Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

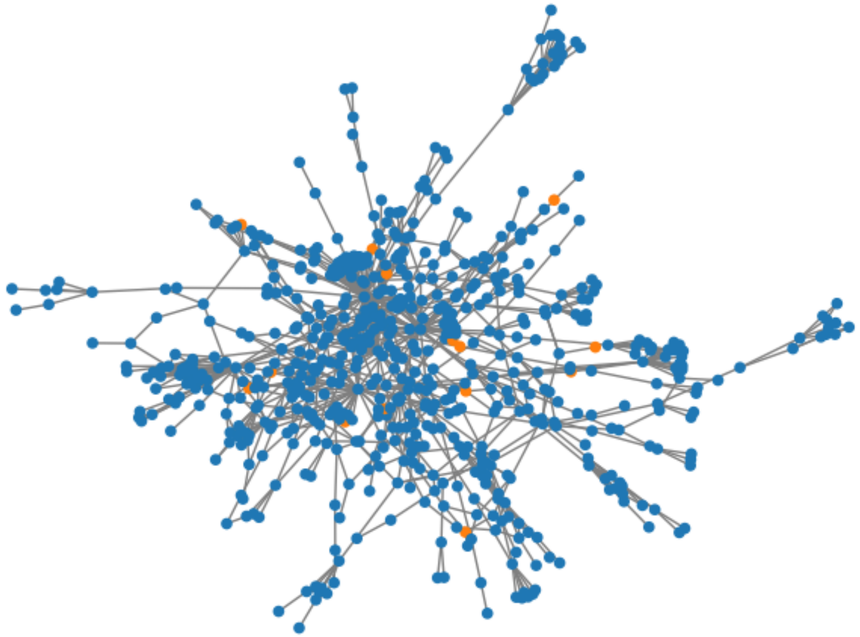
Qi Zhu[1], Natalia Ponomareva[2], Jiawei Han[1], Bryan Perozzi[2]

[1]University of Illinois at Urbana-Champaign
[2]Google Inc.

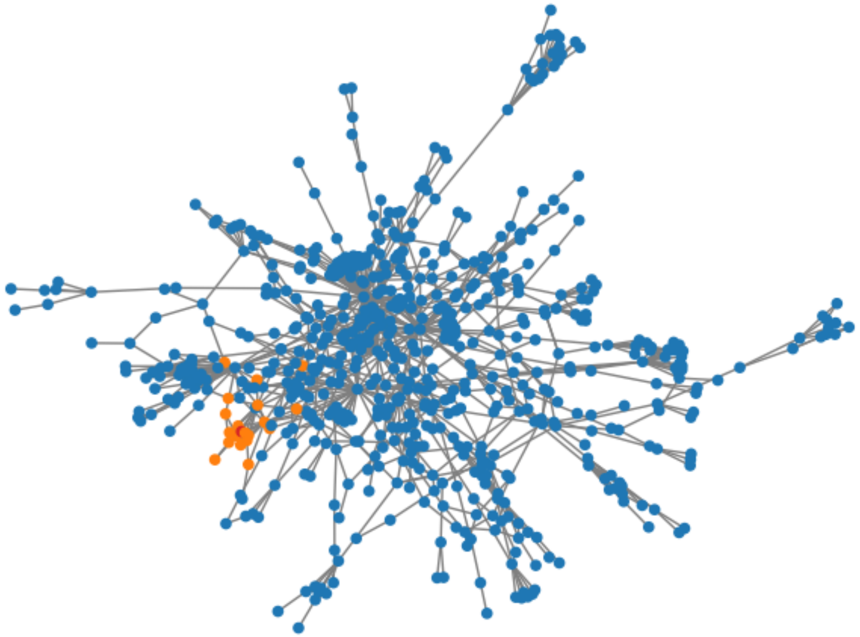
Overview

- What is localized training data ?
- Quantify the training bias
  - Distribution shift as domain adaptation
- Proposed Shift-Robust framework
  - Standard GNN models
  - Linearized GNN models
- Experiments
- Future work

IID vs. localized training data



IID training sample



localized training sample

Localized annotations in real-world

- Spam and abuse detection problems typically have very imbalanced label distribution (e.g., < 1% positive).
- Choosing the nodes to acquire labels in an IID manner is not feasible!
    - We want to have a reasonable amount of data points from the rare positive class.

Localized data is biased

- A general graph neural network layer, final representation Z = H^k

$$H^k = \sigma(\tilde{A} H^{k-1} \theta^k)$$

- To learn a semi-supervised classifier, cross-entropy loss function l is widely used

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} l(y_i, z_i),$$

- Data-shift [1] happens when the training data is biased from testing
  - $\mathbf{Pr}_{\text{train}}$ (X, Y) $\neq$ $\mathbf{Pr}_{\text{test}}$ (X, Y)
  - In a neural network, we care about the shift happens in the last hidden activated layer Z, i.e. $\mathbf{Pr}_{\text{train}}$ (Z, Y) $\neq$ $\mathbf{Pr}_{\text{test}}$ (Z, Y)
  - Standard learning theory assumes, $\mathbf{Pr}_{\text{train}}$ (Y|Z) = $\mathbf{Pr}_{\text{test}}$ (Y|Z), such that,

$$\Pr_{\text{train}}(Z, Y) \neq \Pr_{\text{test}}(Z, Y) \rightarrow \Pr_{\text{train}}(Z) \neq \Pr_{\text{test}}(Z)$$

[1] Quiñonero-Candela, Joaquin, et al., eds. *Dataset shift in machine learning*. Mit Press, 2009.

Overview

- What is localized training data ?
- Quantify the training bias
  - Distribution shift
- Proposed Shift-Robust framework
  - Standard GNN models
  - Linearized GNN models
- Experiments
- Future work

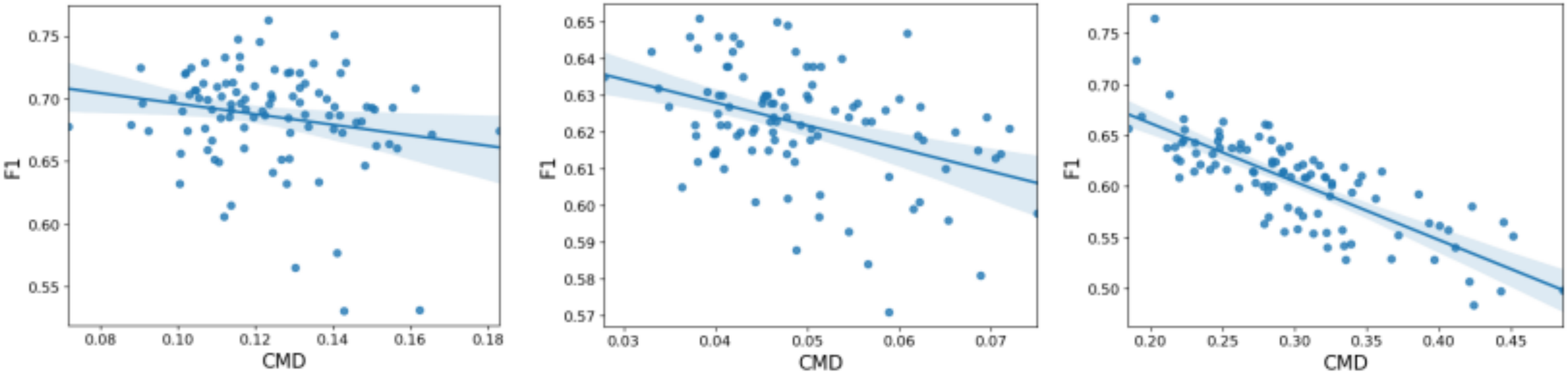
Quantify the distribution shift

- Assume two sets of representation vectors are generated by probability distribution p and q, a valid discrepancy metric measures the distribution shifts, CMD [1] for example,

$$\mathrm{CMD} = \frac{1}{|b-a|}\|\mathrm{E}(p) - \mathrm{E}(q)\|_2 + \sum_{k=2}^{\infty} \frac{1}{|b-a|^k}\|c_k(p) - c_k(q)\|_2,$$

[1] Zellinger, Werner, et al. "Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning." ICLR, 2016.

Negative effect of distribution shifts



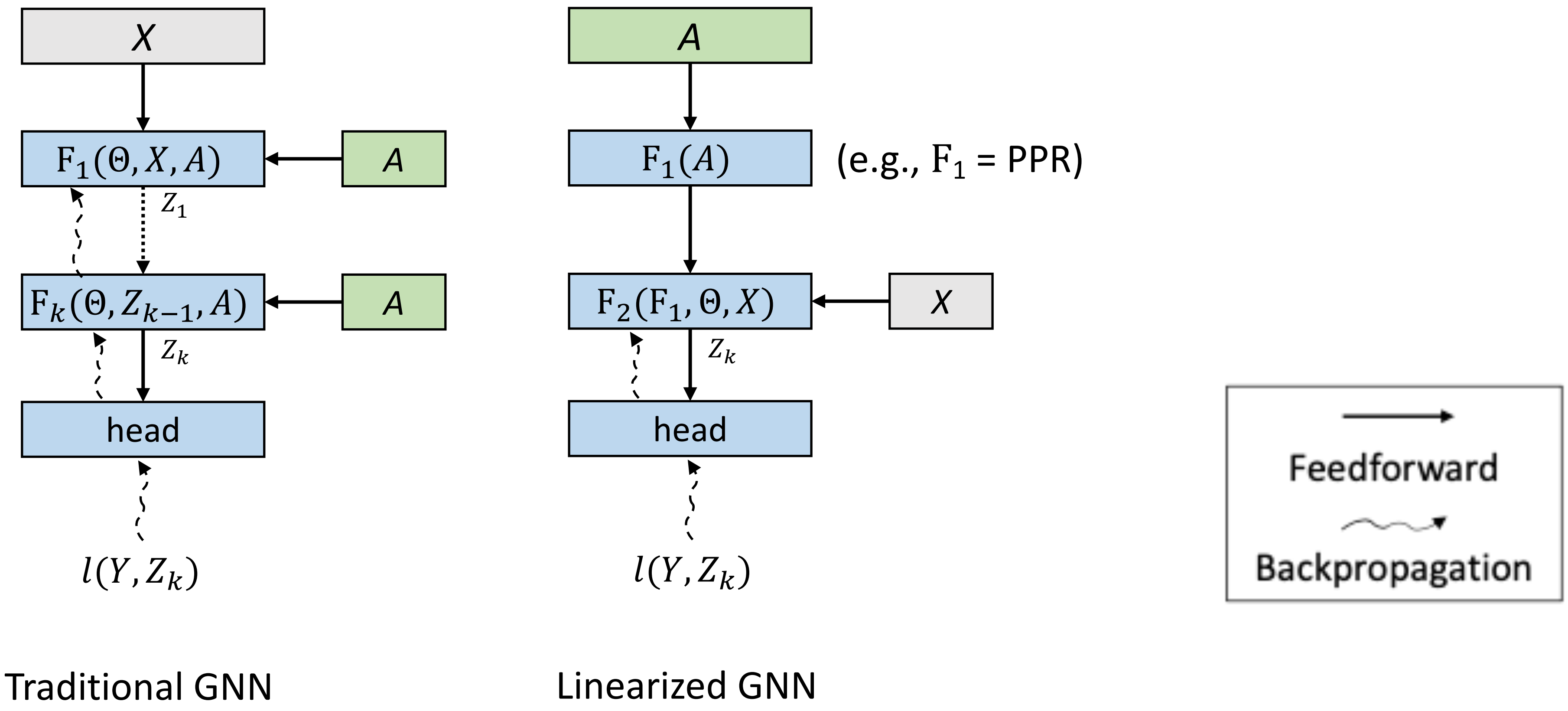
(a) Cora **(b)** Citeseer **(c)** Pubmed

Distribution shift (CMD) between training and testing data could be a good indicator of performance (F1) !

