Meta Propagation Networks for Graph Few-shot Semi-supervised Learning（AAAI_2022）

Kaize Ding , Jianling Wang , James Caverlee , and Huan Liu
Arizona State University
Texas A&M University
{kaize.ding, huan.liu}@asu.edu
{jlwang, caverlee}@tamu.edu

2022. 3. 13 • ChongQing

gesis
Leibniz-Institut
für Sozialwissenschaften

Reported by Lele Duan

Code&datasets: https://github.com/kaize0409/Meta-PN

Leibniz-Institut für Sozialwissenschaften

Background

- Few-shot semi-supervised node classification : only few labeled nodes per class are available.
  - Existing GNNs developed for semi-supervised node classification predominantly assume that the provided gold-labeled nodes are relatively abundant.
  - Overfitting and oversmoothing.
  - No auxiliary knowledge.

- Solution:
  - Inferring optimal pseudo labels on unlabeled nodes.

Method

➢ Over view

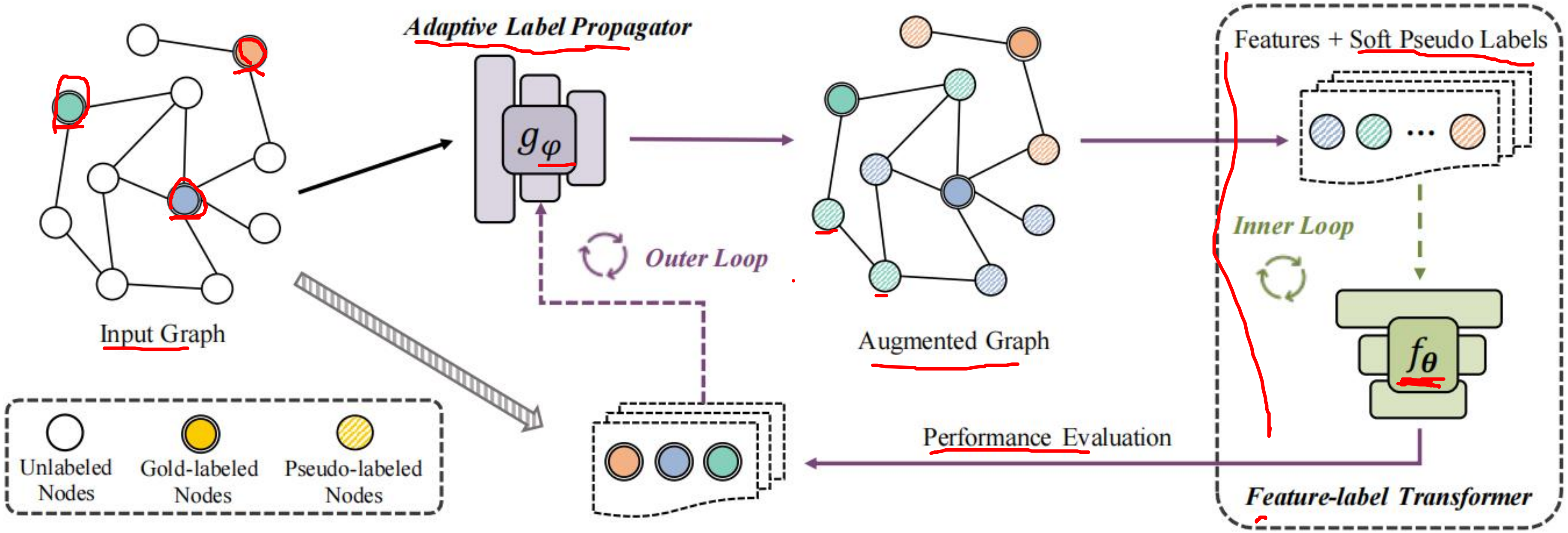


Figure 1: Illustration of our Meta-PN framework. The *adaptive label propagator* propagates known labels to unlabeled nodes and the *feature-label transformer* transforms the features of each node to a soft label vector. Specifically, the *adaptive label propagator* is meta-learned to adjust its label propagation strategy to infer accurate pseudo labels on unlabeled nodes, according to the *feature-label transformer*'s performance change on the labeled nodes. We omit the node features for simplicity.

