Can Graph Neural Networks Go "Online"? An Analysis of Pretraining and Inference

Lukas Galke, Iacopo Vagliano, and Ansgar Scherp

Kiel University, ZBW – Leibniz Information Centre for Economics, University of Essex

Objectives

- Create an experimental setup to evaluate graph neural networks on newly inserted nodes after training.
- Evaluate whether re-training from scratch or fine-tuning a pre-trained model is preferable.
- Is there a difference when the training graph is large or when it is small?

Motivation

How do graph neural networks deal with previously unseen nodes as present in:

- Classification
- New user/item in recommendation [1]
- Online learning [2]

Retrain from scratch or fine-tune a pre-trained model? Retraining is expensive.
→ Inductive learning [3] on partial graphs

Experimental Setup

Procedure

- Pretrain model on subgraph induced by training nodes
- Insert new nodes and edges
- Optional: update normalizing factor
- Continue training: inference epochs
- Evaluate accuracy on new nodes

Datasets: Cora, Citeseer, Pubmed
→ **Setting A:** Small training graph
→ **Setting B:** Large training graph

Datasets

Dataset	**Cora**	**Citeseer**	**Pubmed**
Classes	7	6	3
Features	1,433	3,703	500
Nodes	2,708	3,327	19,717
Edges	5,278	4,552	44,324
Avg. Degree	3.90	2.77	4.50

Setting	**A**	**B**	**A**	**B**	**A**	**B**
Train Samples	440	2,268	620	2,707	560	19,157
Train Edges	342	3,582	139	2,939	34	41,858
Unseen Nodes	2,268	440	2,707	620	19,157	560
Unseen Edges	4,936	1,696	4,413	1,613	44,290	2,466
Test Samples	1,000	440	1,000	620	1,000	560
Label Rate	**16.2%**	**83.8%**	**18.6%**	**81.4%**	**2.8%**	**97.2%**

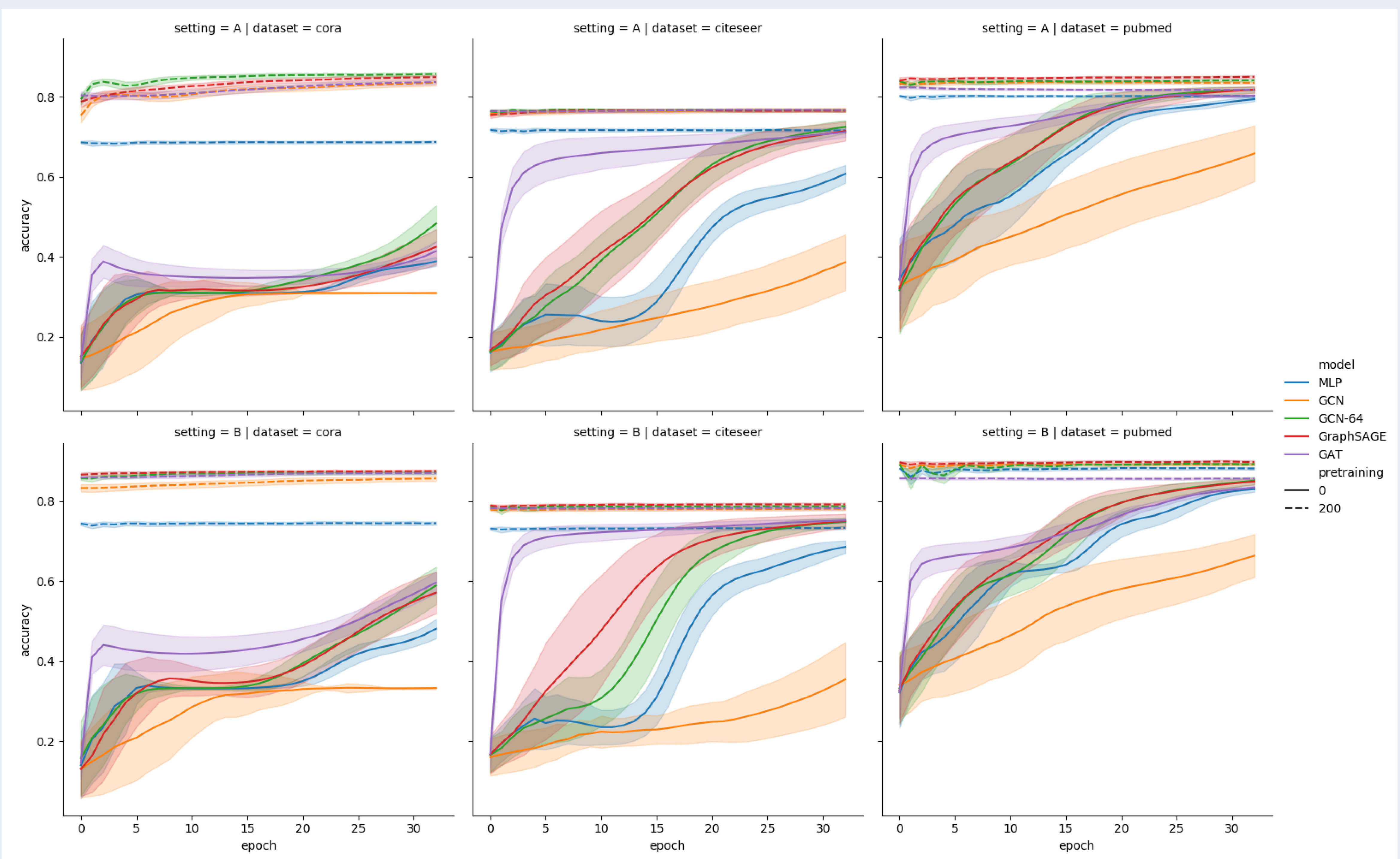
Models

Propagation rule for node i in layer $l+1$:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)}\right)$$

- GCNs [4] use $c_{ij} = \sqrt{|\mathcal{N}(i)|} \cdot \sqrt{|\mathcal{N}(j)|}$
- GraphSAGE-mean [5] uses $c_{ij} = |\mathcal{N}(i)|$
- GATs [6] use attention instead of $\frac{1}{c_{ij}}$
- All models use two convolution layers
- Hyperparameters as in original works

Main Result: Pretraining works.



- Mean test accuracy and standard deviation after 0–32 inference epoch across 100 runs
- Pretrained for 200 epochs (*Dashed lines*) — retraining from scratch (*Solid lines*)
- Setting A: Small training graph *(Top)* — Setting B: Large training graph *(Bottom)*

Conclusion

- Pretrained graph neural networks yield high accuracy with low variance even though new nodes and edges are inserted into the graph.
- This property is mandatory for applying graph neural networks to large-scale, dynamic graphs as often found in real-world scenarios.
- Code: lgalke/gnn-pretraining-evaluation

References

[1] Lukas Galke, Florian Mai, Iacopo Vagliano, and Ansgar Scherp. Multi-modal adversarial autoencoders for recommendations of citations and subject labels. In *UMAP*, pages 197–205. ACM, 2018.

[2] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Comput. Surv.*, 47(1):10:1–10:36, May 2014.

[3] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 40–48. JMLR.org, 2016.

[4] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

[5] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, 2017.

[6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018.

Contact Information

- Name: Lukas Galke
- Web: http://www.lpag.de
- Email: lga@informatik.uni-kiel.de
- Twitter: @_lpag

C A U Christian-Albrechts-Universität zu Kiel

ZBW Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics