When Does Self-Supervision Help Graph Convolutional Networks?

Yuning You*, Tianlong Chen*, Zhangyang Wang, Yang Shen

Texas A&M University

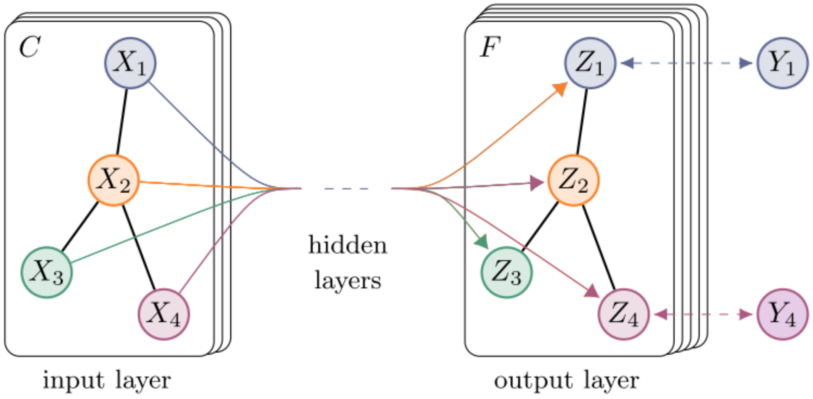
* Equal Contribution

Contents

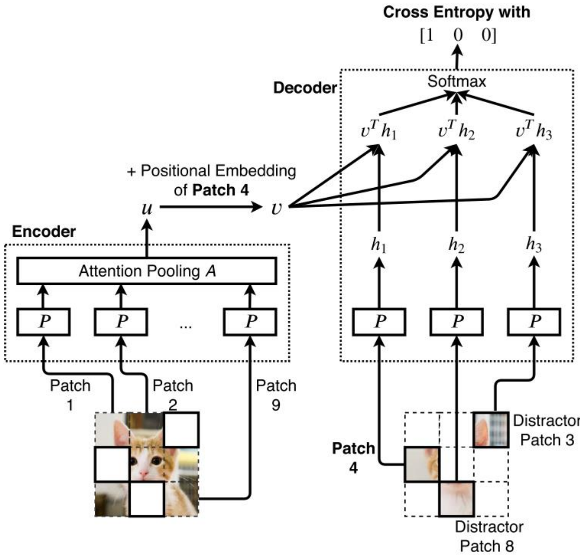
- **Introduction**
- Gap
- Overview of contributions
- Contribution 1. How to incorporate SS in GCNs?
- Contribution 2. How to design SS tasks to improve generalizability?
- Contribution 3. Does SS boost robustness?
- Conclusion

Introduction

- Graph convolutional networks (GCNs, ICLR'17):



- Self-supervision (SS) in images (e.g. Selfie, preprint'19):



Contents

- Introduction
- **Gap**
- Overview of contributions
- Contribution 1. How to incorporate SS in GCNs?
- Contribution 2. How to design SS tasks to improve generalizability?
- Contribution 3. Does SS boost robustness?
- Conclusion

Gap

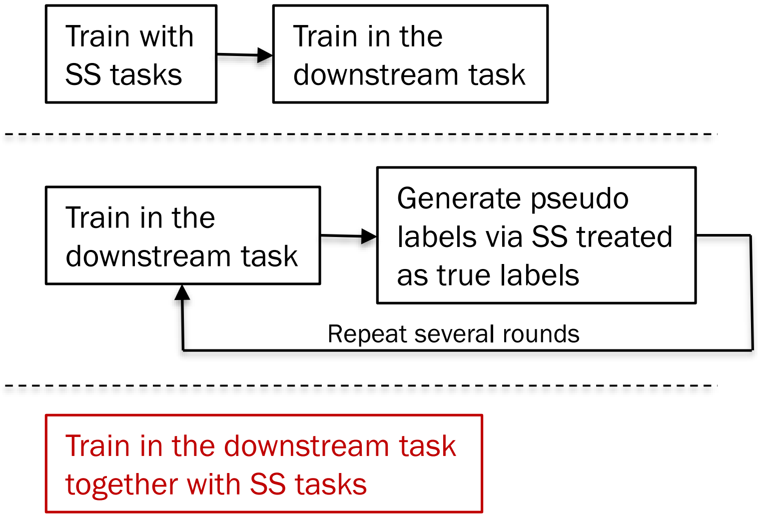
- Semi-supervised learning is an important field of graph-based applications with abundant unlabeled data available;
- SS is a promising technique in the few-shot scenario (of the computer vision domain) via using unlabeled data;
- SS in GCNs is still under-explored with an exception (M3S, AAAI’19).

