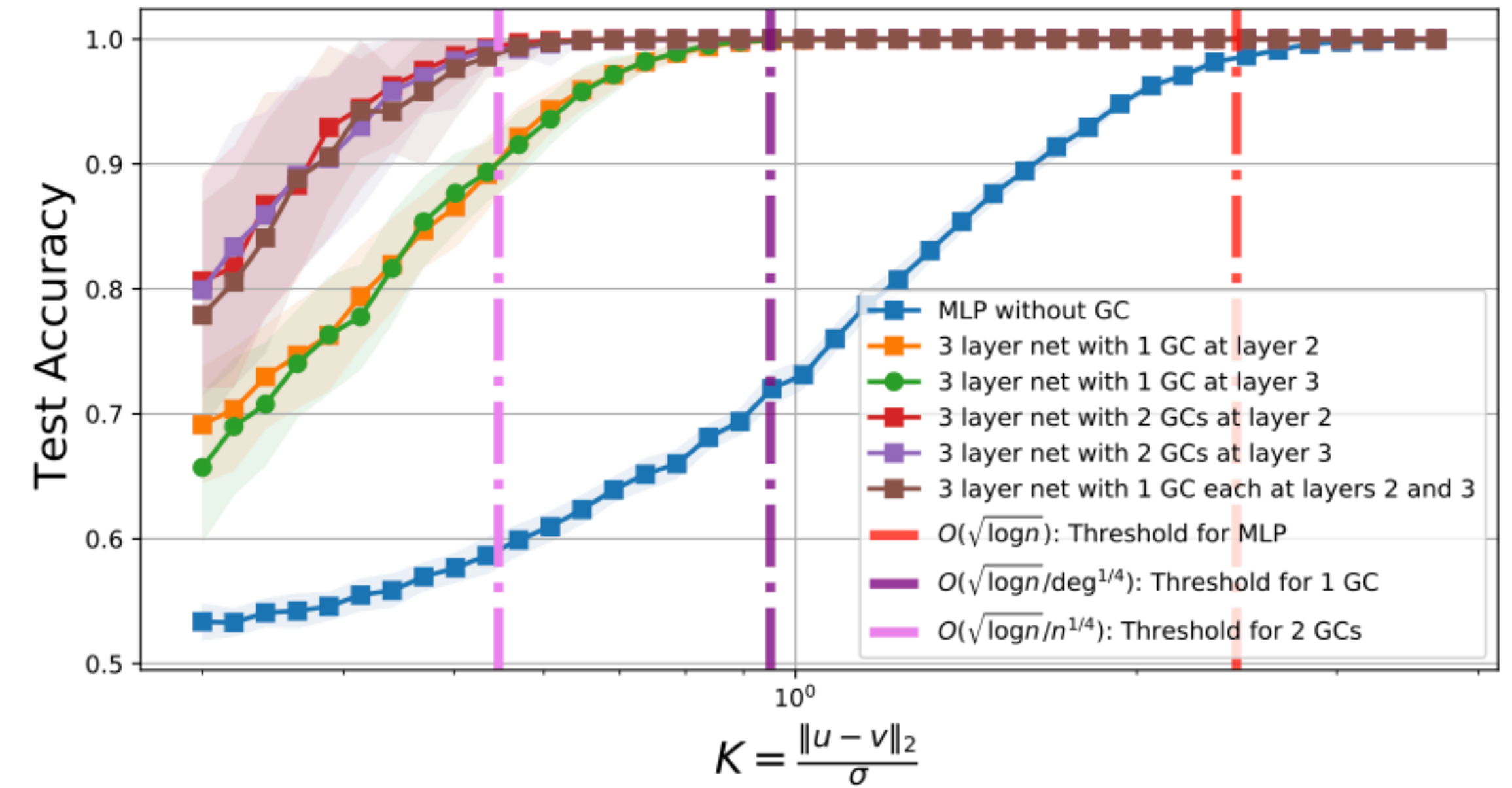$$X' := \sigma_2(\sigma_1((D^{-1}A)^2 XW_1)W_2)$$