Overview

- What is localized training data ?
- Quantify the training bias
  - Distribution shift as domain adaption
- Proposed Shift-Robust framework
  - Standard GNN models
  - Linearized GNN models
- Experiments
- Future work

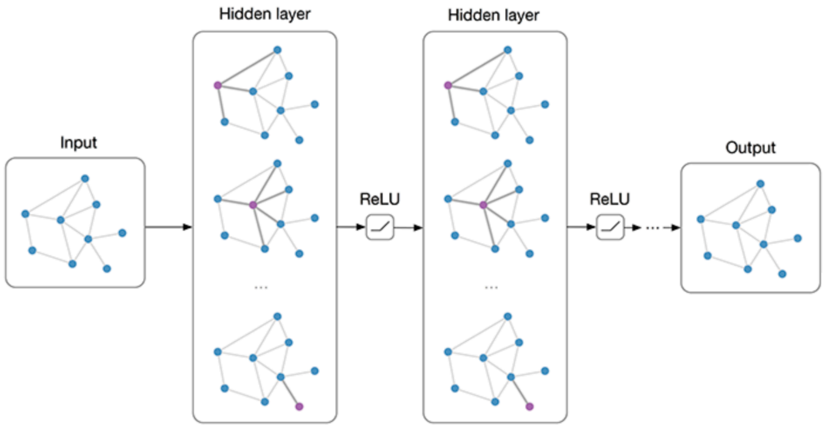
Two major variants of GNNs



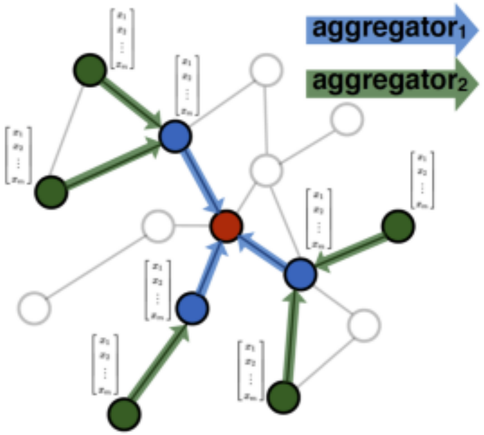
Standard GNNs: the graph inductive bias $\tilde{A}$ is differentiable

Linearized GNNs: the graph inductive bias $\tilde{A}$ is **not** differentiable

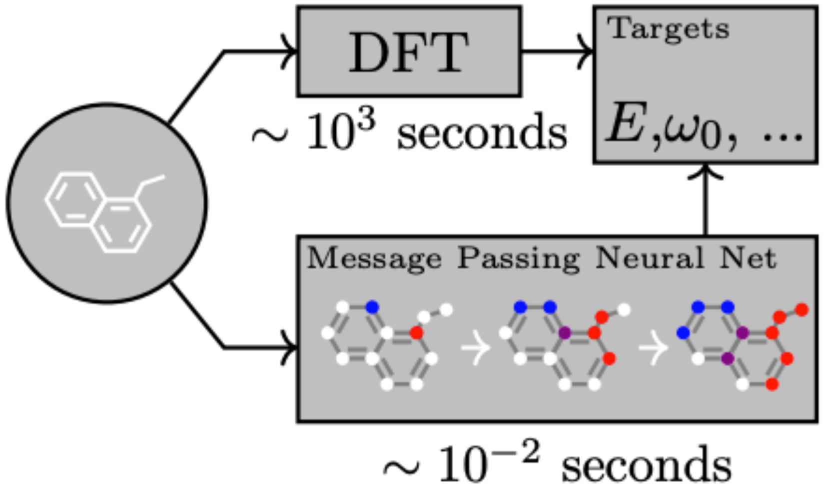
Examples of standard (deep) models



Graph Convolutional Networks [1]



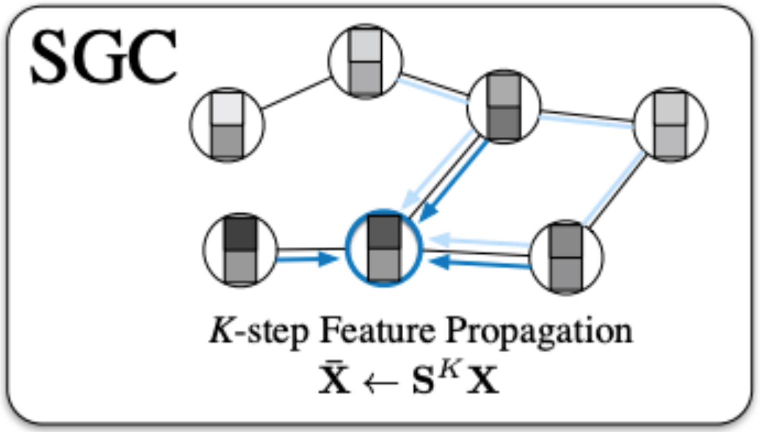
GraphSAGE [3]



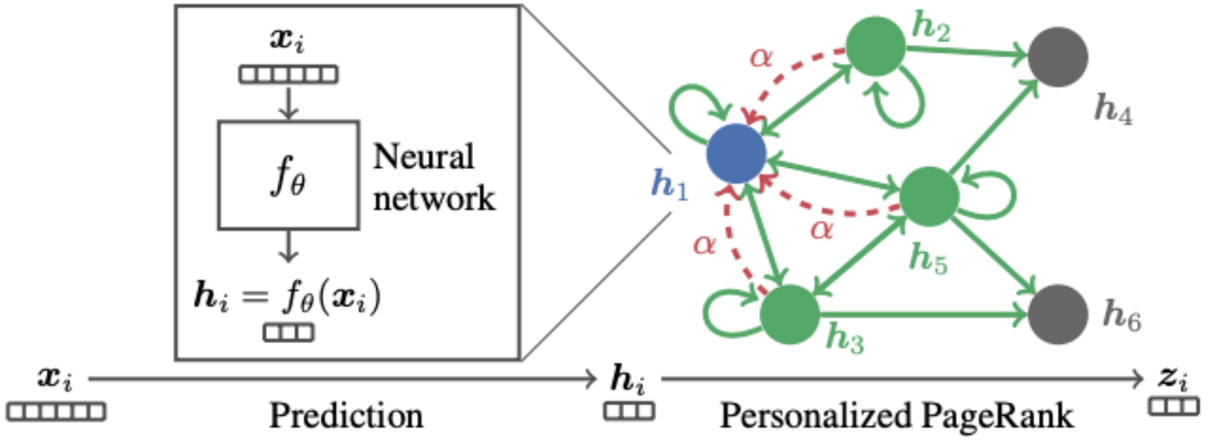
Message Pass Neural Networks [2]