$G = (\mathcal{V}, \mathcal{E})$ $\quad \mathbf{X} \in \mathbb{R}^{n \times f}$ $\quad \mathbf{A} \in \{0,1\}^{n \times n}$

$\tilde{\mathbf{A}}_{sym} = \underbrace{\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}}_{}$ $\quad \mathbf{Y} \in \mathbb{R}^{n \times c}$

A+I

Adaptive Label Propagator (Meta Learner)

$$\hat{\mathbf{Y}}_{i,:} = \sum_{k=0}^{K} \gamma_{ik} \mathbf{Y}_{i,:}^{(k)}, \mathbf{Y}^{(k+1)} = \mathbf{T}\mathbf{Y}^{(k)}, \tag{2}$$

$$\gamma_{ik} = \frac{\exp\left(\mathbf{a}^{\mathrm{T}} \mathrm{ReLU}\left(\mathbf{W}\mathbf{Y}_{i,:}^{(k)}\right)\right)}{\sum_{k'=0}^{K} \exp\left(\mathbf{a}^{\mathrm{T}} \mathrm{ReLU}\left(\mathbf{W}\mathbf{Y}_{i,:}^{(k')}\right)\right)}, \tag{3}$$

where $\mathbf{a} \in \mathbb{R}^{c}$ is the attention vector and $\mathbf{W} \in \mathbb{R}^{c \times c}$ is a weight matrix. By setting the attention vector and weight

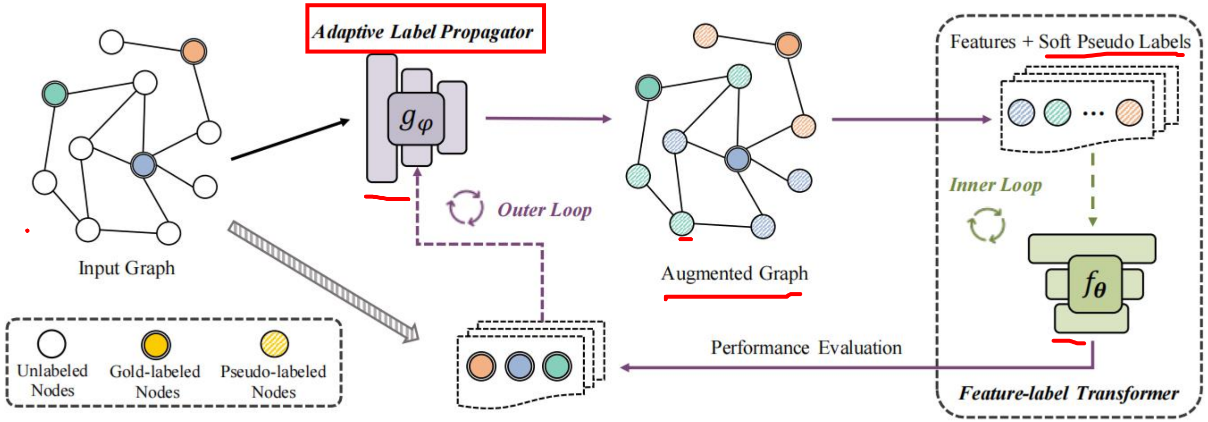


Figure 1: Illustration of our Meta-PN framework. The *adaptive label propagator* propagates known labels to unlabeled nodes and the *feature-label transformer* transforms the features of each node to a soft label vector. Specifically, the *adaptive label propagator* is meta-learned to adjust its label propagation strategy to infer accurate pseudo labels on unlabeled nodes, according to the *feature-label transformer*'s performance change on the labeled nodes. We omit the node features for simplicity.