2-layer, 2 convolutions at the last layer

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

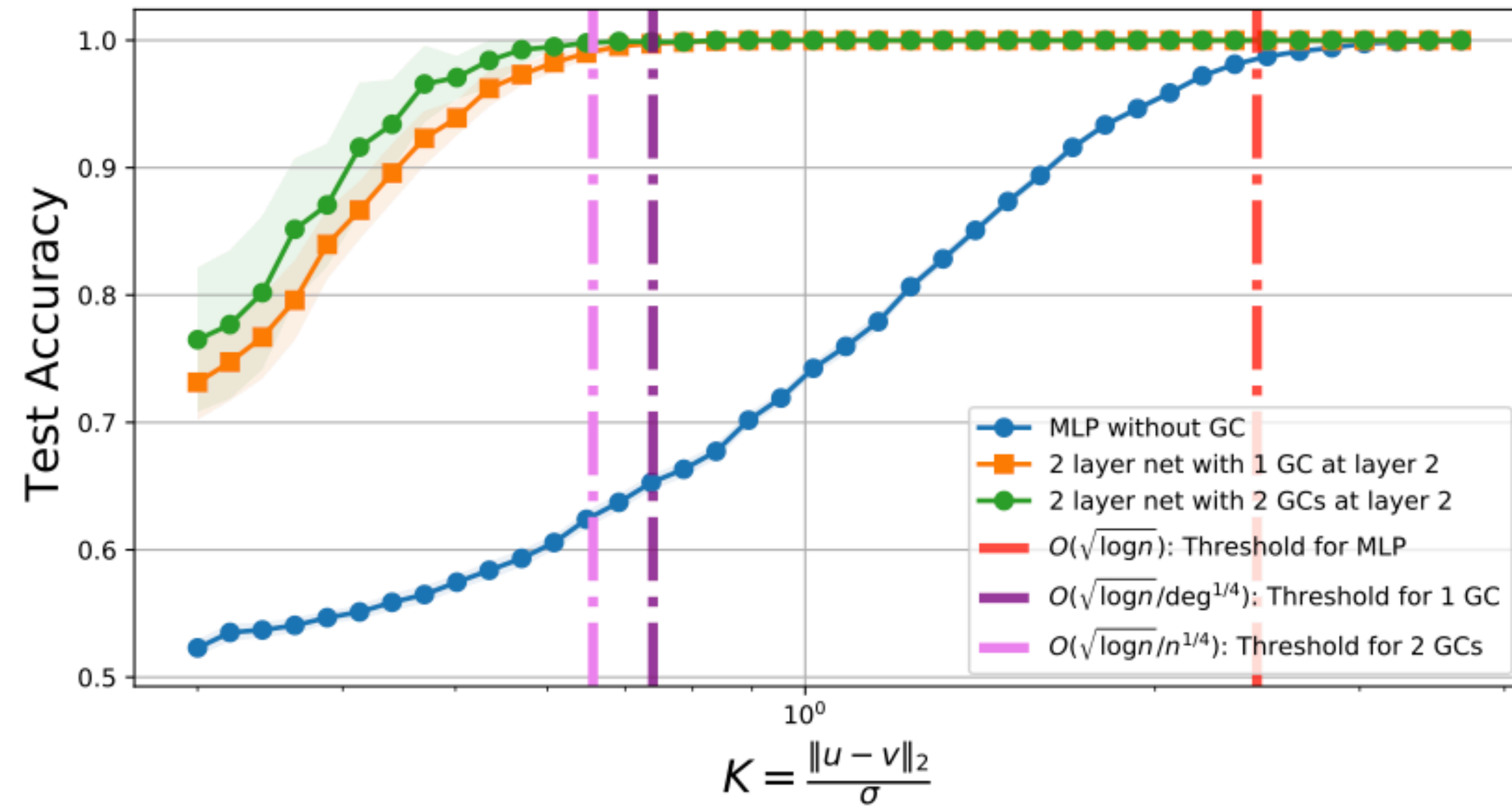# Placement of convolutions does not affect performance