[1] Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." *ICLR*, 2016.
[2] Gilmer, Justin, et al. "Neural message passing for quantum chemistry." *ICML*, 2017.
[3] Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." *NeurIPS,* 2017.

Examples of linearized (shallow) models



SGC [1]



APPNP [2], PPRGo [3]

Complexity of neural networks do not grow as number of propagations increase !

[1] Wu, Felix, et al. "Simplifying graph convolutional networks." *ICML*, 2019.
[2] Klicpera, Johannes, Aleksandar Bojchevski, and Stephan Günnemann. "Predict then Propagate: Graph Neural Networks meet Personalized PageRank." *ICLR,* 2018.
[3] Bojchevski, Aleksandar, et al. “Scaling graph neural networks with approximate pagerank.” *KDD*, 2020.

Standard GNN – regularization on Z

$$\Phi = F(\Theta, A)$$

- Φ is fully differentiable. We sample an IID data of the same size of training data and minimize the distribution shift between Z_{train} and Z_{IID}

$$\mathcal{L} = \frac{1}{M}\sum_{i} l(y_i, z_i) + \lambda \cdot d(Z_{\text{train}}, Z_{\text{IID}}).$$

$$d_{\text{CMD}}(Z_{\text{train}}, Z_{\text{IID}}) = \frac{1}{b-a}\|\mathbf{E}(Z_{\text{train}}) - \mathbf{E}(Z_{\text{IID}})\| + \sum_{k=2}^{\infty} \frac{1}{|b-a|^k}\|c_k(Z_{\text{train}}) - c_k(Z_{\text{IID}})\|,$$

Linearized GNN – instance re-weighting

$$\Phi = F_2(\Theta, F_1\ (A))$$

$$\mathcal{L} = \frac{1}{M}\beta_i l(y_i, \Phi(h_i)),$$

- We use importance sampling to mitigate the shift, calculate the instance weight via kernel mean matching [1],

$$\min_{\beta_i} \| \frac{1}{M}\sum_{i=1}^{M}\beta_i\psi(h_i) - \frac{1}{M'}\sum_{i=1}^{M'}\psi(h'_i)\|^2, \ \textbf{s.t.}\ B_l \leq \beta < B_u$$

[1] Gretton, Arthur, et al. "Covariate shift by kernel mean matching." *Dataset shift in machine learning* 3.4 (2009): 5

Shift-Robust training framework

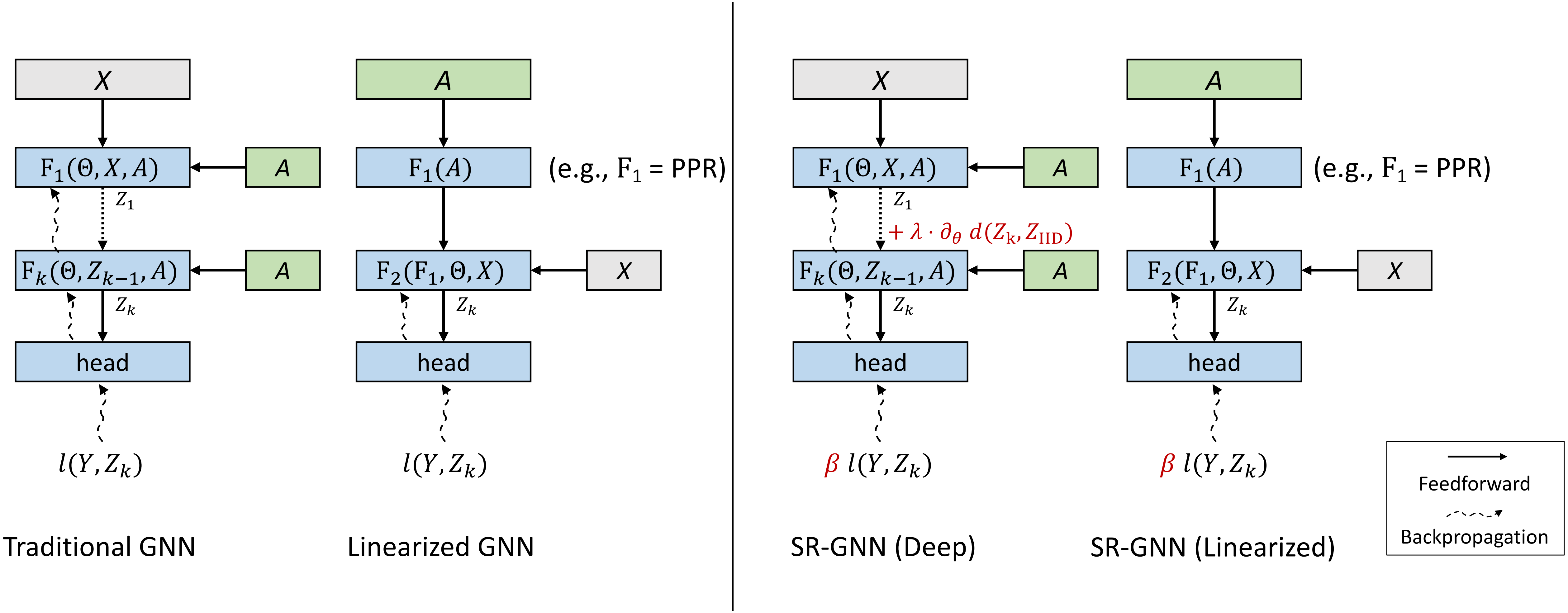
$$\mathcal{L}_{\text{SR-GNN}} = \frac{1}{M}\beta_i l(y_i, \Phi(x_i, A)) + \lambda \cdot d(Z_{\text{train}}, Z_{\text{IID}}).$$