Contents

- Introduction
- Gap
- **Overview of contributions**
- Contribution 1. How to incorporate SS in GCNs?
- Contribution 2. How to design SS tasks to improve generalizability?
- Contribution 3. Does SS boost robustness?
- Conclusion

Overview of contributions

- We perform a systematic study on SS + GCNs:
  - 1. How to incorporate SS in GCNs?
    - Pretraining & finetuning;
    - Self-training (M3S, AAAI’19);
    - Multi-task learning.



Overview of contributions

- We perform a systematic study on SS + GCNs:
  - 2. How to design SS tasks to improve generalizability?
    - We investigate three SS tasks: node feature clustering, graph partitioning and graph completion;
    - We illustrate that different SS tasks benefit generalizability in different cases.

Overview of contributions

- We perform a systematic study on SS + GCNs:
  - 3. Does SS boost robustness?
    - We generalize SS into adversarial training;
    - We show SS also improves GCN robustness without requiring larger models nor additional data.

Contents

- Introduction
- Gap
- Overview of contributions
- **Contribution 1. How to incorporate SS in GCNs?**
- Contribution 2. How to design SS tasks to improve generalizability?
- Contribution 3. Does SS boost robustness?
- Conclusion

Contribution 1. How to incorporate SS in GCNs?

- Pretraining & finetuning:

Train with SS tasks → Train in the downstream task

  – Little performance gains on a large dataset PubMed;

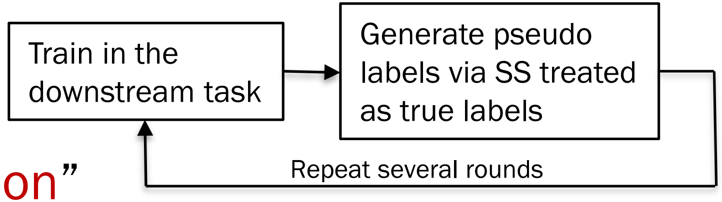
  – Conjecture: The performance behavior is due to "switching" the loss function of training shallow GCNs from pretraining to finetuning;

  – Shallow GCNs are easily "overwritten" after loss-function switching.

Table 1: Comparing performances of GCN through pretraining & finetuning (P&F) and multi-task learning (MTL) with graph partitioning (see Section 3.3) on the PubMed dataset. Reported numbers correspond to classification accuracy in percent.

Pipeline	GCN	P&F	MTL
Accuracy	79.10 ± 0.21	79.19 ± 0.21	80.00 ± 0.74

Contribution 1. How to incorporate SS in GCNs?



- Self-training (M3S, AAAI'19):
  - Performance gain encounters "saturation" as the label rate grows higher;
  - Conjecture: pseudo labels are assigned based on their proximity to labeled nodes in embeddings;
  - Less general (in pseudo labels) compared with multi-task learning.

Table 2: Experiments for GCN through M3S. Gray numbers are from (Sun et al., 2019).

Label Rate	0.03%	0.1%	0.3% (Conventional dataset split)
GCN	51.1	67.5	79.10 ± 0.21
M3S	59.2	70.6	79.28 ± 0.30

Contribution 1. How to incorporate SS in GCNs?