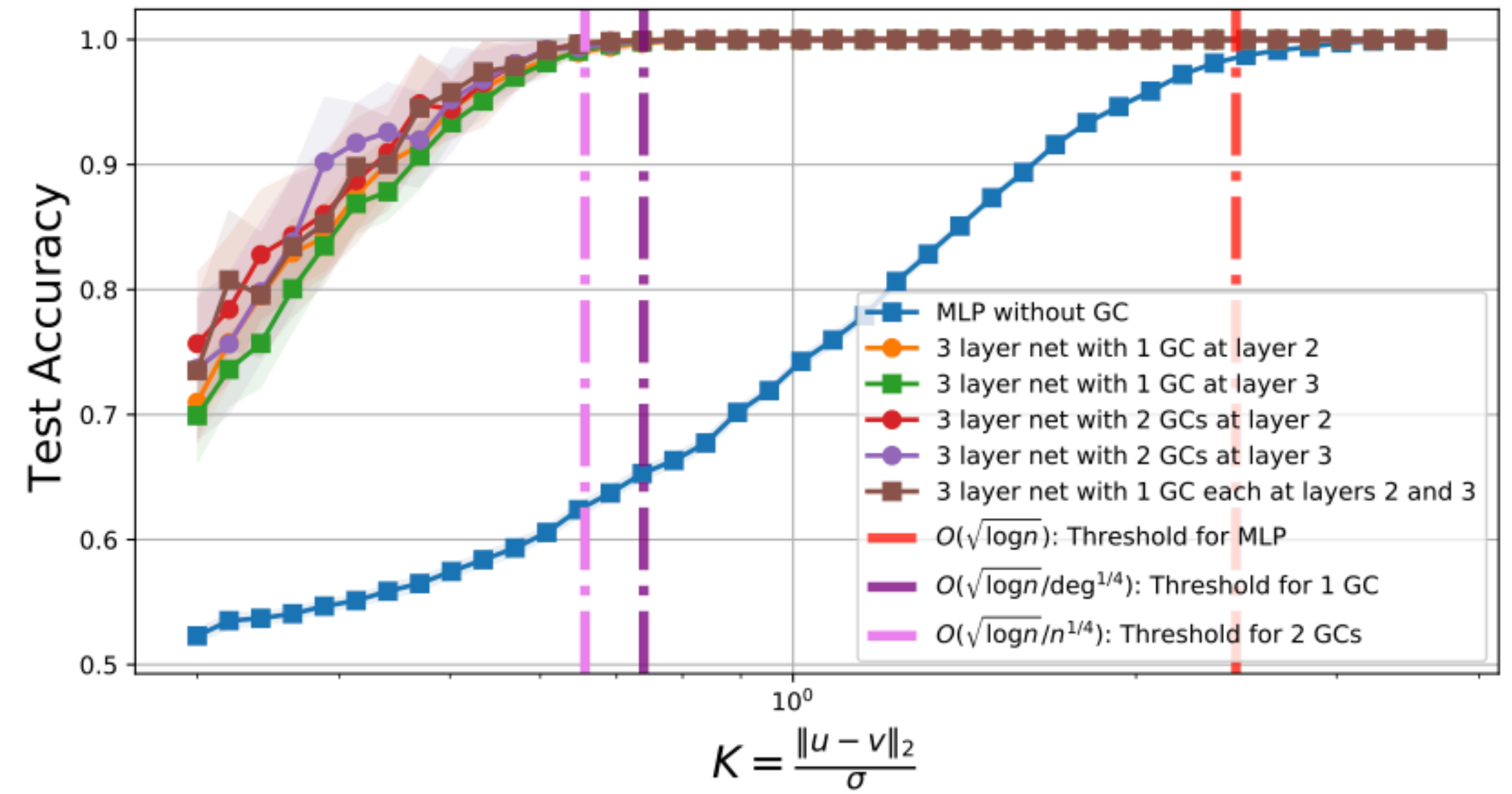(a) Two-layer networks with $(p, q) = (0.2, 0.02)$.

(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)

