When Does Self-Supervision Help Graph Convolutional Networks?

Yuning You [*1] **Tianlong Chen** [*1] **Zhangyang Wang** [1] **Yang Shen** [1]

Appendix

A. Implementation Details

Datasets and network hyper-parameters. Three standard citation networks Cora, Citeseer and PubMed (Sen et al., 2008) are used and their statistics are summarized in Table 4 (Section 4), with 500 validation and 1,000 test nodes in each dataset. Hyper-parameters for all three state-of-the-arts (SOTAs) are kept default except that the learning rate and the weight decay of graph convolutional network (GCN) on Cora were slightly tuned to be 0.008 and 8e-5 based on validation, respectively.

Self-supervised tasks. We select K-means as the algorithm for node clustering and METIS (Karypis & Kumar, 1998) for graph partitioning. For hyper-parameters in self-supervised tasks, we perform grid search to tune their values based on validation performance. Let $\{n_1 : n_2 : n_3\}$ denote the set $\{n_1, n_1+n_2, n_1+2n_2, ..., n_3-n_2, n_3\}$ where $n_3 > n_1$ and $(n_3 - n_1)$ mod $n_2 = 0$. In node clustering (with the alignment procedure (Sun et al., 2019)), the number of clusters is set at 200 as suggested by Sun et al. (2019), and we set the loss weights $\alpha_1 + \alpha_2 = 1$ tuning α_1 in $\{0.2 : 0.1 : 0.9\}$. In graph partitioning, the partitioning number is tuned in $\{8 : 1 : 16\}$, and we set the loss weights $\alpha_1 + \alpha_2 = 1$ tuning α_1 in $\{0.2 : 0.1 : 0.9\}$. In graph completion, we reduce the dimension of the self-supervised label matrix with singular value decomposition, tuning the reduced dimension in $\{24 : 4 : 48\}$. The reason is that, as the output dimension of each SOTA's feature extractor is set as 16, it would be difficult to regress to the vector with dimension higher than thousands. In addition, we set $\alpha_1 = 1$ and tune α_2 in $\{0.3 : 0.05 : 0.7\}$ for GCN and $\{0.1 : 0.05 : 0.9\}$ for graph markov neural network (GMNN) and GraphMix.

Adversarial attacks and defense. We choose Nettack (Zügner et al., 2018), a well-known and effective algorithm as the attacker. It allows for attacking links, features, and both (attacking links first and then features). Since graph structures (and often node features) are discrete and Nettack constructs adversarial attacks in a discrete domain, we conduct experiments of graph adversarial attacks and defenses just for the Cora and Citeseer datasets. Let n_{perturb} denote the *perturbation intensity*. We set attacks on links with intensity n_{perturb} as perturbing (add or delete) n_{perturb} links, on features as perturbing (flip 1 to 0 or reversely) $10n_{\text{perturb}}$ features, and on links & features as perturbing n_{perturb} links and $10n_{\text{perturb}}$ features. In adversarial training, we use links & features attacks with $n_{\text{perturb}} = 2$ to generate adversarial examples and set $\alpha_3 = 1, |\mathcal{V}_{\text{clean}}| = 100, |\mathcal{V}_{\text{attack}}| = 200$.

B. Inductive Fully Supervised Node Classification with Multi-Task Self-Supervision

We perform experiments on the same three standard citation networks Cora, Citeseer and PubMed in Table 4 (Section 4) with 500 validation and 1,000 test nodes in each dataset, while all the rest nodes are used for training following the inductive fully supervised setting in (Chen et al., 2018). Hyper-parameters (tuning) are consistent to that in the semi-supervised setting (see Section A). Similar to that in the convolutional neural networks (CNNs), Table 9 shows self-supervision in GCNs is more helpful to semi-supervised/few-shot cases than fully supervised ones, although in the latter case it also shows moderate improvements.

Table 9: Inductive fully supervised node classification experiments on SOTAs (GCN, GAT and GIN) with multi-task self-supervision.

