Adaptive Diffusions for Scalable and Robust Learning over Graphs

D. K. Berberidis ***Georgios B. Giannakis*** *A. N. Nikolakopoulos*

Dept. of ECE and Digital Tech. Center, University of Minnesota

Acknowledgments: NSF 1500713,1711471, NIH 1R01GM104975-01

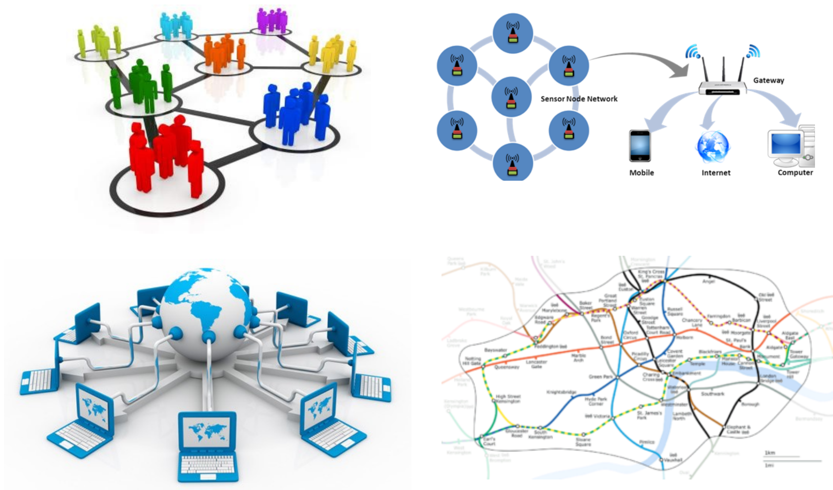
UNIVERSITY OF MINNESOTA
Driven to Discover℠

Shanghai, P. R. China
July 2, 2018

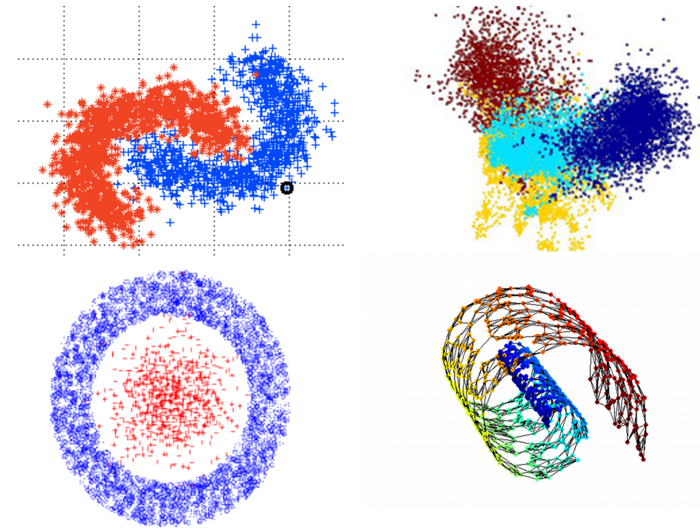
Motivation

Graph representations

Real networks



Data similarities

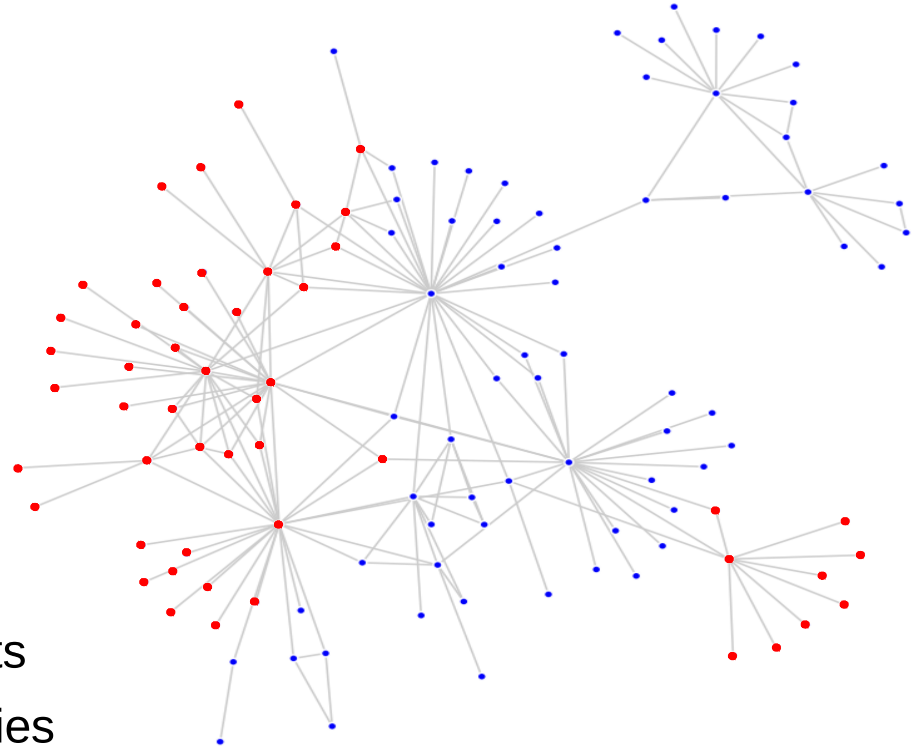


Objective: Learn values or labels of graph nodes, as e.g., in citation networks

Challenges: Graphs can be **huge** and are **scarcely labeled**

- Due to privacy, cost of battery, (un) reliable human annotators …

Problem statement



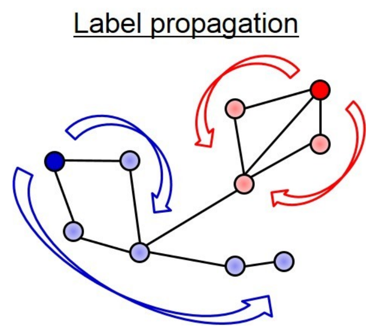
- Graph $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$
  - Weighted adjacency matrix $\mathbf{W}$
  - Label $y_i \in \mathcal{Y}$ per node v_i

- Topology given or identifiable
  - Given in e.g. WSNs and social nets
  - Identifiable via e.g., nodal similarities

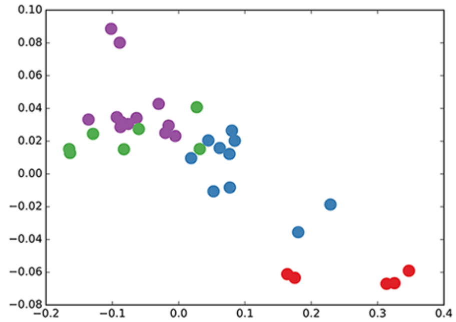
Goal: Given labels on $\mathcal{L} \subseteq \mathcal{V}$ learn unlabeled nodes $\mathcal{U} := \mathcal{V}/\mathcal{L}$

Work in context

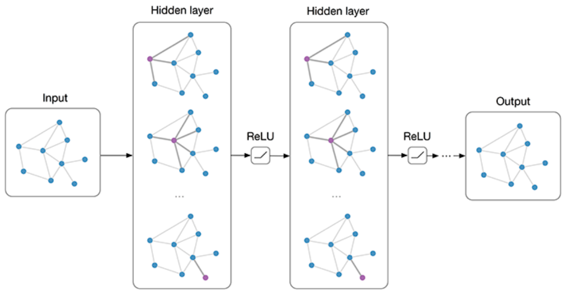
- Non-parametric semi-supervised learning (SSL) on graphs
  - Graph partitioning [Joachims et al ‘03]
  - Manifold regularization [Belkin et al ‘06]
  - Label propagation [Zhu et al’03, Bengio et al‘06]
  - Bootstrapped label propagation [Cohen‘17]
  - Competitive infection models [Rosenfeld‘17]



- Node embedding + classification of vectors
  - Node2vec [Grover et al ’16]
  - Planetoid [Yang et al ‘16]
  - Deepwalk [Perozzi et al ‘14]



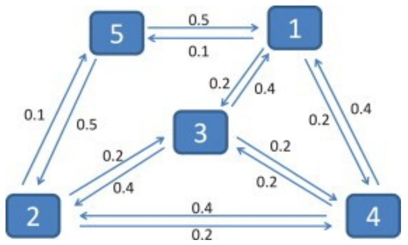
- Graph convolutional networks (GCNs)
  - [Atwood et al ‘16], [Kipf et al ‘16]



Random walks on graphs

- Position of random walker at step *k* : $X_k \in \mathcal{V}$
  - Transition probabilities

$$\begin{aligned}\Pr\{X_k = i | X_{k-1} = j\} &= W_{ij}/d_j \\ &:= [\mathbf{H}]_{ij} = [\mathbf{W}\mathbf{D}^{-1}]_{ij}\end{aligned}$$



- Steady-state probs.

$$\pi_i := \lim_{k\to\infty} \sum_{j\in\mathcal{V}} \Pr\{X_k = i | X_0 = j\} \Pr\{X_0 = j\} = \frac{d_i}{2|\mathcal{E}|}$$

  - Presumes undirected, connected, and non-bipartite graphs
  - **Not** informative for SSL

- Step-*k* landing probabilities

$$p_i^{(k)} := \sum_{j\in\mathcal{V}} \Pr\{X_k = i | X_0 = j\} \Pr\{X_0 = j\}$$

$$\mathbf{p}^{(k)} = \mathbf{H}^k \mathbf{p}^{(0)} := [p_1^{(k)} \dots p_N^{(k)}]^T$$

  - Measure influence of $\mathbf{p}^{(0)}$ on every node in $\mathcal{V}$ - informative for SSL!

Landing probabilities for SSL

- Random walk per class with $\mathbf{p}_c^{(k)} = \mathbf{H}^k \mathbf{v}_c$

$$\mathbf{P}_c^{(K)} := \left[\mathbf{p}_c^{(1)} \quad \cdots \quad \mathbf{p}_c^{(K)}\right]$$

  - Initial ("root") probability distribution

$$[\mathbf{v}_c]_i = \begin{cases} 1/|\mathcal{L}_c|, & i \in \mathcal{L}_c \\ 0, & \text{else} \end{cases}$$

  - Per step landing probabilities found by multiplying with sparse ***H***

$$\mathcal{L}_c := \{i \in \mathcal{L} : y_i = c\}$$

- Family of per-class *diffusions*

$$\mathbf{f}_c(\boldsymbol{\theta}) := \sum_{k=1}^{K} \theta_k \mathbf{p}_c^{(k)} = \mathbf{P}_c^{(K)} \boldsymbol{\theta}, \quad \boldsymbol{\theta} \in \mathcal{S}^K$$

  - Valid pmf with ***K*-dim** probability simplex

$$\mathcal{S}^K := \{\boldsymbol{\theta} \in \mathbb{R}^K : \boldsymbol{\theta} \geq \mathbf{0}, \ \mathbf{1}^\top \boldsymbol{\theta} = 1\}$$

- Max-likelihood per-node classifier

$$\hat{y}_i(\boldsymbol{\theta}) := \arg\max_{c \in \mathcal{Y}} [\mathbf{f}_c(\boldsymbol{\theta})]_i$$

Unifying diffusion-based SSL

Special case 1: Personalized page rank (PPR) diffusion [Lin'10]

$$\mathbf{f}_c(\boldsymbol{\theta}_{\mathrm{PPR}}) = (1-\alpha)\sum_{k=1}^{K} \alpha^k \mathbf{p}_c^{(k)} \qquad \boldsymbol{\theta}_{\mathrm{PPR}} := (1-\alpha)\left[\alpha \cdots \alpha^K\right]^{\mathsf{T}},\ \alpha \in (0,1)$$

- Pmf of random walk with restart probability 1-*α* ; in steady-state

$$\lim_{K\to\infty} \mathbf{f}_c(\boldsymbol{\theta}_{\mathrm{PPR}}) = (1-\alpha)(\mathbf{I}-\alpha\mathbf{H})^{-1}\mathbf{v}_c$$

Special case 2: Heat kernel (HK) diffusion [Chung'07]

$$\mathbf{f}_c(\boldsymbol{\theta}_{\mathrm{HK}}) = e^{-t}\sum_{k=0}^{K} \frac{t^k}{k!}\mathbf{p}_c^{(k)} \qquad \boldsymbol{\theta}_{\mathrm{HK}} := e^{-t}\left[t \quad \frac{t^2}{2} \quad \cdots \quad \frac{t^K}{K!}\right]^{\mathsf{T}},\ t>0$$