- We choose APPNP [1] (a linearized model) as a concrete example that both techniques can be applied

$$\Phi_{\text{APPNP}} = \underbrace{\left((1-\alpha)^k \tilde{A}^k + \alpha \sum_{i=0}^{k-1} (1-\alpha)^i \tilde{A}^i\right)}_{\text{approximated personalized page rank}} \underbrace{\mathbf{F}(\Theta, X)}_{\text{feature encoder}}.$$

[1] Klicpera, Johannes, Aleksandar Bojchevski, and Stephan Günnemann. "Predict then Propagate: Graph Neural Networks meet Personalized PageRank." *ICLR,* 2018.

Shift-Robust training framework



Overview

- What is localized training data ?
- Quantify the training bias
  - Distribution shift as domain adaption
- Proposed Shift-Robust framework
  - Standard GNN models
  - Linearized GNN models
- Experiments
  - Main Result
  - Parameter sensitivity
- Future work

Biased training set creation

- The localized training data in real-world applications is not easy to control the degree of bias. We propose a scalable biased training data generation process based on fast Personalized Page Rank computation [1].

Algorithm 1: Biased Training Set Creation PPR-S(γ, ϵ, α)

Given a class c, label ratio τ, graph size N;
Initialize the biased training set X = {} ;
while *len(X)*$< N \cdot \tau$ **do**
    Sample node i of class c, compute its top-γ entries in $\pi_i^{\text{ppr}}(\epsilon)$ via [2];
    if $\pi_i^{ppr}(\epsilon)$ *has* γ *non-zero entries* **then**
        X.add($\pi_i^{\text{ppr}}(\epsilon)$) ;
    end
end

[1] Andersen, Reid, Fan Chung, and Kevin Lang. "Local graph partitioning using pagerank vectors." FOCS, 2006.

Biased training data example

(a) IID sample **(b)** Biased sample **(c)** PPR-score on biased sample

Figure 1: A biased sample on Cora dataset for one class, orange indicates the training data, red indicates the initial seed used in our PPR-S sampler. The PPR-score is presented in figure (c).

Experimental result on small benchmarks

Table 1: Semi-supervised classification on three different citation networks using biased training samples. Our proposed framework (SR-GNN) outperforms **all** baselines on biased training input.

Method	Cora			Citeseer			PubMed		
	Micro-F1↑	Macro-F1↑	ΔF1 ↓	Micro-F1↑	Macro-F1↑	ΔF1 ↓	Micro-F1↑	Macro-F1↑	ΔF1 ↓
GCN (IID)	80.8 ± 1.6	80.1 ± 1.3	0	70.3 ± 1.9	66.8 ± 1.3	0	79.8 ± 1.4	78.8 ± 1.4	0
Feat.+MLP	49.7 ± 2.5	48.3 ± 2.2	31.1	55.1 ± 1.3	52.7 ± 1.3	25.2	51.3 ± 2.8	41.8 ± 6.2	28.5
Emb.+MLP	57.6 ± 3.0	56.2 ± 3.0	23.2	38.5 ± 1.2	38.6 ± 1.1	31.8	60.4 ± 2.1	56.6 ± 2.0	19.4
DGI	71.7 ± 4.2	69.2 ± 3.7	9.1	62.6 ± 1.6	60.0 ± 1.6	7.6	58.0 ± 5.3	52.4 ± 8.3	21.8
GCN	67.6 ± 3.5	66.4 ± 3.0	13.2	62.7 ± 1.8	60.4 ± 1.6	7.6	60.6 ± 3.8	56.0 ± 6.0	19.2
GAT	58.4 ± 5.7	58.5 ± 5.0	22.4	58.0 ± 3.5	55.0 ± 2.7	12.3	55.2 ± 3.7	46.0 ± 6.4	14.6
SGC	70.2 ± 3.0	68.0 ± 3.8	10.6	65.4 ± 0.8	62.5 ± 0.8	4.9	61.8 ± 4.5	57.4 ± 7.2	18.0
APPNP	71.3 ± 4.1	69.2 ± 3.4	9.5	63.4 ± 1.8	61.2 ± 1.6	6.9	63.4 ± 4.2	58.7 ± 7.0	16.4
w.o. KMM	72.1 ± 4.4	69.8 ± 3.7	8.7	63.9 ± 0.7	61.8 ± 0.6	6.4	69.4± 3.4	67.6 ± 4.0	10.4
w.o. CMD	72.0 ± 3.2	69.5 ± 3.7	8.8	66.1 ± 0.9	63.4 ± 0.9	4.2	66.4 ± 4.0	64.0 ± 5.5	13.4
SR-GNN (Ours)	**73.5 ± 3.3**	**71.4± 3.5**	**7.3**	**67.1 ± 0.9**	**64.0 ± 0.9**	**3.2**	**71.3 ± 2.2**	**70.2 ± 2.4**	**8.5**

SR-GNN outperforms other GNN baselines by accurately eliminating at least (~40%) of the negative effect.

