GC-Flow: A Graph-Based Flow Network for Effective Clustering

Tianchun Wang [1] Farzaneh Mirzazadeh [2] Xiang Zhang [1] Jie Chen [2]

[1]The Pennsylvania State University [2]MIT-IBM Watson AI Lab, IBM Research

GCN's Limitation

Graph convolutional network (GCN) [1] is a seminal deep learning model for semi-supervised learning on graphs. It capitalizes on the graph structure underlying data to achieve effective classification.

However, GCN and other GNNs may not produce node representations with useful information for goals different from classification; e.g., the representations do not cluster well.

GC-Flow

Question: Can we build a graph representation model that is effective for not only classification but also clustering?

Answer: Rather than construct a discriminative model $p(y|\mathbf{x})$ as all GNNs do, we build a generative model $p(\mathbf{x}|y)p(y)$ whose class conditional likelihood is defined by explicitly modeling the representation space.

FlowGMM [2]: FlowGMM is a normalizing flow that maps the distribution of input features to a Gaussian mixture, resulting in well-structured clusters.

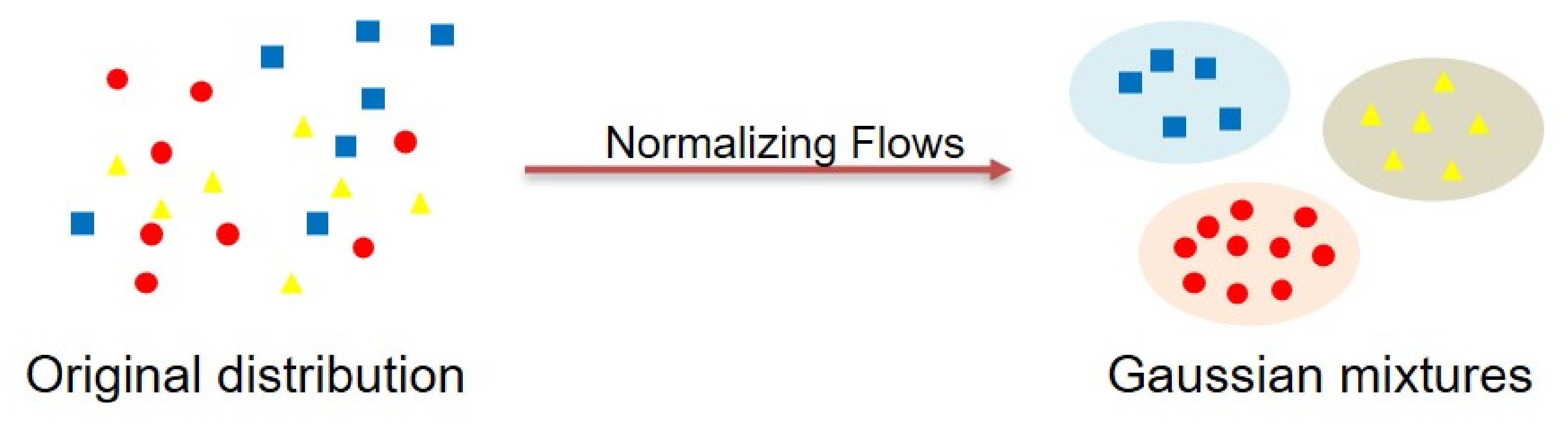
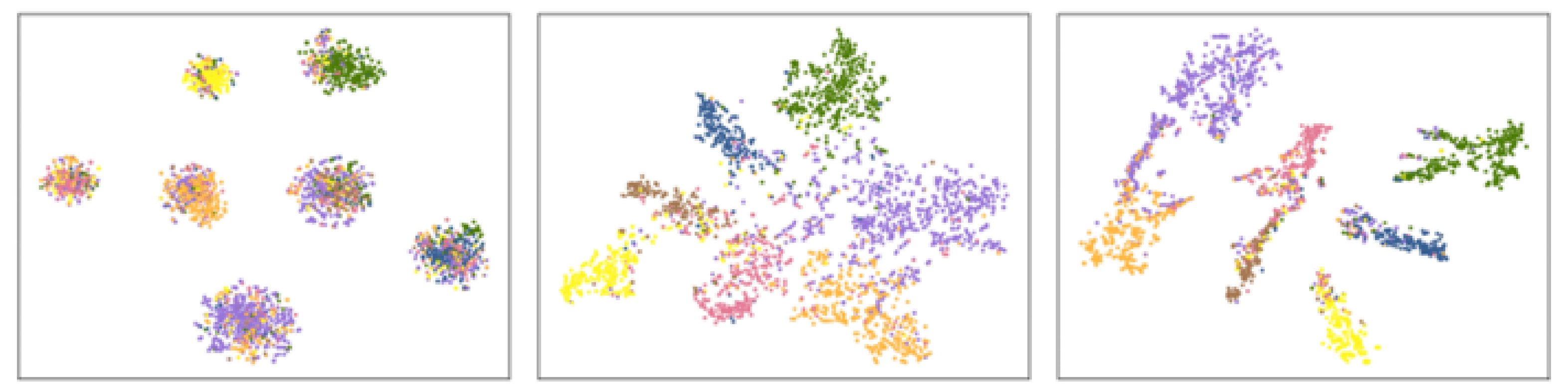


Figure: The framework of FlowGMM.

GC-Flow: We propose **graph convolutional normalizing flow** (GC-Flow), which extends FlowGMM that acts on data points separately to one that acts on all graph nodes collectively.



(a) FlowGMM, Silhouette = 0.73, F1 = 0.52 (b) GCN, Silhouette = 0.34, F1 = 0.81 (c) GC-Flow, Silhouette = 0.73, F1 = 0.82

Figure: Representation space of Cora. Left to right: FlowGMM; GCN; GC-Flow.

Mathematical Definition

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of a given graph with n nodes and let $\widehat{\mathbf{A}}$ be a certain normalization of $\mathbf{A}$ (e.g., the normalization defined by GCN). Let $\mathbf{X}$ be the $n \times D$ input feature matrix. Starting with $\mathbf{X}^{(0)} \equiv \mathbf{X}$, we define GC-Flow $\mathbf{F}(\mathbf{X}) : \mathbb{R}^{n \times D} \to \mathbb{R}^{n \times D}$ as a composition of T constituent flows $\mathbf{F} = \mathbf{F}_T \circ \mathbf{F}_{T-1} \circ \cdots \circ \mathbf{F}_1$, where each constituent flow $\mathbf{F}_i$ has parameter $\mathbf{W}^{(i-1)}$ and computes

$$\mathbf{X}^{(i)} = \mathbf{F}_i(\underbrace{\widehat{\mathbf{A}}\mathbf{X}^{(i-1)}}_{\widetilde{\mathbf{X}}^{(i)}}; \mathbf{W}^{(i-1)}), \quad i = 1, \ldots, T. \tag{1}$$

The final node representation is $\mathbf{Z} \equiv \mathbf{X}^{(T)} \in \mathbb{R}^{n \times D}$.

Parameterization: In addition to $\mathbf{W}^{(i-1)}$, $\widehat{\mathbf{A}}$ can also be parameterized.

Constituent flows: Any normalizing flow will do.

Properties: 1) GC-Flow is a normalizing flow; 2) GC-Flow is a GNN.

Probability Model and Training Objective

A major distinction between GC-Flow and a usual GNN lies in the training objective (which requires a probability model). We use a maximum-likelihood type of objective, which is equivalent to maximizing the likelihood that $\mathbf{Z}$ forms a Gaussian mixture:

$$\max\ \mathcal{L} := \frac{1-\lambda}{|\mathcal{D}_l|} \sum_{(\mathbf{x}, y=k) \in \mathcal{D}_l} \log p(\mathbf{x}, y=k) + \frac{\lambda}{|\mathcal{D}_u|} \sum_{\mathbf{x} \in \mathcal{D}_u} \log p(\mathbf{x}), \tag{2}$$

where $\mathcal{D}_l$ and $\mathcal{D}_u$ denote the set of labeled and unlabeled nodes, respectively.