- "Heat" flowing from roots after time *t* ; in steady-state

$$\lim_{K\to\infty} \mathbf{f}_c(\boldsymbol{\theta}_{\mathrm{HK}}) = e^{-t(\mathbf{I}-\mathbf{H})}\mathbf{v}_c$$

- HK and PPR have fixed parameters (t,α)

Our key contribution: Graph- and label-adaptive selection of $\boldsymbol{\theta}_c \in \mathcal{S}^K$

Adaptive diffusions

$$\hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}\in\mathbb{R}^N} \ell(\mathbf{y}_{\mathcal{L}_c}, \mathbf{f}) + \lambda R(\mathbf{f})$$

$$\ell(\mathbf{y}_{\mathcal{L}_c}, \mathbf{f}) = \sum_{i\in\mathcal{L}} \frac{1}{d_i}(y_i - f_i)^2 = (\bar{\mathbf{y}}_{\mathcal{L}_c} - \mathbf{f})^\top \mathbf{D}_{\mathcal{L}}^{-1}(\bar{\mathbf{y}}_{\mathcal{L}_c} - \mathbf{f})$$

Normalized label indicator vector

$$R(\mathbf{f}) = \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\left(\frac{f_i}{d_i} - \frac{f_j}{d_j}\right)^2 = \mathbf{f}^\top\mathbf{D}^{-1}\mathbf{L}\mathbf{D}^{-1}\mathbf{f}$$

- **AdaDIF** scalable to large-scale graphs (*K* << *N*)

$$\hat{\boldsymbol{\theta}}_c = \arg\min_{\boldsymbol{\theta}\in\mathcal{S}^K} \ell(\mathbf{y}_{\mathcal{L}_c}, \mathbf{f}_c(\boldsymbol{\theta})) + \lambda R(\mathbf{f}_c(\boldsymbol{\theta}))$$

- Linear-quadratic $\hat{\boldsymbol{\theta}}_c = \arg\min_{\boldsymbol{\theta}\in\mathcal{S}^K} \boldsymbol{\theta}^\top\mathbf{A}_c\boldsymbol{\theta} + \boldsymbol{\theta}^\top\mathbf{b}_c$

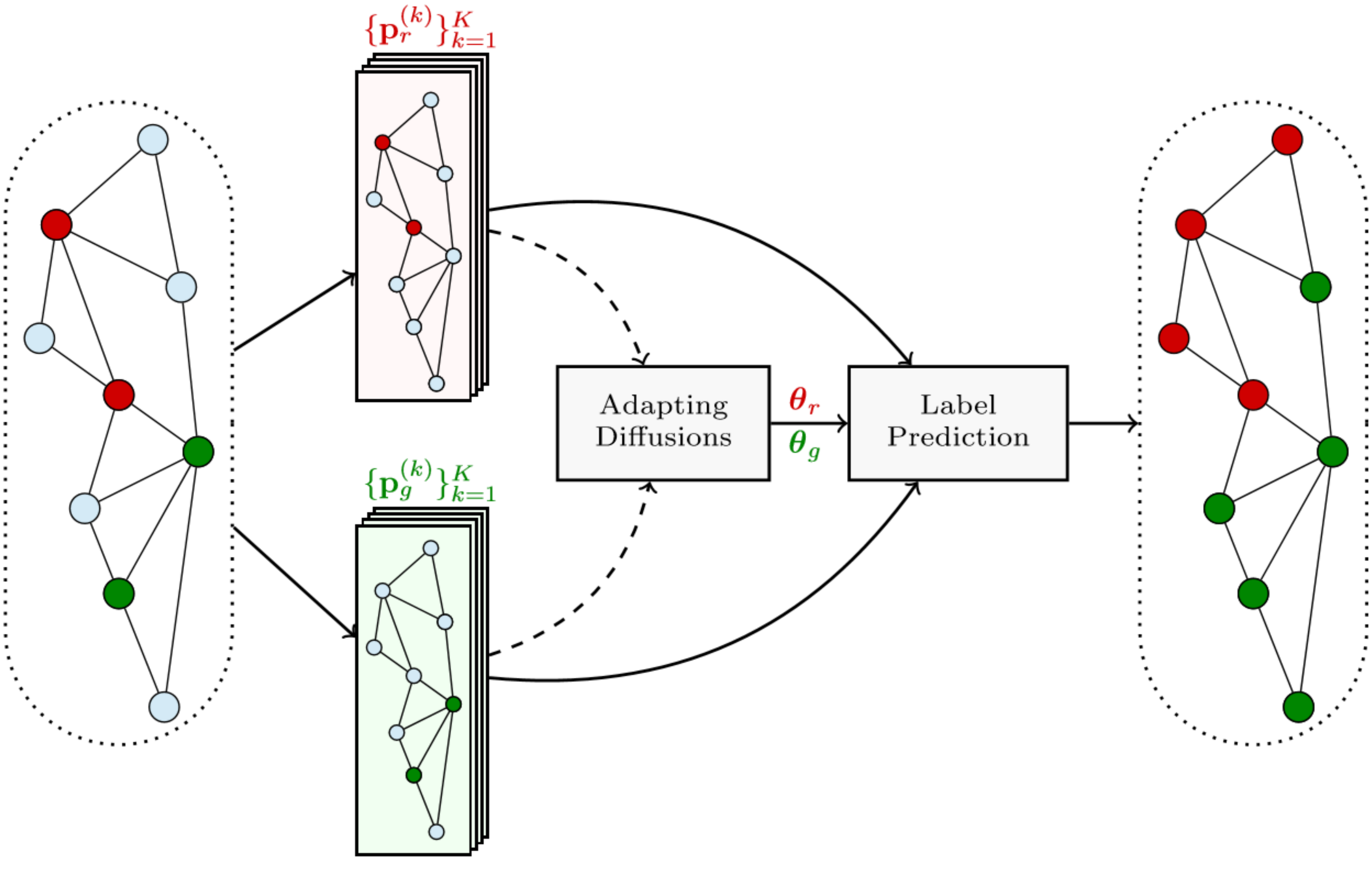
$$\mathrm{b}_c = -\frac{2}{|\mathcal{L}|}(\mathrm{P}_c^{(K)})^\top \mathrm{D}_{\mathcal{L}}^{-1}\mathrm{y}_{\mathcal{L}^c}$$

$$\mathbf{A}_c = (\mathbf{P}_c^{(K)})^\top\left(\mathbf{D}_{\mathcal{L}}^{-1}\mathbf{P}_c^{(K)} + \lambda\mathbf{D}^{-1}\tilde{\mathbf{P}}_c^{(K)}\right)$$