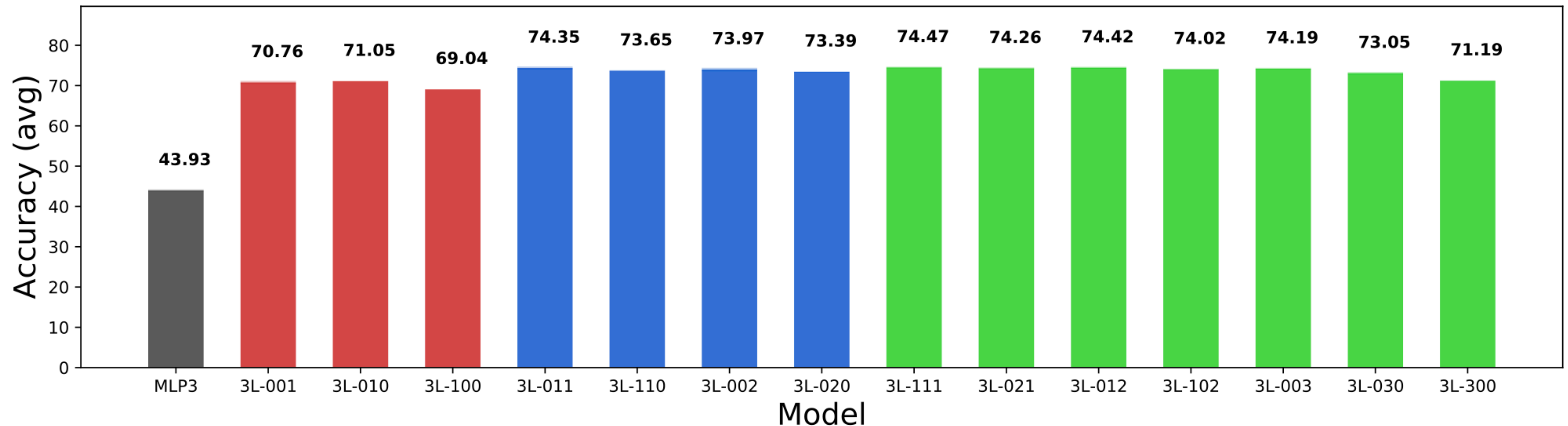# Placement of convolutions does not affect performance

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.

OGB

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)

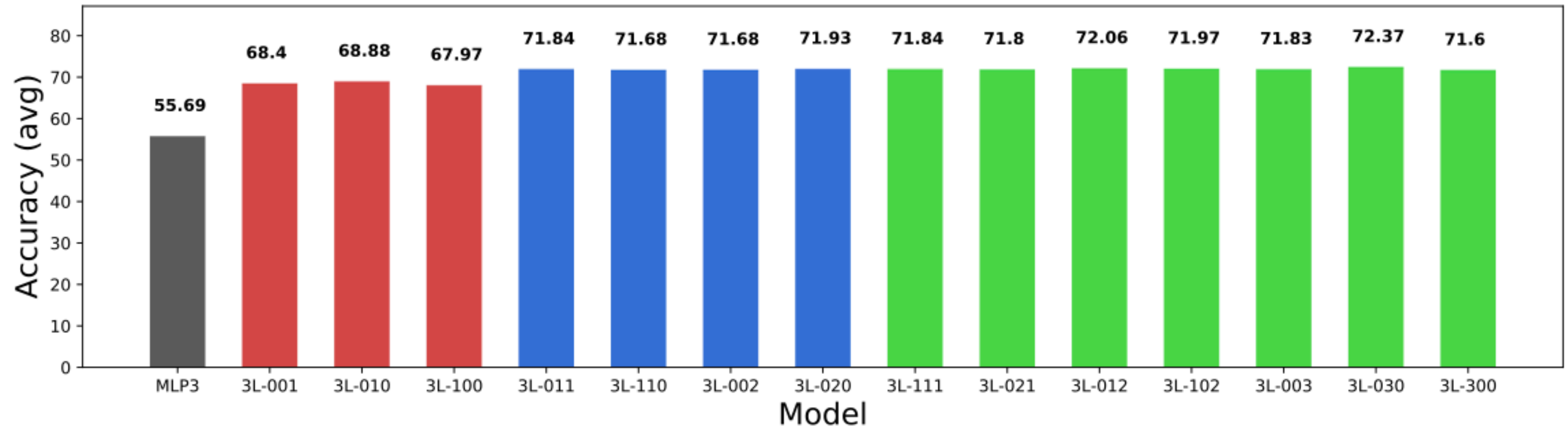# Placement of convolutions does not affect performance

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.