We use $p(\mathbf{X})$ and $\pi(\mathbf{Z})$ to denote the joint distribution of the node feature vectors and that of the representations, respectively. The change-of-variable formula gives

$$p(\mathbf{X}) = \pi(\mathbf{Z})|\det \nabla \mathbf{F}(\mathbf{X})|. \tag{3}$$

We assume the node features to be iid $p(\mathbf{X}) = p(\mathbf{x}_1)p(\mathbf{x}_2)\cdots p(\mathbf{x}_n)$ and also model the representations to be iid $p(\mathbf{Z}) = p(\mathbf{z}_1)p(\mathbf{z}_2)\cdots p(\mathbf{z}_n)$.

Determinant Lemma

The Jacobian determinant of the GC-Flow $\mathbf{F}$ is

$$|\det \nabla \mathbf{F}(\mathbf{X})| = |\det \widehat{\mathbf{A}}|^{TD} \prod_{j=1}^{T} \prod_{i=1}^{n} |\det \nabla \mathbf{f}_j(\widetilde{\mathbf{x}}_i^{(j)})|. \tag{4}$$

Based on the lemma, we model the class prior $p(y)$ and the class conditional likelihood $p(\mathbf{x}|y)$ as

$$p(\mathbf{x}_i|y_i = k) := \mathcal{N}(\mathbf{z}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)|\det \widehat{\mathbf{A}}|^{TD/n} \times \prod_{j=1}^{T} |\det \nabla \mathbf{f}_j(\widetilde{\mathbf{x}}_i^{(j)})|$$
$$p(y_i = k) := \phi_k. \tag{5}$$

Then, the marginal likelihood is

$$p(\mathbf{x}_i) = \pi(\mathbf{z}_i)|\det \widehat{\mathbf{A}}|^{TD/n} \prod_{j=1}^{T} |\det \nabla \mathbf{f}_j(\widetilde{\mathbf{x}}_i^{(j)})|, \tag{6}$$

which is consistent with (3). We use (5) and (6) to compute the labeled part and the unlabeled part of the loss (2), respectively.

Clustering & Classification Performance

Table: Comparison of GMM-based generative models, GNN-based discriminative models, and GC-Flow for semi-supervised clustering and classification.

	Cora		Citeseer		Pubmed	
	Silhouette	Micro-F1	Silhouette	Micro-F1	Silhouette	Micro-F1
FlowGMM	**0.739** $\pm$ **0.015**	0.504 $\pm$ 0.021	**0.609** $\pm$ **0.034**	0.512 $\pm$ 0.044	**0.653** $\pm$ **0.031**	0.734 $\pm$ 0.014
GMM on $\mathbf{X}$	0.162 $\pm$ 0.000	0.163 $\pm$ 0.000	0.071 $\pm$ 0.000	0.085 $\pm$ 0.000	0.062 $\pm$ 0.000	0.581 $\pm$ 0.000
GMM on $\widehat{\mathbf{A}}\mathbf{X}$	0.144 $\pm$ 0.000	0.173 $\pm$ 0.000	0.089 $\pm$ 0.000	0.182 $\pm$ 0.000	0.183 $\pm$ 0.000	0.411 $\pm$ 0.000
GCN	0.340 $\pm$ 0.003	0.813 $\pm$ 0.007	0.314 $\pm$ 0.016	0.700 $\pm$ 0.025	0.453 $\pm$ 0.006	**0.791** $\pm$ **0.004**
GraphSAGE	0.346 $\pm$ 0.004	0.801 $\pm$ 0.005	0.278 $\pm$ 0.007	0.697 $\pm$ 0.007	0.440 $\pm$ 0.018	0.769 $\pm$ 0.011
GAT	0.383 $\pm$ 0.003	**0.825** $\pm$ **0.005**	0.304 $\pm$ 0.003	**0.702** $\pm$ **0.007**	0.435 $\pm$ 0.010	0.774 $\pm$ 0.005
GC-Flow	**0.734** $\pm$ **0.006**	**0.815** $\pm$ **0.011**	**0.538** $\pm$ **0.022**	**0.714** $\pm$ **0.011**	**0.669** $\pm$ **0.021**	**0.791** $\pm$ **0.009**