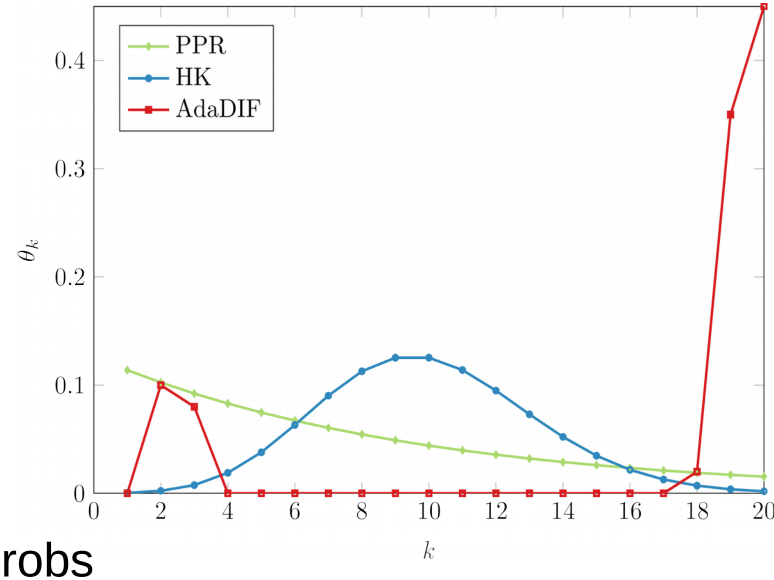
``Differential'' landing prob.

$$\tilde{\mathbf{p}}_c^{(k)} := \mathbf{p}_c^{(k)} - \mathbf{p}_c^{(k+1)}$$

AdaDIF in a nutshell



Interpretation and complexity



$$\hat{\boldsymbol{\theta}}_c = \arg\min_{\boldsymbol{\theta} \in \mathcal{S}^K} \ell(\mathbf{y}_{\mathcal{L}_c}, \mathbf{f}_c(\boldsymbol{\theta})) + \lambda R(\mathbf{f}_c(\boldsymbol{\theta}))$$

- For $\lambda \to \infty$ (smoothness-only), $\hat{\boldsymbol{\theta}}_c \to \mathbf{e}_K$
  - Weight concentrates on last landing prob.
- For $\lambda \to 0$ (fit-only)
  - Weight concentrates on first few landing probs
  - **Intuition:** very short walks visit similarly labeled nodes
- AdaDIF targets a "sweet-spot" between the two
  - Simplex constraint promotes sparsity on $\boldsymbol{\theta}$
- If $K < |\mathcal{E}|/N$, per-class complexity $\mathcal{O}(|\mathcal{E}|K)$ thanks to sparsity of ***H***
  - Same as non-adaptive HK and PPR; also parallelizable across classes
  - Reflect on PPR and Google … just avoid *K* >>

Boosting AdaDIF

- Dictionary of *D* << *K* diffusions

$$\mathbf{f}_c(\boldsymbol{\theta}) = \sum_{k=1}^{K} a_k(\theta)\mathbf{p}_c^{(k)} = \mathbf{P}_c^{(K)}\mathbf{a}(\boldsymbol{\theta}) = \mathbf{P}_c^{(K)}\,\mathbf{C}\boldsymbol{\theta}$$

$$\mathbf{C} := \left[\mathbf{c}_1 \cdots \mathbf{c}_D\right] \in \mathbb{R}^{K \times D}$$

  - Dictionary may include PPR, HK, and more
  - Complexity $\mathcal{O}(|\mathcal{E}|(K+D))$

- Unconstrained diffusions (relax simplex constraints $\theta_i \in \mathbb{R}$)
  - Retain hyperplane constraint to avoid all-zero solution
  - Closed-form solution

$$\hat{\boldsymbol{\theta}}_c = \mathbf{A}_c^{-1}(\mathbf{b}_c - \lambda^* \mathbf{1}) \qquad \lambda^* = \frac{\mathbf{1}^\top \mathbf{A}_c^{-1}\mathbf{b}_c - 1}{\mathbf{b}^\top \mathbf{A}_c^{-1}\mathbf{b}_c}$$

On the choice of *K*

Definition. Let $\mathbf{p}_+$ and $\mathbf{p}_-$ denote respectively the seed vectors for nodes of class "+" and "-," initializing the landing probability vectors in matrices $\mathbf{X}_c := \mathbf{P}_c^{(K)}$, and $\check{\mathbf{X}}_c := \left[\mathbf{p}_c^{(1)} \cdots \mathbf{p}_c^{(K-1)} \mathbf{p}_c^{(K+1)}\right]$, $c \in \{+,-\}$ .. With $\mathbf{y} := \mathbf{X}_+\boldsymbol{\theta} - \mathbf{X}_-\boldsymbol{\theta}$ and $\check{\mathbf{y}} := \check{\mathbf{X}}_+\boldsymbol{\theta} - \check{\mathbf{X}}_-\boldsymbol{\theta}$, the γ-distinguishability threshold of the diffusion-based classifier is the smallest integer K_γ satisfying $\|\mathbf{y} - \check{\mathbf{y}}\| \le \gamma$.