- Multi-task learning:
  - Empirically outperforms other two schemes;
  - We regard the SS task as a regularization term throughout the network training;
  - Act as a data-driven regularizer.

Train in the downstream task together with SS tasks

Table 1: Comparing performances of GCN through pretraining & finetuning (P&F) and multi-task learning (MTL) with graph partitioning (see Section 3.3) on the PubMed dataset. Reported numbers correspond to classification accuracy in percent.

Pipeline	GCN	P&F	MTL
Accuracy	79.10 ± 0.21	79.19 ± 0.21	80.00 ± 0.74

TEXAS A&M UNIVERSITY Engineering

Contents

Contribution 2. How to design SS tasks to improve generalizability?

TEXAS A&M UNIVERSITY Engineering

Table 3: Overview of three self-supervised tasks.

Task	Relied Feature	Primary Assumption	Type
Clustering	Nodes	Feature Similarity	Classification
Partitioning	Edges	Connection Density	Classification
Completion	Nodes & Edges	Context based Representation	Regression

- We investigate three SS tasks:

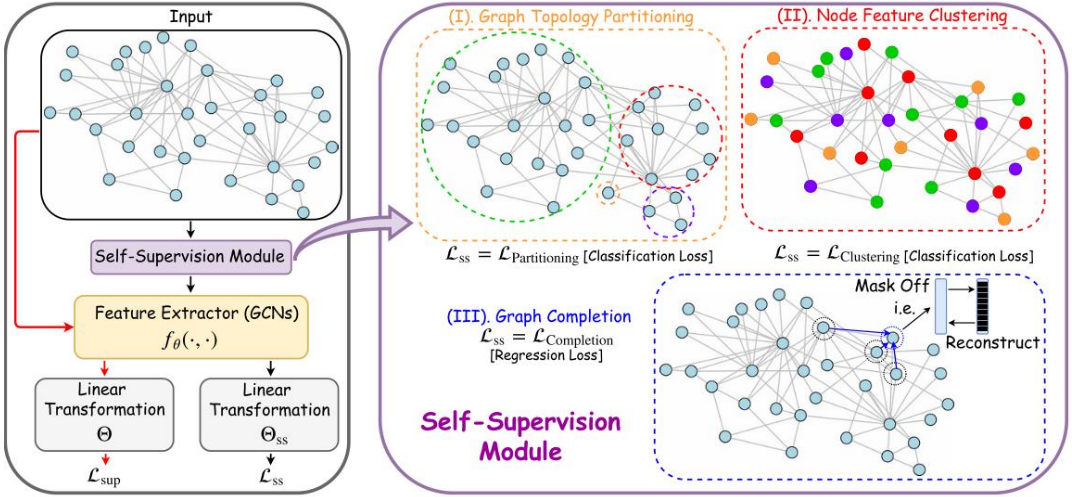


Figure 1: The overall framework for self-supervision on GCN through *multi-task learning*. The target task and auxiliary self-supervised tasks share the same feature extractor $f_\theta(\cdot,\cdot)$ with their individual linear transformation parameters Θ, Θ_{ss}.

Contribution 2. Whether the design of SS tasks matter?

- We illustrate that different SS tasks benefit generalizability in different cases:
  - Clu(stering) assumes that feature similarity implies target-label similarity;
  - Challenged in large datasets with low feature dimensions (such as PubMed).

Table 6: Experiments on SOTAs (GCN, GAT, GIN, GMNN, and GraphMix) with multi-task self-supervision. Red numbers indicate the best two performances for each SOTA.