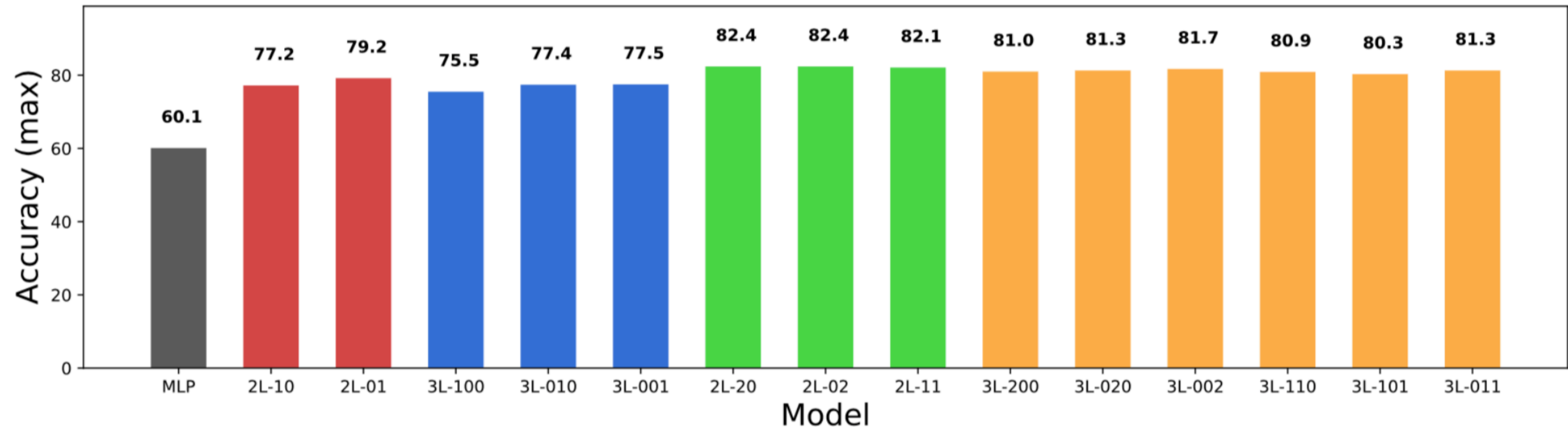# Placement of convolutions does not affect performance

OGB arXiv

- Multi-class dataset with 169,343 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

Cora dataset

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

PubMed dataset

A. Baranwal, K. Fountoulakis, A. Jagannath, Effects of Graph Convolutions in Multi-Layer Networks, arXiv:2204.09297 (2022)
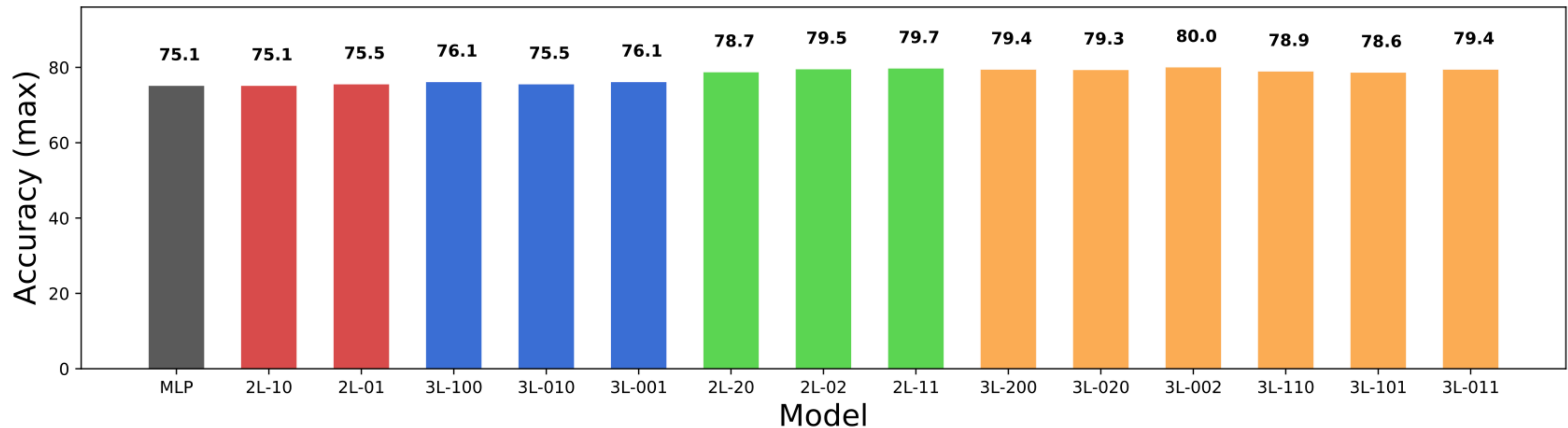
Placement of convolutions does not affect performance

CiteSeer dataset

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