Theorem. For any diffusion-based classifier with coefficients $\boldsymbol{\theta}$ constrained to a probability simplex of appropriate dimensions, it holds that

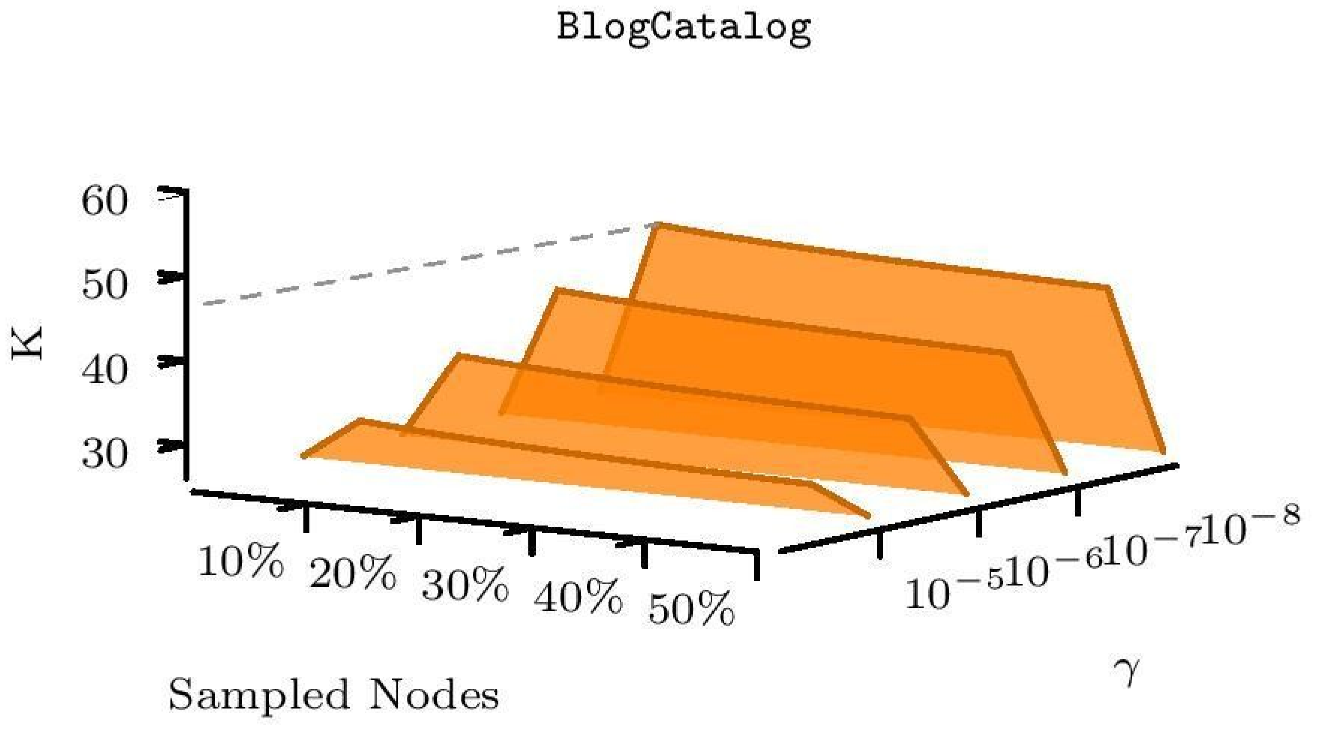
$$K_\gamma \le \frac{1}{\mu'} \log\left[\frac{2\sqrt{d_{\max}}}{\gamma}\left(\sqrt{\frac{1}{d_{\min_-}|\mathcal{L}_-|}} + \sqrt{\frac{1}{d_{\min_+}|\mathcal{L}_+|}}\right)\right]$$

$d_{\min+} := \min_{i \in \mathcal{L}_+} d_i,\ d_{\min-} := \min_{j \in \mathcal{L}_-} d_j,\ d_{\max} := \max_{i \in \mathcal{V}} d_i$ and $\mu' := \min\{\mu_2, 2-\mu_N\}$,

$\{\mu_n\}_{n=1}^N$ eigenvalues of the normalized graph Laplacian in ascending order.

- Message: Increasing *K* does not help distinguishing between classes
  - Large *K* may even degrade performance due to over-parametrization

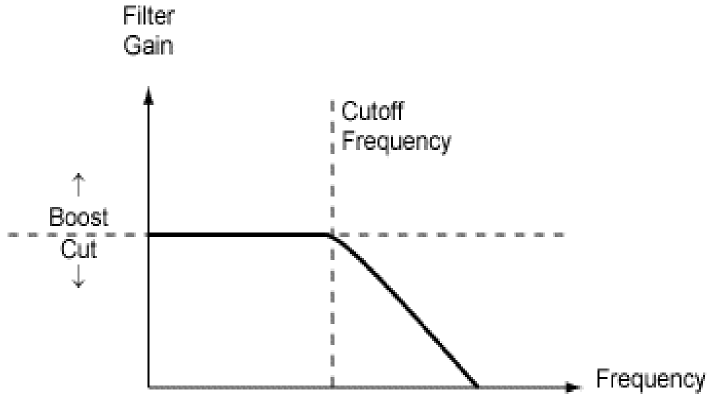
In practice


BlogCatalog
60
50
40
30
K
10%
20%
30%
40%
50%
Sampled Nodes
10^{-5} 10^{-6} 10^{-7} 10^{-8}
γ

Contributions and links with GSP

AdaDif vis-à-vis graph filters [Sandryhaila-Moura ‘13, Chen et al ‘14]

- Different losses and regularizers, including those for outlier resilience
- Multiple class case readily addressed
- AdaDif’s simplex constraint can afford
  - Random walk interpretation
  - Search space reduction
- Rigorous analysis using basic graph properties



AdaDif vis-a-vis GCNs

- Small number of constrained parameters: reduced overfitting
- Simpler and easily parallelizable training: no back propagation
- No feature inputs: operates naturally on graph-only settings

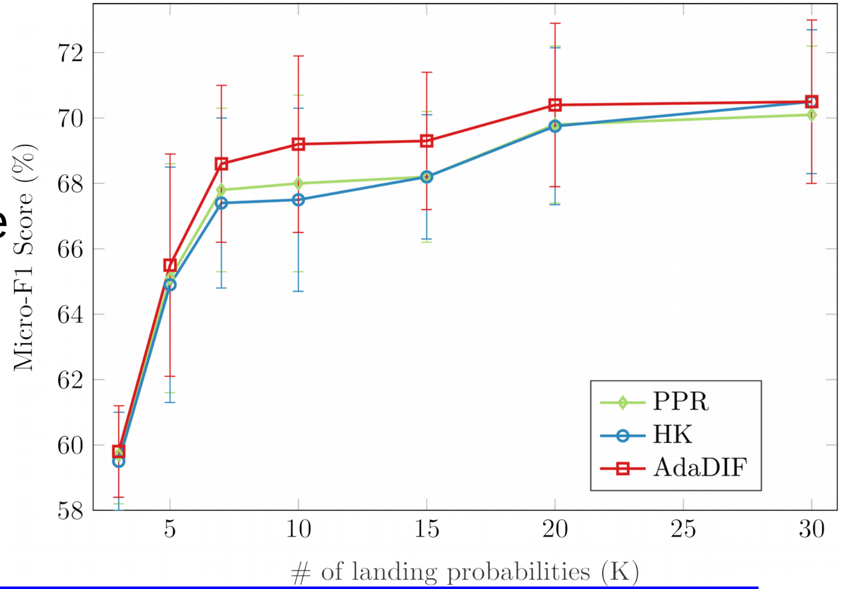
Real data tests

- Real graphs
  - Citation networks
  - Blog networks
  - Protein interaction network

Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{Y}\rvert$	Multi-label
Citeseer	3,233	9,464	6	No
Cora	2,708	10,858	7	No
PubMed	19,717	88,676	3	No
PPI (H. Sapiens)	3,890	76,584	50	Yes
Wikipedia	4,733	184,182	40	Yes
BlogCatalog	10,312	333,983	39	Yes

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot \text{true positive}}{2 \cdot \text{true positive} + \text{false positive} + \text{false negative}}$$

- Micro-F1: node-centric accuracy measure
- Macro-F1: class-centric accuracy measure



- HK and PR run with *K =30* for convergence
  - AdaDIF relies just on *K=15*

Multiclass graphs

- State-of-the-art performance
  - Large margin improvement over Citeseer

	Graph	Cora			Citeseer			PubMed		
	$\|\mathcal{L}\|/\|\mathcal{V}\|$	2.5%	5%	10%	2.5%	5%	10%	0.25%	0.5%	1.0%
Micro-F1	AdaDIF	**70.5 ± 2.4**	**73.7 ± 1.7**	**77.0 ± 1.0**	**51.9 ± 0.9**	**55.1 ± 1.0**	**58.6 ± 0.7**	**72.8 ± 2.4**	**76.1 ± 0.8**	76.5 ± 0.5
	PPR	69.8 ± 2.5	73.3 ± 1.4	**77.0 ± 1.0**	49.7 ± 2.2	53.0 ± 1.5	57.5 ± 0.8	71.4 ± 2.6	74.4 ± 1.1	76.0 ± 0.8
	HK	70.0 ± 2.4	73.5 ± 1.8	76.7 ± 1.2	50.0 ± 2.1	53.5 ± 1.5	57.3 ± 0.9	**72.8 ± 2.6**	75.1 ± 1.0	**76.8 ± 0.7**
	Node2vec	69.5 ± 1.8	73.0 ± 1.6	75.5 ± 1.4	46.0 ± 2.7	49.7 ± 1.7	52.1 ± 1.4	**72.8 ± 2.8**	74.8 ± 1.6	75.1 ± 1.4
	Deepwalk	68.2 ± 2.5	72.1 ± 1.8	74.9 ± 1.2	45.0 ± 2.4	48.5 ± 1.7	51.2 ± 1.2	72.4 ± 2.6	73.8 ± 1.3	74.5 ± 1.2
	Planetoid-G	62.5 ± 5.1	67.3 ± 4.3	75.8 ± 1.1	43.0 ± 1.8	46.8 ± 1.9	55.2 ± 1.3	63.4 ± 3.7	65.2 ± 2.0	67.8 ± 1.5
	GCN	58.3 ± 4.0	66.5 ± 2.1	71.3 ± 1.7	38.9 ± 2.7	44.5 ± 2.0	50.3 ± 1.6	57.7 ± 3.4	64.5 ± 2.7	70.0 ± 1.5
Macro-F1	AdaDIF	**69.0 ± 2.3**	**72.3 ± 1.8**	**75.7 ± 1.2**	**46.6 ± 1.1**	**49.6 ± 1.6**	**53.9 ± 1.0**	**71.5 ± 2.5**	**74.2 ± 0.7**	75.2 ± 0.8
	PPR	66.7 ± 4.2	71.8 ± 1.6	75.3 ± 1.1	44.1 ± 2.0	48.4 ± 1.5	53.5 ± 0.8	69.5 ± 2.6	72.8 ± 1.1	74.7 ± 0.8
	HK	67.1 ± 4.2	72.1 ± 1.9	75.5 ± 1.4	44.8 ± 2.0	48.9 ± 1.5	53.7 ± 1.0	71.0 ± 2.6	73.5 ± 1.1	**75.6 ± 0.8**
	Node2vec	67.1 ± 2.6	71.6 ± 1.8	74.0 ± 1.3	42.6 ± 2.5	46.6 ± 1.7	48.7 ± 1.3	70.3 ± 3.2	73.0 ± 1.8	73.5 ± 1.4
	Deepwalk	66.1 ± 3.2	70.5 ± 2.1	73.8 ± 1.4	41.6 ± 2.4	45.5 ± 1.5	48.5 ± 1.2	70.0 ± 3.2	72.0 ± 1.7	73.1 ± 1.3
	Planetoid-G	58.0 ± 5.1	64.3 ± 4.3	74.3 ± 1.6	37.4 ± 2.1	41.6 ± 2.2	52.0 ± 2.4	61.0 ± 3.9	63.7 ± 3.0	65.2 ± 2.0
	GCN	52.0 ± 6.8	61.9 ± 2.6	64.8 ± 1.9	33.0 ± 3.0	39.2 ± 1.7	43.3 ± 1.6	52.1 ± 4.4	60.2 ± 3.9	65.3 ± 2.2

Multilabel graphs

- Number of labels per node assumed known (typical)
  - Evaluate accuracy of top-ranking classes