Datasets	Cora	Citeseer	PubMed
GCN	81.00 ± 0.67	70.85 ± 0.70	79.10 ± 0.21
GCN+Clu	81.57 ± 0.59	70.73 ± 0.84	78.79 ± 0.36
GCN+Par	81.83 ± 0.65	71.34 ± 0.69	80.00 ± 0.74
GCN+Comp	81.03 ± 0.68	71.66 ± 0.48	79.14 ± 0.28
GAT	77.66 ± 1.08	68.90 ± 1.07	78.05 ± 0.46
GAT+Clu	79.40 ± 0.73	69.88 ± 1.13	77.80 ± 0.28
GAT+Par	80.11 ± 0.84	69.76 ± 0.81	80.11 ± 0.34
GAT+Comp	80.47 ± 1.22	70.62 ± 1.26	77.10 ± 0.67
GIN	77.27 ± 0.52	68.83 ± 0.40	77.38 ± 0.59
GIN+Clu	78.43 ± 0.80	68.86 ± 0.91	76.71 ± 0.36
GIN+Par	81.83 ± 0.58	71.50 ± 0.44	80.28 ± 1.34
GIN+Comp	76.62 ± 1.17	68.71 ± 1.01	78.70 ± 0.69
GMNN	83.28 ± 0.81	72.83 ± 0.72	81.34 ± 0.59
GMNN+Clu	83.49 ± 0.65	73.13 ± 0.72	79.45 ± 0.76
GMNN+Par	83.51 ± 0.50	73.62 ± 0.65	80.92 ± 0.77
GMNN+Comp	83.31 ± 0.81	72.93 ± 0.79	81.33 ± 0.59
GraphMix	83.91 ± 0.63	74.33 ± 0.65	80.68 ± 0.57
GraphMix+Clu	83.87 ± 0.56	75.16 ± 0.52	79.99 ± 0.82
GraphMix+Par	84.04 ± 0.57	74.93 ± 0.43	81.36 ± 0.33
GraphMix+Comp	83.76 ± 0.64	74.43 ± 0.72	80.82 ± 0.54

Contribution 2. Whether the design of SS tasks matter?

TEXAS A&M UNIVERSITY Engineering

- We illustrate that different SS tasks benefit generalizability in different cases:
  - Par(tition) assumes that connections in topology implies similarity in labels;
  - Safe for the three citation networks.

Table 6: Experiments on SOTAs (GCN, GAT, GIN, GMNN, and GraphMix) with multi-task self-supervision. Red numbers indicate the best two performances for each SOTA.

Datasets	Cora	Citeseer	PubMed
GCN	81.00 ± 0.67	70.85 ± 0.70	79.10 ± 0.21
GCN+Clu	81.57 ± 0.59	70.73 ± 0.84	78.79 ± 0.36
GCN+Par	81.83 ± 0.65	71.34 ± 0.69	80.00 ± 0.74
GCN+Comp	81.03 ± 0.68	71.66 ± 0.48	79.14 ± 0.28
GAT	77.66 ± 1.08	68.90 ± 1.07	78.05 ± 0.46
GAT+Clu	79.40 ± 0.73	69.88 ± 1.13	77.80 ± 0.28
GAT+Par	80.11 ± 0.84	69.76 ± 0.81	80.11 ± 0.34
GAT+Comp	80.47 ± 1.22	70.62 ± 1.26	77.10 ± 0.67
GIN	77.27 ± 0.52	68.83 ± 0.40	77.38 ± 0.59
GIN+Clu	78.43 ± 0.80	68.86 ± 0.91	76.71 ± 0.36
GIN+Par	81.83 ± 0.58	71.50 ± 0.44	80.28 ± 1.34
GIN+Comp	76.62 ± 1.17	68.71 ± 1.01	78.70 ± 0.69
GMNN	83.28 ± 0.81	72.83 ± 0.72	81.34 ± 0.59
GMNN+Clu	83.49 ± 0.65	73.13 ± 0.72	79.45 ± 0.76
GMNN+Par	83.51 ± 0.50	73.62 ± 0.65	80.92 ± 0.77
GMNN+Comp	83.31 ± 0.81	72.93 ± 0.79	81.33 ± 0.59
GraphMix	83.91 ± 0.63	74.33 ± 0.65	80.68 ± 0.57
GraphMix+Clu	83.87 ± 0.56	75.16 ± 0.52	79.99 ± 0.82
GraphMix+Par	84.04 ± 0.57	74.93 ± 0.43	81.36 ± 0.33
GraphMix+Comp	83.76 ± 0.64	74.43 ± 0.72	80.82 ± 0.54

Contribution 2. Whether the design of SS tasks matter?