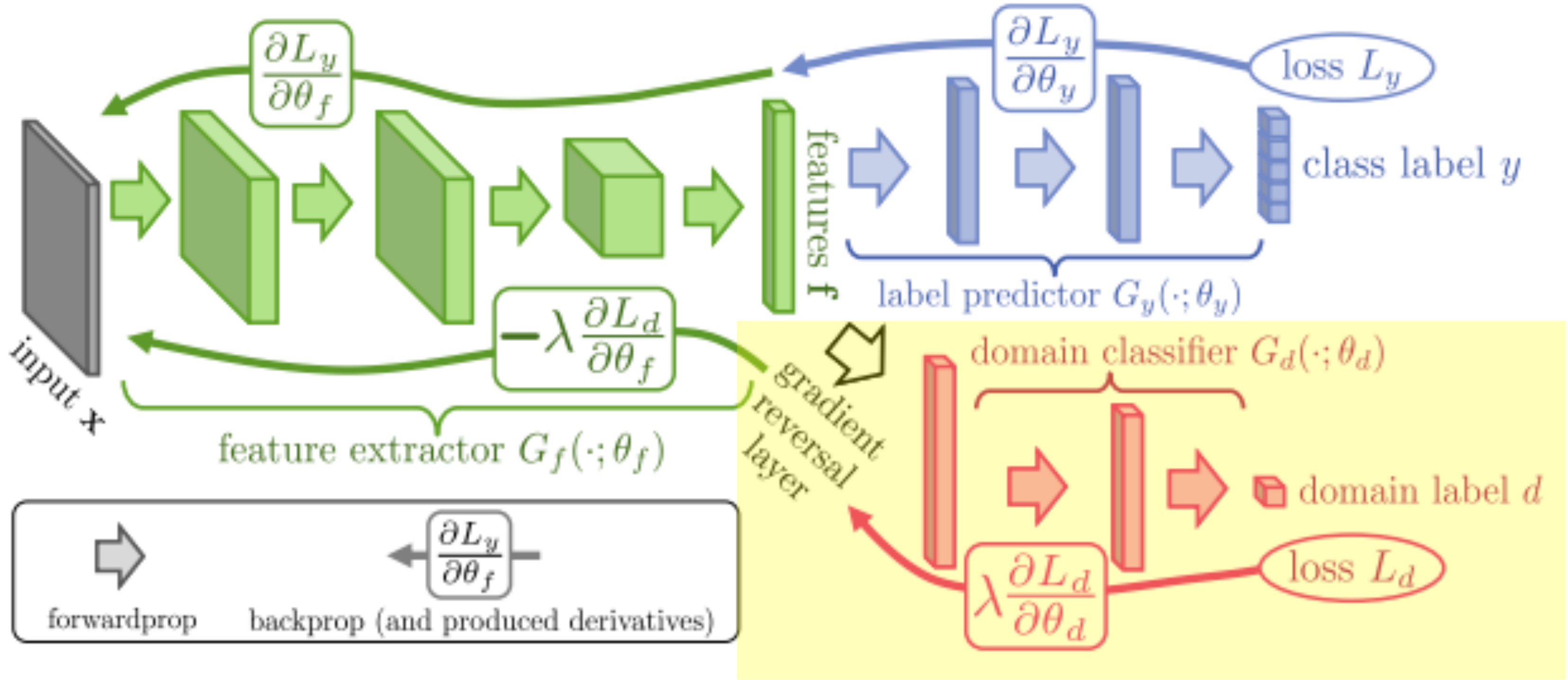
Experimental result on large benchmark

Table 2: Semi-supervised classification on ogb-arxiv varying label ratio.

label(%) Method	1 % Accuracy	$\Delta\downarrow$	5 % Accuracy	$\Delta\downarrow$
GCN (IID)	66.0± 0.6	0	69.1± 0.6	0
Feat.+MLP	45.5± 0.6	21.5	43.7± 0.3	25.4
Emb.+MLP	51.1± 1.3	14.9	56.9± 0.8	13.2
DGI	44.8± 3.0	21.2	49.7± 3.3	19.4
GCN	59.3± 1.2	6.7	65.3 ± 0.6	3.8
GAT	58.6± 1.0	7.4	63.4 ± 1.0	5.7
SGC	59.0± 0.7	7.0	64.2 ± 1.3	4.9
APPNP	59.8± 1.1	6.2	65.1 ± 2.6	4.0
w.o. KMM	60.6± 0.2	5.4	65.1±1.8	4.0
w.o. CMD	61.0± 0.3	5.0	65.8±2.0	3.3
SR-GNN (Ours)	**61.6±0.6**	**4.4**	**66.5±0.6**	**2.6**

SR-GNN improve 2% absolute accuracy and eliminate ~30% of the negative effect by biased data.

Comparison with domain adversarial network



- DANN [1] is a method that uses an adversarial domain classifier to encourage similar feature distributions between different domains.

[1] Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." *JMLR,* 2016.

Comparison with domain adversarial network

Table 6: Comparison of Domain-Adversarial Neural Network (DANN) and CMD regularizer used in SR-GNN with biased training data.

Method	Cora Micro-F1↑	Cora Macro-F1↑	Citeseer Micro-F1↑	Citeseer Macro-F1↑	PubMed Micro-F1↑	PubMed Macro-F1↑
GCN	68.3	67.2	62.4	60.2	59.2	53.8
DANN	69.8	68.5	63.8	61.0	64.8	61.8
CMD (Ours)	**71.0**	**69.4**	**65.0**	**62.3**	**67.5**	**66.2**
APPNP	71.3	69.2	63.9	61.6	64.8	60.4
DANN	71.6	69.5	64.3	61.8	67.8	65.4
CMD (Ours)	**72.4**	**70.1**	**65.0**	**62.4**	**70.4**	**68.7**

Under semi-supervised setting, the performance of DANN is more sensitive to the domain loss. CMD regularizer performs better with more robust weight selection. Not that CMD regularizer is one component of the proposed SR-GNN.