	Computers		Photo		Wiki-CS	
	Silhouette	Micro-F1	Silhouette	Micro-F1	Silhouette	Micro-F1
FlowGMM	**0.540** $\pm$ **0.024**	0.614 $\pm$ 0.026	**0.704** $\pm$ **0.027**	0.599 $\pm$ 0.089	**0.677** $\pm$ **0.011**	0.671 $\pm$ 0.011
GMM on $\mathbf{X}$	-0.018 $\pm$ 0.00	0.102 $\pm$ 0.000	-0.024 $\pm$ 0.00	0.120 $\pm$ 0.000	0.088 $\pm$ 0.000	0.124 $\pm$ 0.000
GMM on $\widehat{\mathbf{A}}\mathbf{X}$	-0.021 $\pm$ 0.00	0.062 $\pm$ 0.000	-0.041 $\pm$ 0.00	0.098 $\pm$ 0.000	0.026 $\pm$ 0.000	0.188 $\pm$ 0.000
GCN	0.357 $\pm$ 0.026	0.812 $\pm$ 0.016	0.388 $\pm$ 0.003	0.891 $\pm$ 0.012	0.264 $\pm$ 0.005	**0.775** $\pm$ **0.005**
GraphSAGE	0.434 $\pm$ 0.030	0.761 $\pm$ 0.024	0.386 $\pm$ 0.007	0.839 $\pm$ 0.020	0.233 $\pm$ 0.009	0.771 $\pm$ 0.003
GAT	0.431 $\pm$ 0.015	**0.814** $\pm$ **0.023**	0.425 $\pm$ 0.020	**0.900** $\pm$ **0.009**	0.278 $\pm$ 0.008	0.773 $\pm$ 0.003
GC-Flow	**0.487** $\pm$ **0.012**	**0.847** $\pm$ **0.007**	**0.655** $\pm$ **0.013**	**0.917** $\pm$ **0.004**	**0.717** $\pm$ **0.010**	**0.775** $\pm$ **0.002**

More Results

Comparison with more clustering methods:

Table: Clustering performance of various GNN methods (Cora).

	NMI	ARI	Silhouette
DGI	0.592 $\pm$ 0.001	0.570 $\pm$ 0.002	0.330 $\pm$ 0.000
GRACE	0.475 $\pm$ 0.028	0.394 $\pm$ 0.047	0.153 $\pm$ 0.011
GCA	0.418 $\pm$ 0.053	0.259 $\pm$ 0.058	0.301 $\pm$ 0.005
GraphCL	0.577 $\pm$ 0.002	0.482 $\pm$ 0.003	0.297 $\pm$ 0.002
MVGRL	**0.612** $\pm$ **0.026**	**0.576** $\pm$ **0.062**	0.369 $\pm$ 0.013
R-GMM-VGAE	0.559 $\pm$ 0.006	0.557 $\pm$ 0.009	**0.430** $\pm$ **0.013**
GC-Flow	**0.621** $\pm$ **0.013**	**0.631** $\pm$ **0.008**	**0.734** $\pm$ **0.006**

Training behavior:

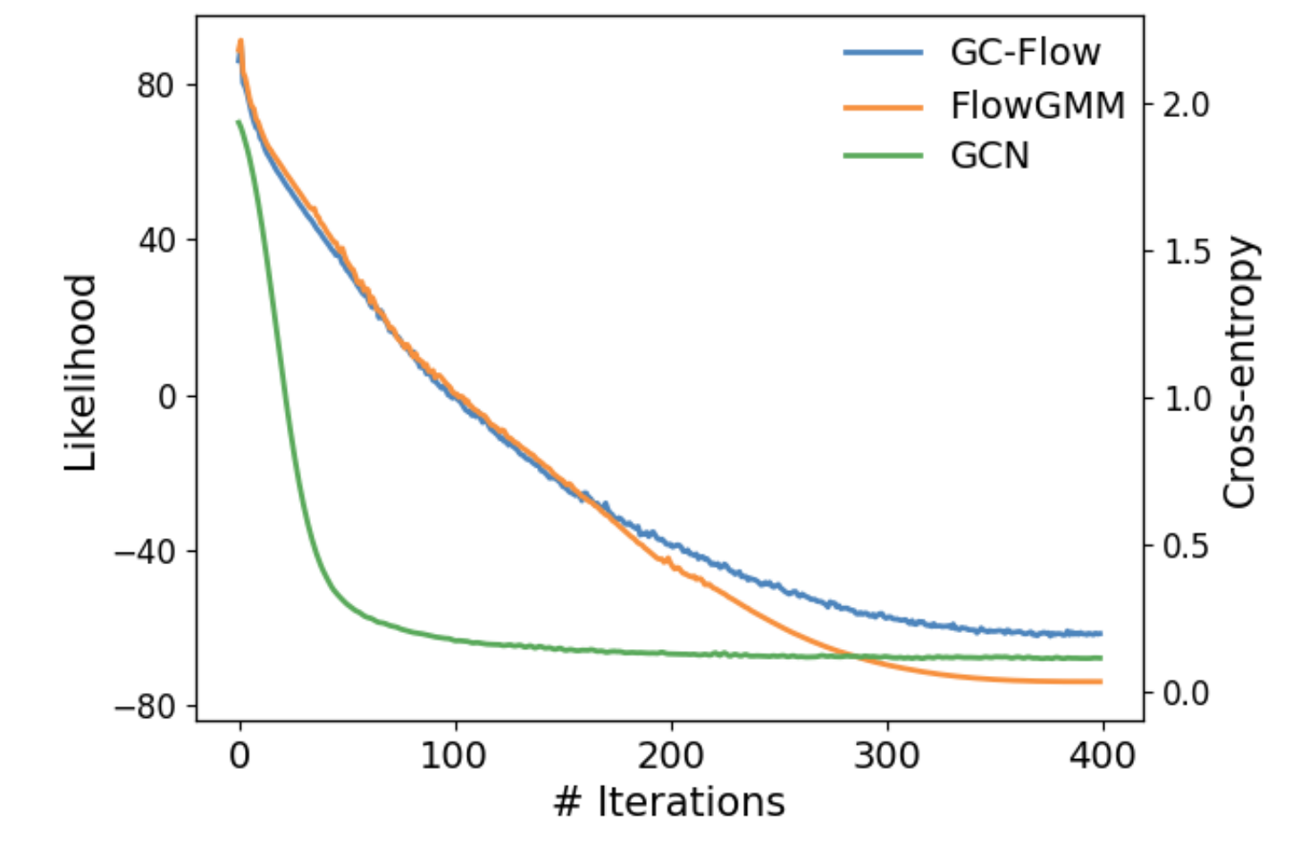


Figure: Convergence of the training loss (Cora).

Analysis on labeling rate:

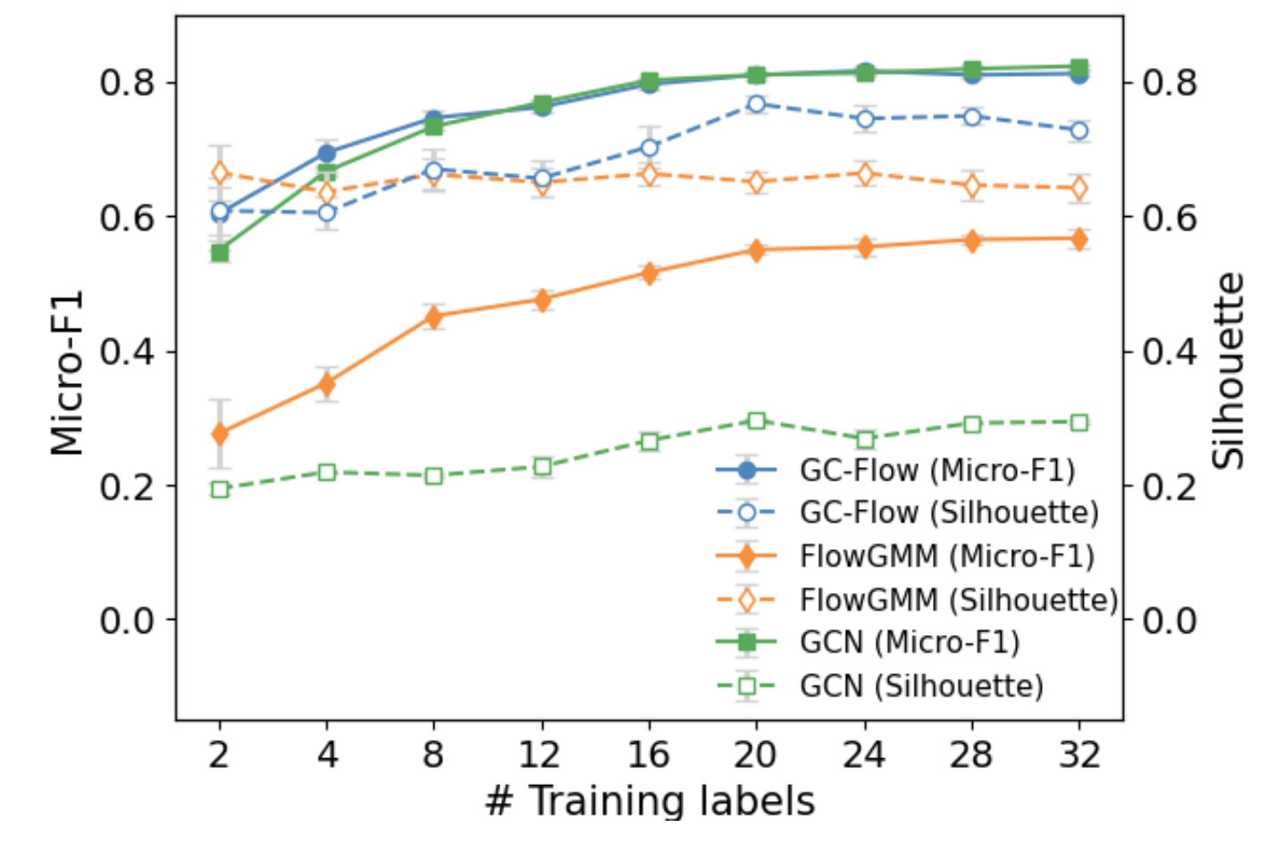
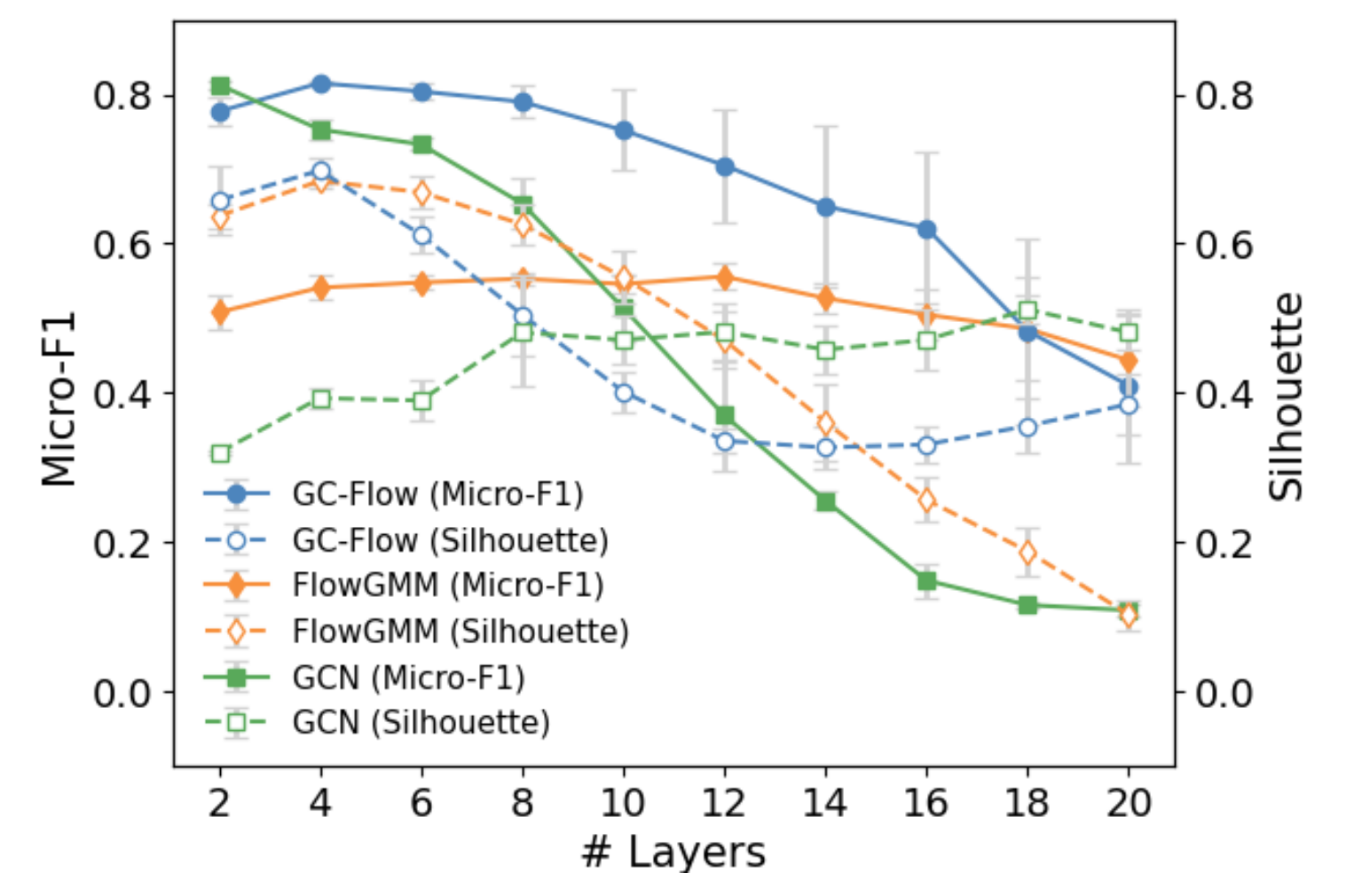