Feature-label Transformer (Target Model)

$$\mathbf{P}_{i,:} = f_{\boldsymbol{\theta}}(\mathbf{X}_{i,:}), \tag{4}$$

where $f_{\boldsymbol{\theta}}(\cdot)$ is a multi-layer perceptron (MLP) followed by a softmax function.

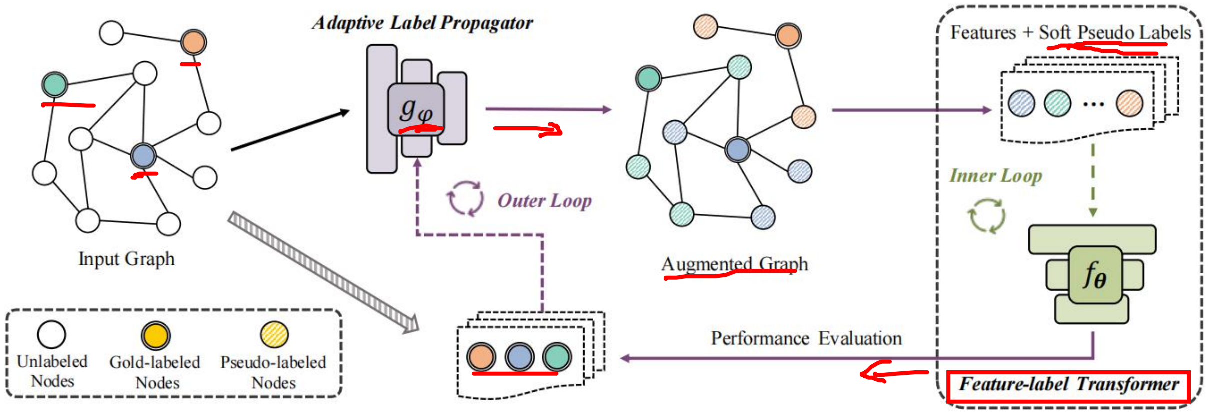


Figure 1: Illustration of our Meta-PN framework. The *adaptive label propagator* propagates known labels to unlabeled nodes and the *feature-label transformer* transforms the features of each node to a soft label vector. Specifically, the *adaptive label propagator* is meta-learned to adjust its label propagation strategy to infer accurate pseudo labels on unlabeled nodes, according to the *feature-label transformer*'s performance change on the labeled nodes. We omit the node features for simplicity.

Model Learning via Bi-level Optimization

Outer loop:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{v_i \in \mathcal{V}^L} [\mathcal{L}(f_{\boldsymbol{\theta}^*(\phi)}(\mathbf{X}_{i,:}), \mathbf{Y}_{i,:})], \tag{5}$$

Inner loop:

$$\boldsymbol{\theta}^*(\phi) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{v_i \in \mathcal{V}^U} [\mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{X}_{i,:}), g_\phi(\mathbf{Y}, \mathbf{A})_{i,:})].$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} J_{\text{pseudo}}(\boldsymbol{\theta}, \phi). \tag{6}$$

$$\phi' = \phi - \eta_\phi \nabla_\phi J_{\text{gold}}(\boldsymbol{\theta}'(\phi)). \tag{7}$$

Experiment

Table 1: Summary statistics of the evaluation datasets.

Dataset	# Nodes	# Edges	# Features	# Classes
Cora-ML	2,810	7,981	2,879	7
CiteSeer	2,110	3,668	3,703	6
PubMed	19,717	44,324	500	3
MS-CS	18,333	81,894	6,805	15
ogbn-arxiv	169,343	1,166,243	15	40

Experiment

Table 2: Test accuracy on few-shot semi-supervised node classification: mean accuracy (%) ± 95% confidence interval.