Apply Shift-Robust on other GNN instances

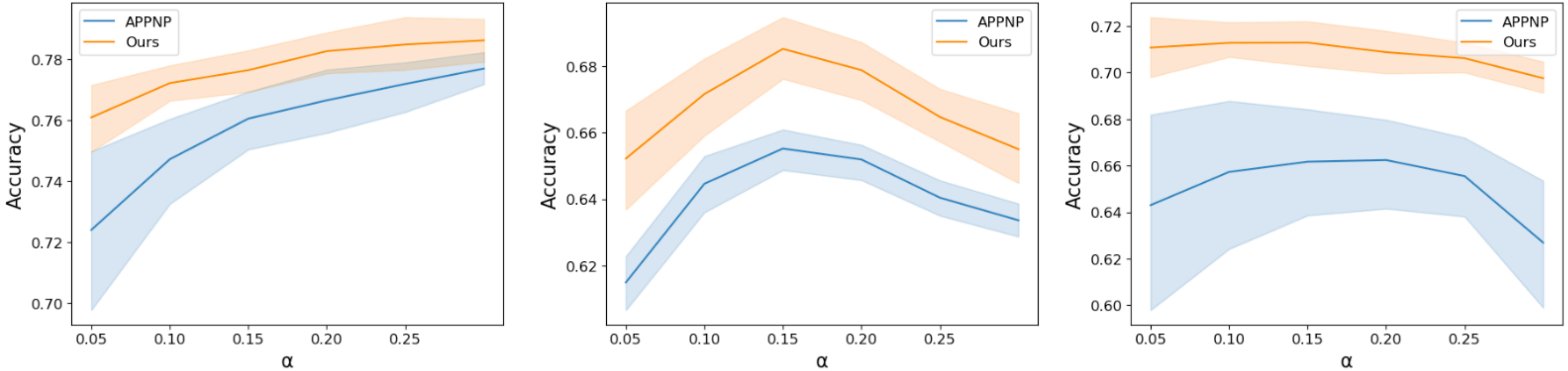
Table 3: Comparison of baseline and our SR(Shift-Robust) version ($\Delta(\%)$ -relative loss with biased sample) .

Method	Cora			Citeseer			PubMed		
	Micro-F1↑	Macro-F1↑	$\Delta(\%)$	Micro-F1↑	Macro-F1↑	$\Delta(\%)$	Micro-F1↑	Macro-F1↑	$\Delta(\%)$
GCN (IID)	80.8	80.1	0%	70.3	66.8	0%	79.8	78.8	0%
GCN	67.6	66.4	-12%	62.7	60.4	-8%	60.6	56.0	-19%
SR-GCN	**69.6**	**68.2**	-10%	**64.7**	**62.0**	-6%	**67.0**	**65.2**	-13%
DGI (IID)	80.6	79.3	0%	70.8	66.7	0%	77.6	77.0	0%
DGI	71.7	69.2	-9%	62.6	60.0	-8%	58.0	52.4	-20%
SR-DGI	**74.3**	**72.6**	-6%	**65.8**	**62.6**	-6%	**62.0**	**57.8**	-16%

Overview

- What is localized training data ?
- Quantify the training bias
  - Distribution shift as domain adaption
- Proposed shift-robust framework
  - Standard GNN models
  - Linearized GNN models
- Experiments
  - Main result
  - Parameter sensitivity
- Future work

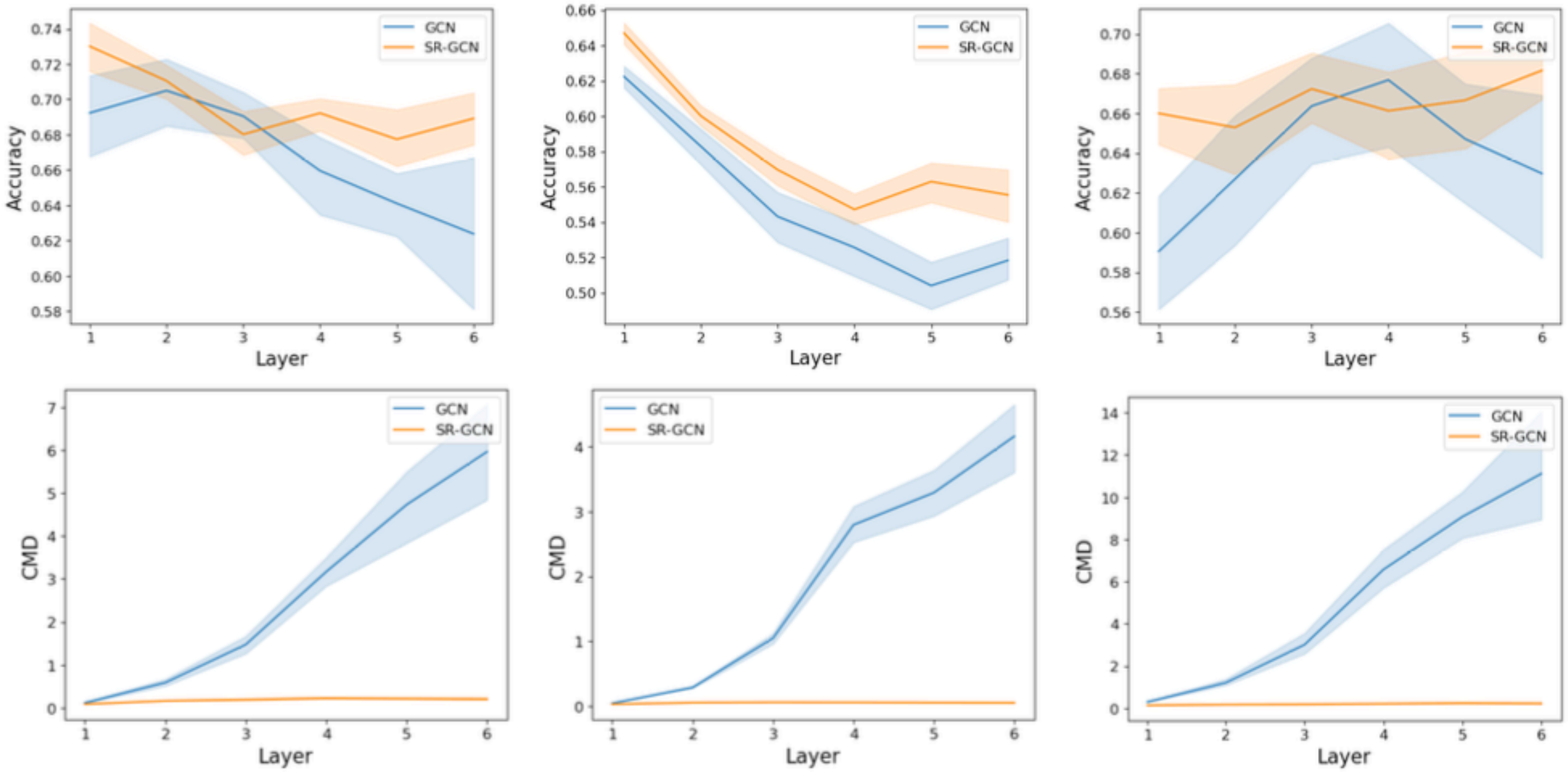
Varying **α** in biased training set creation