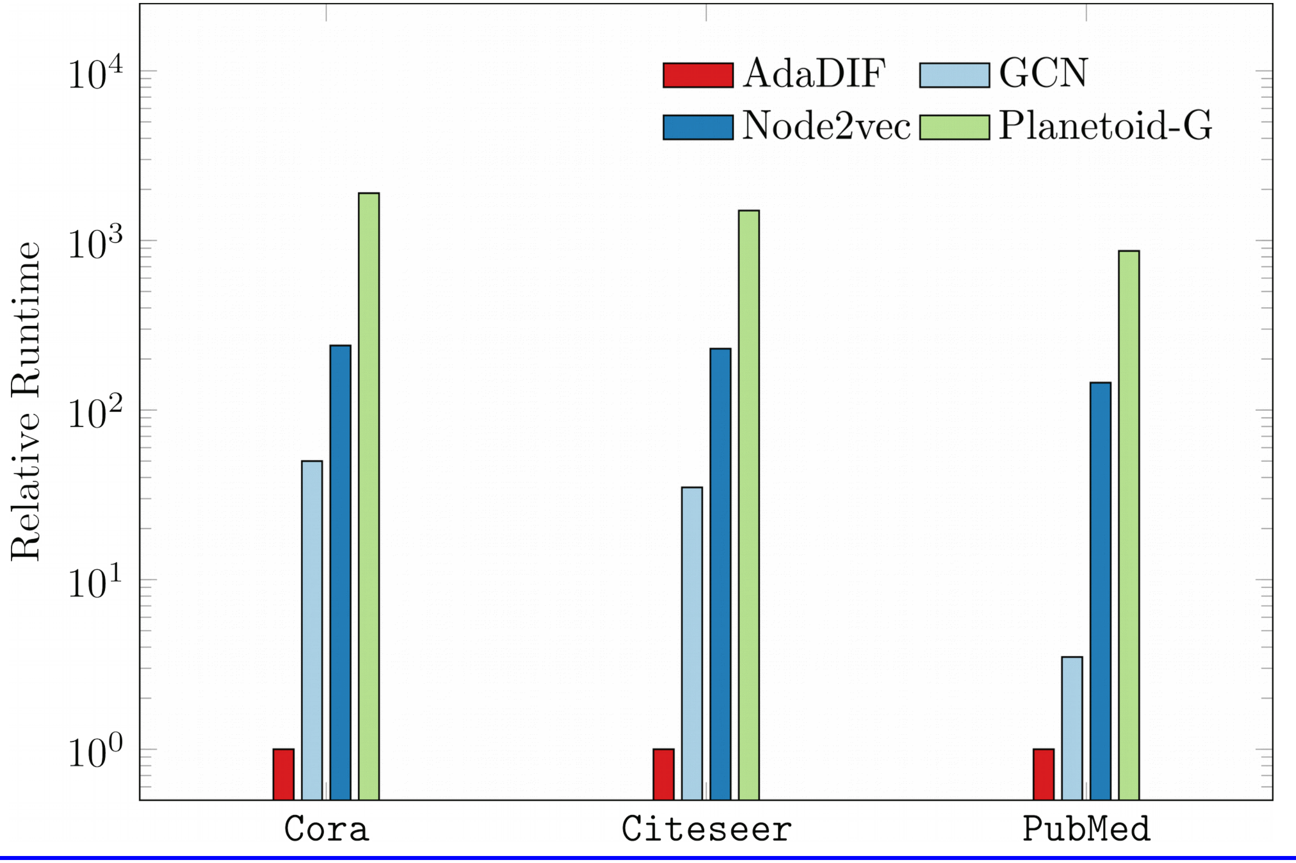
- AdaDIF approaches Node2vec Micro-F1 accuracy for PPI and BlogCatalog
  - Significant improvement over non-adaptive PPR and HK for all graphs

- AdaDIF achieves state-of-the-art Macro-F1 performance

	Graph	PPI			BlogCatalog			Wikipedia		
	$\lvert\mathcal{L}\rvert/\lvert\mathcal{V}\rvert$	10%	20%	30%	10%	20%	30%	10%	20%	30%
Micro-F1	AdaDIF	15.4 ± 0.5	17.9 ± 0.7	$\mathbf{19.2 \pm 0.6}$	31.5 ± 0.6	34.4 ± 0.5	36.3 ± 0.4	28.2 ± 0.9	30.0 ± 0.5	31.2 ± 0.7
	PPR	13.8 ± 0.5	15.8 ± 0.6	17.0 ± 0.4	21.1 ± 0.8	23.6 ± 0.6	25.2 ± 0.6	10.5 ± 1.5	8.1 ± 0.7	7.2 ± 0.5
	HK	14.5 ± 0.5	16.7 ± 0.6	18.1 ± 0.5	22.2 ± 1.0	24.7 ± 0.7	26.6 ± 0.7	9.3 ± 1.4	7.3 ± 0.7	6.0 ± 0.7
	Node2vec	$\mathbf{16.5 \pm 0.6}$	$\mathbf{18.2 \pm 0.3}$	19.1 ± 0.3	$\mathbf{35.0 \pm 0.3}$	$\mathbf{36.3 \pm 0.3}$	$\mathbf{37.2 \pm 0.2}$	$\mathbf{42.3 \pm 0.9}$	$\mathbf{44.0 \pm 0.6}$	$\mathbf{45.1 \pm 0.4}$
	Deepwalk	16.0 ± 0.6	17.9 ± 0.5	18.8 ± 0.4	34.2 ± 0.4	35.7 ± 0.3	36.4 ± 0.4	41.0 ± 0.8	43.5 ± 0.5	44.1 ± 0.5
Macro-F1	AdaDIF	$\mathbf{13.4 \pm 0.6}$	$\mathbf{15.4 \pm 0.7}$	$\mathbf{16.5 \pm 0.7}$	$\mathbf{23.0 \pm 0.6}$	$\mathbf{25.3 \pm 0.4}$	$\mathbf{27.0 \pm 0.4}$	$\mathbf{7.7 \pm 0.3}$	$\mathbf{8.3 \pm 0.3}$	$\mathbf{9.0 \pm 0.2}$
	PPR	12.9 ± 0.4	14.7 ± 0.5	15.8 ± 0.4	17.3 ± 0.5	19.5 ± 0.4	20.8 ± 0.3	4.4 ± 0.3	3.8 ± 0.6	3.6 ± 0.2
	HK	$\mathbf{13.4 \pm 0.6}$	$\mathbf{15.4 \pm 0.5}$	$\mathbf{16.5 \pm 0.4}$	18.4 ± 0.6	20.7 ± 0.4	22.3 ± 0.4	4.2 ± 0.4	3.7 ± 0.5	3.5 ± 0.2
	Node2vec	13.1 ± 0.6	15.2 ± 0.5	16.0 ± 0.5	16.8 ± 0.5	19.0 ± 0.3	20.1 ± 0.4	7.6 ± 0.3	8.2 ± 0.3	8.5 ± 0.3
	Deepwalk	12.7 ± 0.7	15.1 ± 0.6	16.0 ± 0.5	16.6 ± 0.5	18.7 ± 0.5	19.6 ± 0.4	7.3 ± 0.3	8.1 ± 0.2	8.2 ± 0.2

Runtime comparison

- AdaDIF can afford **much lower runtimes**
  - Even without parallelization!