- We illustrate that different SS tasks benefit generalizability in different cases:
  - Comp(letion) assumes feature similarity or smoothness in small neighborhoods;
  - Improve performance for datasets with small neighborhoods (such as Citeseer).

Table 6: Experiments on SOTAs (GCN, GAT, GIN, GMNN, and GraphMix) with multi-task self-supervision. Red numbers indicate the best two performances for each SOTA.

Datasets	Cora	Citeseer	PubMed
GCN	81.00 ± 0.67	70.85 ± 0.70	79.10 ± 0.21
GCN+Clu	81.57 ± 0.59	70.73 ± 0.84	78.79 ± 0.36
GCN+Par	81.83 ± 0.65	71.34 ± 0.69	80.00 ± 0.74
GCN+Comp	81.03 ± 0.68	71.66 ± 0.48	79.14 ± 0.28
GAT	77.66 ± 1.08	68.90 ± 1.07	78.05 ± 0.46
GAT+Clu	79.40 ± 0.73	69.88 ± 1.13	77.80 ± 0.28
GAT+Par	80.11 ± 0.84	69.76 ± 0.81	80.11 ± 0.34
GAT+Comp	80.47 ± 1.22	70.62 ± 1.26	77.10 ± 0.67
GIN	77.27 ± 0.52	68.83 ± 0.40	77.38 ± 0.59
GIN+Clu	78.43 ± 0.80	68.86 ± 0.91	76.71 ± 0.36
GIN+Par	81.83 ± 0.58	71.50 ± 0.44	80.28 ± 1.34
GIN+Comp	76.62 ± 1.17	68.71 ± 1.01	78.70 ± 0.69
GMNN	83.28 ± 0.81	72.83 ± 0.72	81.34 ± 0.59
GMNN+Clu	83.49 ± 0.65	73.13 ± 0.72	79.45 ± 0.76
GMNN+Par	83.51 ± 0.50	73.62 ± 0.65	80.92 ± 0.77
GMNN+Comp	83.31 ± 0.81	72.93 ± 0.79	81.33 ± 0.59
GraphMix	83.91 ± 0.63	74.33 ± 0.65	80.68 ± 0.57
GraphMix+Clu	83.87 ± 0.56	75.16 ± 0.52	79.99 ± 0.82
GraphMix+Par	84.04 ± 0.57	74.93 ± 0.43	81.36 ± 0.33
GraphMix+Comp	83.76 ± 0.64	74.43 ± 0.72	80.82 ± 0.54

Contribution 2. Whether the design of SS tasks matter?

TEXAS A&M UNIVERSITY
Engineering

- We illustrate that different SS tasks benefit generalizability in different cases:
  - Architectures also affect performances;
  - Architectures with weaker priors have seen more improvement from SS.

Table 6: Experiments on SOTAs (GCN, GAT, GIN, GMNN, and GraphMix) with multi-task self-supervision. Red numbers indicate the best two performances for each SOTA.