(a) Cora　**(b)** Citeseer　**(c)** Pubmed

α is the termination probability in PPR. Larger α means more localized PPR-neighbors.

SR-GNN on deeper models

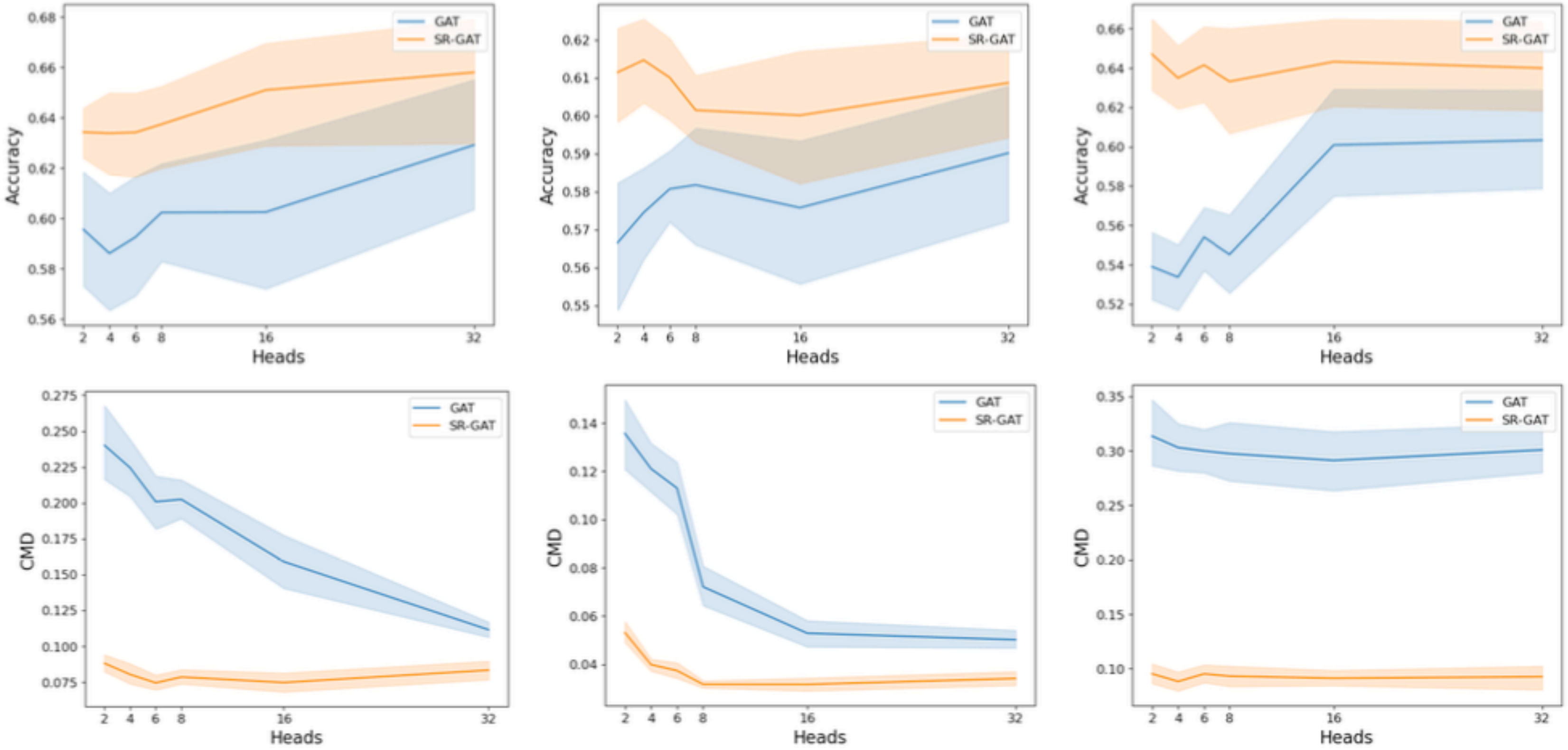


(a) Cora **(b)** Citeseer **(c)** Pubmed

Figure 2: Comparison of GCN vs. SR-GCN model performance with the the same parameters. Our shift-robust algorithm boosts the performance (top) consistently by reducing the distribution shifts (bottom).

Larger shift presented in deeper models! SR-GNN consistently works.

SR-GNN on wider models



(a) Cora **(b)** Citeseer **(c)** Pubmed

Figure 3: Comparison of GAT vs. SR-GAT model performance under increasing attention heads. Our shift-robust algorithm boosts the performance (upper) consistently by reducing the distribution shifts (lower).

Smaller distributional-shift in wider models.

Future work

- Develop Shift-Robust GNNs on specific domains
  - Maximize the performance when dealing with specific shift in spam and abuse detection.
- Theoretical guarantee towards shift-robust requirement
  - Fairness of training data
  - Generalization error in terms of distributional shift

Thanks and Q&A

- More results are available: https://arxiv.org/pdf/2108.01099.pdf
- Questions and discussions: qiz3@Illinois.edu