Datasets	Cora	Citeseer	PubMed
GCN	85.95+-0.41	77.94+-0.45	87.04+-0.17
GCN-Clu	85.89+-0.51	77.58+-0.34	86.54+-1.37
GCN-Par	85.89+-0.41	77.88+-0.42	86.57+-0.22
GCN-Comp	86.32+-0.36	77.75+-0.49	87.60+-0.18
GAT	83.77+-0.67	78.31+-0.23	86.53+-0.24
GAT-Clu	84.14+-0.60	76.87+-0.24	86.34+-0.18
GAT-Par	83.33+-0.79	76.91+-0.17	85.91+-0.20
GAT-Comp	84.46+-0.64	76.58+-0.95	86.94+-0.28
GIN	83.64+-0.39	78.92+-0.31	90.03+-0.18
GIN-Clu	84.06+-0.36	78.94+-0.37	90.24+-0.19
GIN-Par	83.87+-0.45	79.01+-0.38	89.84+-0.26
GIN-Comp	82.39+-0.74	77.33+-0.62	89.88+-0.27

[*]Equal contribution [1]Texas A&M University. Correspondence to: Yang Shen <yshen@tamu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

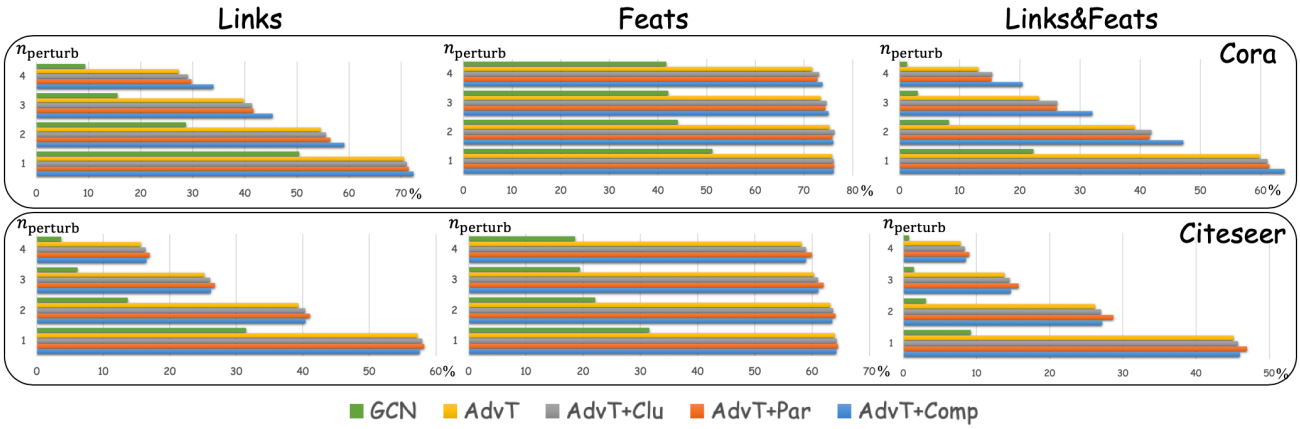

Figure 3: Adversarial attacks on Cora and Citeseer using adversarial training (abbr. AdvT) without or with graph self-supervision under different attack intensities.

C. Adversarial Defense against Attacks with Different Perturbation Intensities

Figure 3 shows the results that we generate attacks with varying perturbation intensities ($n_{\text{perturb}} \in \{1, 2, 3, 4\}$) on graph convolutional network (GCN) to check the generalizabilty of our conclusions in Section 4.2. The corresponding values are shown in Tables 10-17.

Table 10: Adversarial attacks on Cora using adversarial training without or with graph self-supervision with $n_{\text{per}} = 1$.

Attacks	Links	Feats	Links & Feats
GCN	50.42 ± 1.08	51.20 ± 1.01	22.22 ± 1.17
AdvT	70.50 ± 2.37	75.76 ± 1.28	59.82 ± 3.25
AdvT+Clu	71.04 ± 2.42	76.08 ± 1.36	61.10 ± 3.51
AdvT+Par	71.44 ± 2.33	76.19 ± 0.85	61.46 ± 3.08
AdvT+Comp	72.38 ± 1.83	76.12 ± 1.01	63.96 ± 2.95

Table 11: Adversarial attacks on Cora using adversarial training without or with graph self-supervision with $n_{\text{per}} = 2$.

Attacks	Links	Feats	Links & Feats
GCN	28.72 ± 0.63	44.06 ± 1.23	8.18 ± 0.27
AdvT	54.58 ± 2.57	75.25 ± 1.26	39.08 ± 3.05
AdvT+Clu	55.54 ± 3.19	76.24 ± 0.99	41.84 ± 3.48
AdvT+Par	56.36 ± 2.57	75.88 ± 0.72	41.57 ± 3.47
AdvT+Comp	59.05 ± 3.29	76.04 ± 0.68	47.14 ± 3.01

Table 12: Adversarial attacks on Cora using adversarial training without or with graph self-supervision with $n_{\text{per}} = 3$.