Datasets	Cora	Citeseer	PubMed
GCN	81.00 ± 0.67	70.85 ± 0.70	79.10 ± 0.21
GCN+Clu	81.57 ± 0.59	70.73 ± 0.84	78.79 ± 0.36
GCN+Par	81.83 ± 0.65	71.34 ± 0.69	80.00 ± 0.74
GCN+Comp	81.03 ± 0.68	71.66 ± 0.48	79.14 ± 0.28
GAT	77.66 ± 1.08	68.90 ± 1.07	78.05 ± 0.46
GAT+Clu	79.40 ± 0.73	69.88 ± 1.13	77.80 ± 0.28
GAT+Par	80.11 ± 0.84	69.76 ± 0.81	80.11 ± 0.34
GAT+Comp	80.47 ± 1.22	70.62 ± 1.26	77.10 ± 0.67
GIN	77.27 ± 0.52	68.83 ± 0.40	77.38 ± 0.59
GIN+Clu	78.43 ± 0.80	68.86 ± 0.91	76.71 ± 0.36
GIN+Par	81.83 ± 0.58	71.50 ± 0.44	80.28 ± 1.34
GIN+Comp	76.62 ± 1.17	68.71 ± 1.01	78.70 ± 0.69
GMNN	83.28 ± 0.81	72.83 ± 0.72	81.34 ± 0.59
GMNN+Clu	83.49 ± 0.65	73.13 ± 0.72	79.45 ± 0.76
GMNN+Par	83.51 ± 0.50	73.62 ± 0.65	80.92 ± 0.77
GMNN+Comp	83.31 ± 0.81	72.93 ± 0.79	81.33 ± 0.59
GraphMix	83.91 ± 0.63	74.33 ± 0.65	80.68 ± 0.57
GraphMix+Clu	83.87 ± 0.56	75.16 ± 0.52	79.99 ± 0.82
GraphMix+Par	84.04 ± 0.57	74.93 ± 0.43	81.36 ± 0.33
GraphMix+Comp	83.76 ± 0.64	74.43 ± 0.72	80.82 ± 0.54

Contents

- Introduction
- Gap
- Overview of contributions
- Contribution 1. How to incorporate SS in GCNs?
- Contribution 2. How to design SS tasks to improve generalizability?
- **Contribution 3. Does SS boost robustness?**
- Conclusion

Contribution 3. Does SS boost robustness?

- We generalize SS into adversarial training:
  - Adversarial training:

$$\boldsymbol{Z} = f_\theta(\boldsymbol{X}, \hat{\boldsymbol{A}})\boldsymbol{\Theta}, \quad \boldsymbol{Z}' = f_\theta(\boldsymbol{X}', \boldsymbol{A}')\boldsymbol{\Theta},$$
$$\theta^*, \boldsymbol{\Theta}^* = \arg\min_{\theta, \boldsymbol{\Theta}} \big(\mathcal{L}_{\text{sup}}(\theta, \boldsymbol{\Theta}) + \alpha_3 \mathcal{L}_{\text{adv}}(\theta, \boldsymbol{\Theta})\big), \quad (6)$$

  - SS + Adversarial training:

$$\boldsymbol{Z} = f_\theta(\boldsymbol{X}, \hat{\boldsymbol{A}})\boldsymbol{\Theta}, \quad \boldsymbol{Z}' = f_\theta(\boldsymbol{X}', \boldsymbol{A}')\boldsymbol{\Theta},$$
$$\boldsymbol{Z}_{\text{ss}} = f_\theta(\boldsymbol{X}_{\text{ss}}, \boldsymbol{A}_{\text{ss}})$$
$$\theta^*, \boldsymbol{\Theta}^*, \boldsymbol{\Theta}^*_{\text{ss}} = \arg\min_{\theta, \boldsymbol{\Theta}, \boldsymbol{\Theta}_{\text{ss}}} \big(\alpha_1 \mathcal{L}_{\text{sup}}(\theta, \boldsymbol{\Theta}) + \alpha_2 \mathcal{L}_{\text{ss}}(\theta, \boldsymbol{\Theta}_{\text{ss}}) + \alpha_3 \mathcal{L}_{\text{adv}}(\theta, \boldsymbol{\Theta})\big), \quad (7)$$

Contribution 3. Does SS boost robustness?

- We show that SS also improves GCN robustness without requiring larger models or additional data.
  - Clu is more effective against feature attacks;
  - Par is more effective against links attacks;

Table 7: Adversarial defense performances on Cora using adversarial training (abbr. AdvT) without or with graph self-supervision. Attacks include those on links, features (abbr. Feats), and both. Red numbers indicate the best two performances in each attack scenario (node classification accuracy; unit: %).