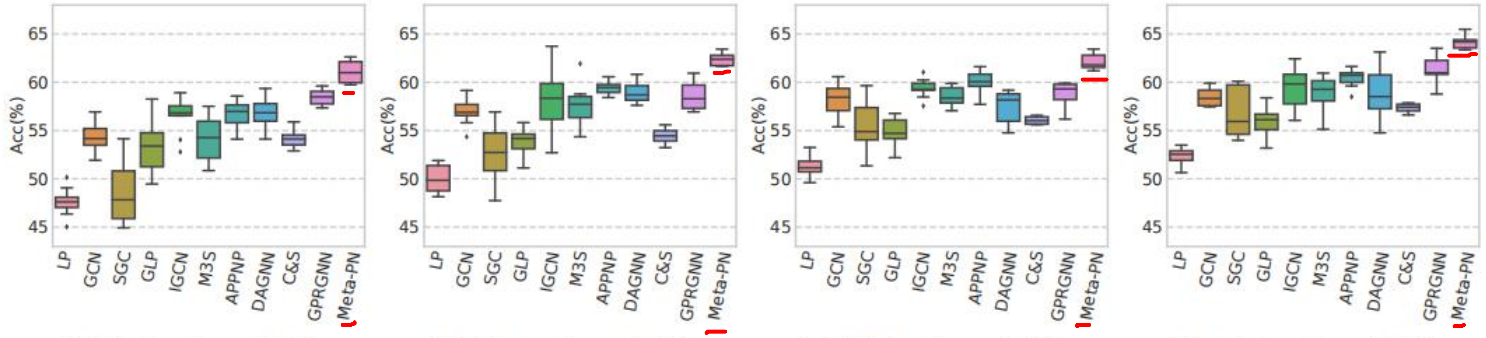
Method	Cora-ML		CiteSeer		PubMed		MS-CS	
	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot
MLP	41.07 ± 0.76	51.12 ± 0.61	43.34 ± 0.56	44.90 ± 0.60	56.59 ± 0.93	59.90 ± 0.84	70.33 ± 0.37	79.41 ± 0.31
LP	62.07 ± 0.71	68.01 ± 0.62	54.07 ± 0.59	55.73 ± 1.19	58.75 ± 0.89	59.91 ± 0.85	57.96 ± 0.69	62.98 ± 0.61
GCN	48.02 ± 0.89	67.32 ± 1.02	53.60 ± 0.86	62.60 ± 0.58	58.89 ± 0.80	65.77 ± 0.98	69.24 ± 0.94	84.43 ± 0.89
SGC	49.60 ± 0.55	67.24 ± 0.86	57.37 ± 0.98	61.55 ± 0.53	63.37 ± 0.93	64.93 ± 0.81	72.11 ± 0.76	87.51 ± 0.27
GLP	65.57 ± 0.26	71.26 ± 0.31	65.76 ± 0.49	71.36 ± 0.18	65.34 ± 0.54	65.26 ± 0.29	86.10 ± 0.21	86.94 ± 0.23
IGCN	66.60 ± 0.29	72.50 ± 0.20	67.47 ± 0.29	72.92 ± 0.10	62.28 ± 0.23	65.19 ± 0.13	85.83 ± 0.06	87.01 ± 0.05
M3S	64.66 ± 0.31	69.64 ± 0.18	65.12 ± 0.20	68.18 ± 0.18	63.40 ± 0.32	68.85 ± 0.26	84.96 ± 0.18	86.83 ± 0.29
APPNP	72.39 ± 0.98	78.32 ± 0.58	67.55 ± 0.77	71.08 ± 0.61	70.52 ± 0.62	74.24 ± 0.87	86.65 ± 0.42	90.13 ± 0.86
DAGNN	71.86 ± 0.75	77.20 ± 0.69	66.62 ± 0.27	70.55 ± 0.12	71.22 ± 0.82	73.91 ± 0.71	86.32 ± 0.57	90.30 ± 0.66
C&S	68.93 ± 0.68	73.37 ± 0.24	63.02 ± 0.72	64.72 ± 0.53	70.51 ± 0.57	73.22 ± 0.57	85.86 ± 0.45	87.99 ± 0.24
GPR-GNN	70.98 ± 0.84	75.18 ± 0.52	64.32 ± 0.81	65.28 ± 0.52	71.03 ± 0.73	74.08 ± 0.65	86.12 ± 0.37	90.29 ± 0.38
Meta-PN	**74.94 ± 0.25**	**79.88 ± 0.15**	**70.48 ± 0.34**	**74.14 ± 0.50**	**73.25 ± 0.77**	**77.78 ± 0.92**	**88.99 ± 0.29**	**91.31 ± 0.22**

Experiment

Table 3: Test accuracy on standard semi-supervised node classification: mean accuracy (%) ± 95% confidence interval.