Attacks	Links	Feats	Links & Feats
GCN	15.56 ± 0.37	42.08 ± 1.20	3.06 ± 0.42
AdvT	39.76 ± 1.49	73.44 ± 1.48	23.19 ± 2.31
AdvT+Clu	41.31 ± 2.30	74.62 ± 1.25	26.19 ± 2.00
AdvT+Par	41.66 ± 2.17	74.36 ± 0.99	26.16 ± 2.64
AdvT+Comp	45.35 ± 2.76	75.06 ± 0.86	32.03 ± 3.20

Table 13: Adversarial attacks on Cora using adversarial training without or with graph self-supervision with $n_{\text{per}} = 4$.

Attacks	Links	Feats	Links & Feats
GCN	9.31 ± 0.37	41.68 ± 0.85	1.26 ± 0.10
AdvT	27.30 ± 1.16	71.68 ± 1.66	13.10 ± 1.54
AdvT+Clu	29.08 ± 1.65	71.08 ± 1.20	15.44 ± 1.90
AdvT+Par	29.77 ± 1.86	72.78 ± 1.20	15.30 ± 1.86
AdvT+Comp	33.98 ± 1.97	73.84 ± 0.92	20.44 ± 2.53

Table 14: Adversarial attacks on Citeseer using adversarial training without or with graph self-supervision with $n_{\text{per}} = 1$.

Attacks	Links	Feats	Links & Feats
GCN	31.46 ± 1.43	31.56 ± 0.62	9.20 ± 0.62
AdvT	57.21 ± 1.80	63.92 ± 0.91	45.11 ± 1.93
AdvT+Clu	57.95 ± 1.71	64.30 ± 0.78	45.74 ± 1.63
AdvT+Par	58.24 ± 1.75	64.50 ± 0.67	46.94 ± 1.93
AdvT+Comp	57.57 ± 1.68	64.24 ± 0.86	26.15 ± 1.93

Table 15: Adversarial attacks on Citeseer using adversarial training without or with graph self-supervision with $n_{\text{per}} = 2$.

Attacks	Links	Feats	Links & Feats
GCN	13.68 ± 1.09	22.08 ± 0.73	3.08 ± 0.17
AdvT	39.32 ± 2.39	63.12 ± 0.62	26.20 ± 2.09
AdvT+Clu	40.32 ± 1.73	63.67 ± 0.45	27.02 ± 1.29
AdvT+Par	41.05 ± 1.91	64.06 ± 0.24	28.70 ± 1.60
AdvT+Comp	40.42 ± 2.09	63.50 ± 0.31	27.16 ± 1.69

Table 16: Adversarial attacks on Citeseer using adversarial training without or with graph self-supervision with $n_{\text{per}} = 3$.

Attacks	Links	Feats	Links & Feats
GCN	6.10 ± 0.52	19.44 ± 0.92	1.44 ± 0.22
AdvT	25.20 ± 2.43	60.30 ± 0.62	13.82 ± 1.70
AdvT+Clu	26.02 ± 1.73	61.02 ± 0.71	14.54 ± 1.15
AdvT+Par	26.78 ± 1.55	62.02 ± 0.67	15.74 ± 0.98
AdvT+Comp	26.15 ± 1.93	61.04 ± 0.78	14.68 ± 1.22

References

Chen, J., Ma, T., and Xiao, C. Fastgcn: fast learning with graph convolutional networks via importance sampling.

Table 17: Adversarial attacks on Citeseer using adversarial training without or with graph self-supervision with $n_{\text{per}} = 4$.

Attacks	Links	Feats	Links & Feats
GCN	3.68 ± 0.19	18.56 ± 1.10	0.76 ± 0.10
AdvT	15.62 ± 1.69	58.13 ± 1.43	7.82 ± 1.12
AdvT+Clu	16.38 ± 1.33	58.96 ± 1.00	8.38 ± 0.73
AdvT+Par	16.96 ± 1.16	59.90 ± 1.17	9.01 ± 0.59
AdvT+Comp	16.46 ± 1.41	58.89 ± 1.15	8.54 ± 0.82

arXiv preprint arXiv:1801.10247, 2018.

Karypis, G. and Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Sun, K., Zhu, Z., and Lin, Z. Multi-stage self-supervised learning for graph convolutional networks. *arXiv preprint arXiv:1902.11038*, 2019.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, 2018.