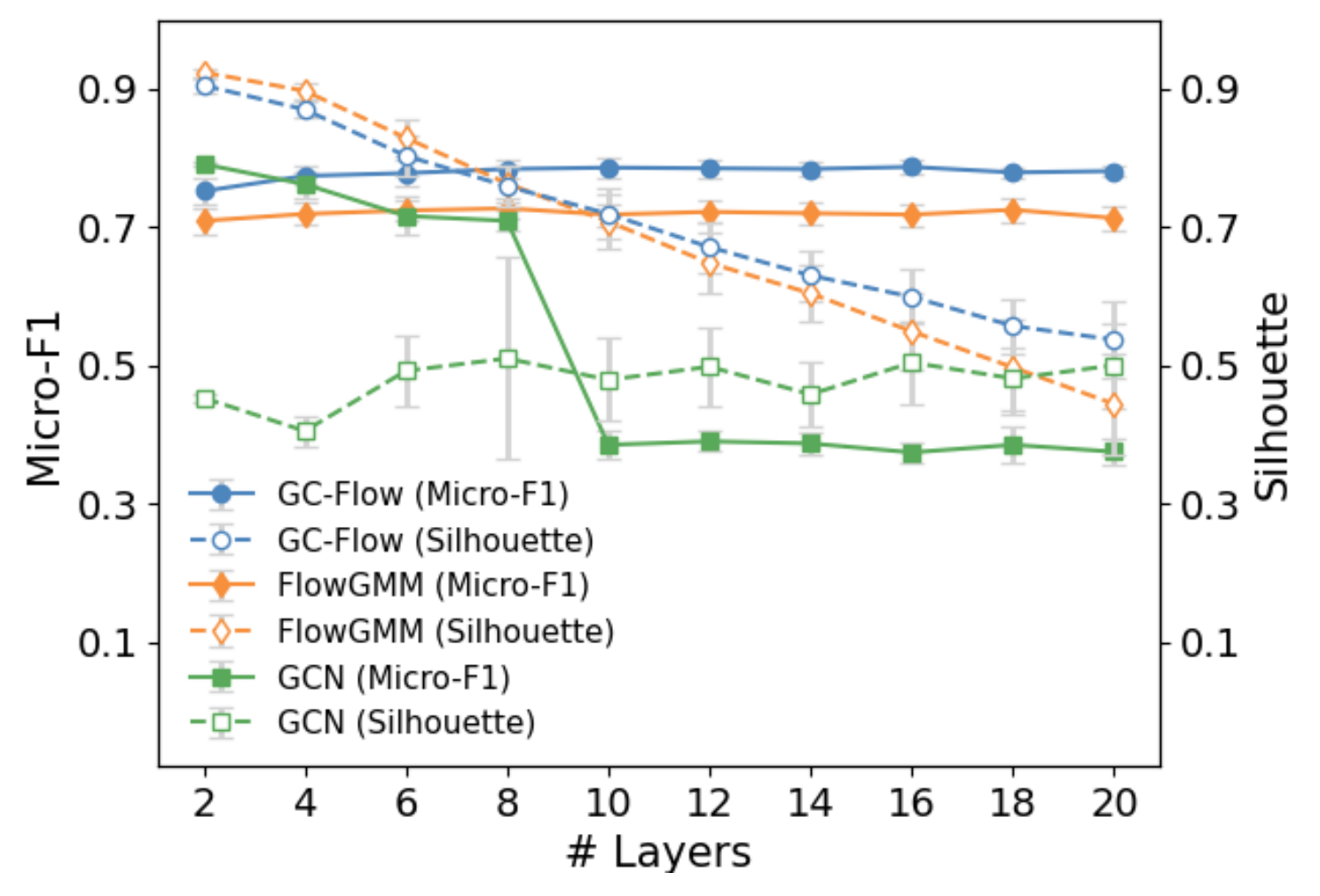