Attacks	None	Links	Feats	Links & Feats
GCN	80.61 ± 0.21	28.72 ± 0.63	44.06 ± 1.23	8.18 ± 0.27
AdvT	80.24 ± 0.74	54.58 ± 2.57	75.25 ± 1.26	39.08 ± 3.05
AdvT+Clu	80.26 ± 0.99	55.54 ± 3.19	76.24 ± 0.99	41.84 ± 3.48
AdvT+Par	80.42 ± 0.76	56.36 ± 2.57	75.88 ± 0.72	41.57 ± 3.47
AdvT+Comp	79.64 ± 0.99	59.05 ± 3.29	76.04 ± 0.68	47.14 ± 3.01

Table 8: Adversarial defense performances on Citeseer using adversarial training without or with graph self-supervision.

Attacks	None	Links	Feats	Links & Feats
GCN	71.05 ± 0.56	13.68 ± 1.09	22.08 ± 0.73	3.08 ± 0.17
AdvT	69.98 ± 1.03	39.32 ± 2.39	63.12 ± 0.62	26.20 ± 2.09
AdvT+Clu	70.13 ± 0.81	40.32 ± 1.73	63.67 ± 0.45	27.02 ± 1.29
AdvT+Par	69.96 ± 0.77	41.05 ± 1.91	64.06 ± 0.24	28.70 ± 1.60
AdvT+Comp	69.98 ± 0.82	40.42 ± 2.09	63.50 ± 0.31	27.16 ± 1.69

Contribution 3. Does SS boost robustness?

- We show that SS also improves GCN robustness without requiring larger models or additional data.
  - Strikingly, Comp significantly boosts robustness against link attacks and link & feature attacks on Cora.

Table 7: Adversarial defense performances on Cora using adversarial training (abbr. AdvT) without or with graph self-supervision. Attacks include those on links, features (abbr. Feats), and both. Red numbers indicate the best two performances in each attack scenario (node classification accuracy; unit: %).

Attacks	None	Links	Feats	Links & Feats
GCN	80.61 ± 0.21	28.72 ± 0.63	44.06 ± 1.23	8.18 ± 0.27
AdvT	80.24 ± 0.74	54.58 ± 2.57	75.25 ± 1.26	39.08 ± 3.05
AdvT+Clu	80.26 ± 0.99	55.54 ± 3.19	76.24 ± 0.99	41.84 ± 3.48
AdvT+Par	80.42 ± 0.76	56.36 ± 2.57	75.88 ± 0.72	41.57 ± 3.47
AdvT+Comp	79.64 ± 0.99	59.05 ± 3.29	76.04 ± 0.68	47.14 ± 3.01

Table 8: Adversarial defense performances on Citeseer using adversarial training without or with graph self-supervision.

Attacks	None	Links	Feats	Links & Feats
GCN	71.05 ± 0.56	13.68 ± 1.09	22.08 ± 0.73	3.08 ± 0.17
AdvT	69.98 ± 1.03	39.32 ± 2.39	63.12 ± 0.62	26.20 ± 2.09
AdvT+Clu	70.13 ± 0.81	40.32 ± 1.73	63.67 ± 0.45	27.02 ± 1.29
AdvT+Par	69.96 ± 0.77	41.05 ± 1.91	64.06 ± 0.24	28.70 ± 1.60
AdvT+Comp	69.98 ± 0.82	40.42 ± 2.09	63.50 ± 0.31	27.16 ± 1.69

Contents

Conclusion

TEXAS A&M UNIVERSITY Engineering

- We demonstrate the effectiveness of incorporating self-supervised learning in GCNs through multi-task learning;
- We illustrate that appropriately designed multi-task self-supervision tasks benefit GCN generalizability in different cases;
- We show that multi-task self-supervision also improves robustness against attacks, without requiring larger models or additional data.

TEXAS A&M UNIVERSITY

Engineering

Thank you for listening.