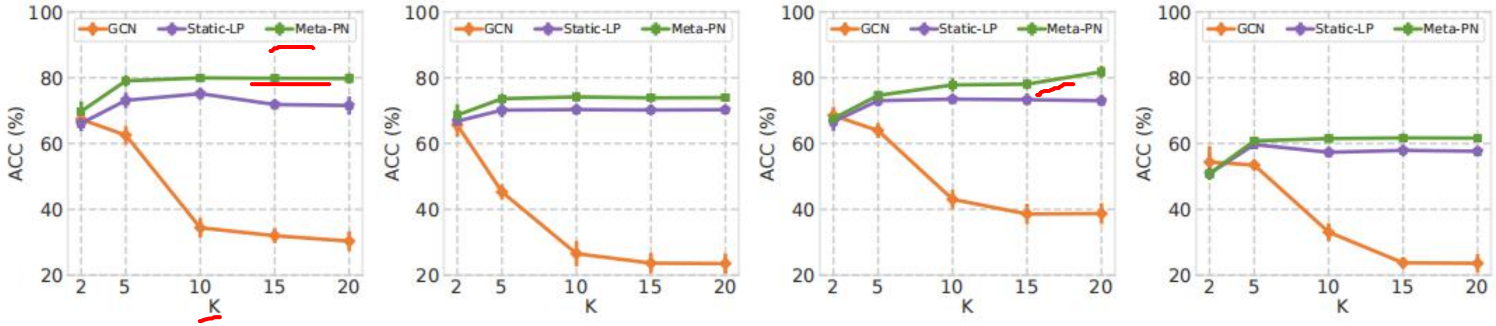
Method	Cora-ML	CiteSeer	PubMed	MS-CS
MLP	68.42 ± .34	63.98 ± .44	69.47 ± .47	88.30 ± .13
LP	75.74 ± .27	65.62 ± .43	69.82 ± .70	72.03 ± .25
GCN	82.70 ± .37	73.62 ± .39	76.84 ± .44	91.10 ± .20
SGC	75.97 ± .72	75.57 ± .28	71.24 ± .86	90.56 ± .14
GLP	81.67 ± .14	75.21 ± .14	78.95 ± .09	91.85 ± .04
IGCN	82.11 ± .09	75.22 ± .10	79.06 ± .07	91.60 ± .03
M3S	82.72 ± .13	73.73 ± .32	77.62 ± .11	91.08 ± .09
APPNP	85.09 ± .25	75.73 ± .30	79.73 ± .31	91.74 ± .16
DAGNN	85.65 ± .23	74.53 ± .17	79.59 ± .37	92.80 ± .17
C&S	83.18 ± .31	70.51 ± .24	77.10 ± .34	92.49 ± .19
GPR-GNN	83.53 ± .31	71.18 ± .25	79.62 ± .46	92.57 ± .21
Meta-PN	**86.33** ± **.36**	**77.13** ± **.31**	**80.39** ± **.53**	**93.92** ± **.17**

Experiment



(a) label ratio = 1.0% (b) label ratio = 1.5% (c) label ratio = 2.0% (d) label ratio = 2.5%

Figure 2: Comparison results on ogbn-arxiv w.r.t different size of training labels.



(a) Cora-ML (b) CiteSeer (c) PubMed (d) Ogbn-arxiv

Figure 3: Few-shot (i.e., 5-shot or 1.0% label ratio) evaluation on different datasets w.r.t. propagation steps (K).

Thank you!