Figure: Performance variation w.r.t. the labeling rate (Cora).

Analysis on depth:



(a) Cora



(b) Pubmed

Figure: Performance variation w.r.t. the network depth (i..e, number of constituent flows).

Improving performance by parameterizing $\widehat{\mathbf{A}}$:

Table: Effect of modeling $\widehat{\mathbf{A}}$. Boldfaced numbers indicate improvement over GC-Flow.

	Pubmed		Computers		Photo	
	Silhouette	Micro-F1	Silhouette	Micro-F1	Silhouette	Micro-F1
GC-Flow	0.669 $\pm$ 0.021	0.791 $\pm$ 0.009	0.487 $\pm$ 0.012	0.847 $\pm$ 0.007	0.655 $\pm$ 0.013	0.917 $\pm$ 0.004
GC-Flow-p	**0.804** $\pm$ **0.010**	0.790 $\pm$ 0.007	**0.706** $\pm$ **0.019**	0.841 $\pm$ 0.006	**0.874** $\pm$ **0.011**	0.914 $\pm$ 0.008
GC-Flow-l	**0.856** $\pm$ **0.029**	0.783 $\pm$ 0.009	**0.582** $\pm$ **0.020**	**0.851** $\pm$ **0.009**	**0.842** $\pm$ **0.006**	0.911 $\pm$ 0.005

References

[1] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Preprint arXiv:1609.02907*, 2016.

[2] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *ICML*, pages 4615–4630. PMLR, 2020.

Code is available at: https://github.com/xztcwang/GCFlow