Leave-one-out fitting loss

- Quantifies how well each (labeled) node is predicted by the rest

$$\ell^c_{\text{rob}}(\mathbf{y}_{\mathcal{L}_c}, \boldsymbol{\theta}) := \sum_{i \in \mathcal{L}} \frac{1}{d_i} \left([\bar{\mathbf{y}}_{\mathcal{L}_c}]_i - [\mathbf{f}_c(\boldsymbol{\theta}; \mathcal{L} \setminus i)]_i \right)^2$$

- $\mathbf{f}_c(\boldsymbol{\theta}; \mathcal{L} \setminus i)$'s obtained via $|\mathcal{L}|$ different random walks ($\mathcal{O}(|\mathcal{L}|K|\mathcal{E}|)$)

$$\mathbf{f}_c(\boldsymbol{\theta}; \mathcal{L} \setminus i) = \begin{cases} \mathbf{f}_c(\boldsymbol{\theta}), & i \notin \mathcal{L}_c \\ \mathbf{f}_c(\boldsymbol{\theta}; \mathcal{L}_c \setminus i), & i \in \mathcal{L}_c \end{cases} \quad \mathbf{f}_c(\boldsymbol{\theta}; \mathcal{L}_c \setminus i) = \sum_{k=1}^{K} \theta_k \mathbf{p}^{(k)}_{\mathcal{L}_c \setminus i}$$

$$\mathbf{p}^{(k)}_{\mathcal{L}_c \setminus i} := \mathbf{H}^k \mathbf{v}_{\mathcal{L}_c \setminus i} \qquad [\mathbf{v}_{\mathcal{L}_c \setminus i}]_j = \begin{cases} 1/|\mathcal{L}_c \setminus i|, & j \in \mathcal{L}_c \setminus i \\ 0, & \text{else} \end{cases}$$

- Compact form

$$\ell^c_{\text{rob}}(\mathbf{y}_{\mathcal{L}_c}, \boldsymbol{\theta}) := \|\mathbf{D}_{\mathcal{L}}^{-\frac{1}{2}} \left(\bar{\mathbf{y}}_{\mathcal{L}_c} - \mathbf{R}_c^{(K)} \boldsymbol{\theta} \right) \|_2^2 \qquad \left[\mathbf{R}_c^{(K)}\right]_{ik} := \begin{cases} \left[\mathbf{p}^{(k)}_{\mathcal{L}_c \setminus i}\right]_i, & i \in \mathcal{L}_c \\ \left[\mathbf{p}^{(k)}_c\right]_i, & \text{else} \end{cases}$$

- Diffusion parameters

$$\hat{\boldsymbol{\theta}}_c = \arg \min_{\boldsymbol{\theta} \in \mathcal{S}^K} \ell^c_{\text{rob}}(\mathbf{y}_{\mathcal{L}_c}, \boldsymbol{\theta}) + \lambda_\theta \|\boldsymbol{\theta}\|_2^2$$

Anomaly identification - removal

- Model outliers as large residuals, captured by nnz entries of sparse vec. $\mathbf{o} \in \mathbb{R}^N$

$$\ell^c_{\text{rob}}(\mathbf{y}_{\mathcal{L}_c}, \mathbf{o}, \boldsymbol{\theta}) := \|\mathbf{D}_{\mathcal{L}}^{-\frac{1}{2}} \left(\mathbf{o} + \bar{\mathbf{y}}_{\mathcal{L}_c} - \mathbf{R}_c^{(K)}\boldsymbol{\theta}\right)\|_2^2$$

- Joint optimization

$$\{\hat{\boldsymbol{\theta}}_c, \hat{\mathbf{o}}_c\}_{c\in\mathcal{Y}} = \arg \min_{\substack{\boldsymbol{\theta}_c \in \mathcal{S}^K \\ \mathbf{o}_c \in \mathbb{R}^N}} \sum_{c\in\mathcal{Y}} \left[\ell^c_{\text{rob}}(\mathbf{y}_{\mathcal{L}_c}, \mathbf{o}_c, \boldsymbol{\theta}_c) + \lambda_\theta \|\boldsymbol{\theta}_c\|_2^2\right] + \lambda_o \|\mathbf{D}_{\mathcal{L}}^{-\frac{1}{2}}\mathbf{O}\|_{2,1}$$

Group sparsity on

$\mathbf{O} := [\mathbf{o}_1 \cdots \mathbf{o}_{|\mathcal{Y}|}]$

i.e., force consensus among classes regarding which nodes are outliers

- While, $\|\hat{\boldsymbol{\theta}}_c^{(t)} - \hat{\boldsymbol{\theta}}_c^{(t-1)}\|_\infty \leq \epsilon, \ \forall c \in \mathcal{Y}$ iterate:

$$\hat{\boldsymbol{\theta}}_c^{(t)} = \arg \min_{\boldsymbol{\theta}\in\mathcal{S}^K} \ell^c_{\text{rob}}(\bar{\mathbf{y}}_{\mathcal{L}_c} + \hat{\mathbf{o}}_c^{(t-1)}, \boldsymbol{\theta}) + \lambda_\theta \|\boldsymbol{\theta}\|_2^2$$

$$\hat{\mathbf{O}}^{(t)} = \text{SoftThres}_{\lambda_o}\left(\tilde{\mathbf{Y}}^{(t)}\right)$$

Residuals

$$\tilde{\mathbf{Y}}^{(t)} := \left[\tilde{\mathbf{y}_1}^{(t)}, \ldots, \mathbf{y}_{|\mathcal{Y}|}^{(t)}\right]$$

$$\tilde{\mathbf{y}}_c^{(t)} := \bar{\mathbf{y}}_{\mathcal{L}_c} - \mathbf{R}_c^{(K)}\hat{\boldsymbol{\theta}}_c^{(t)}$$

Row-wise soft-thresholding

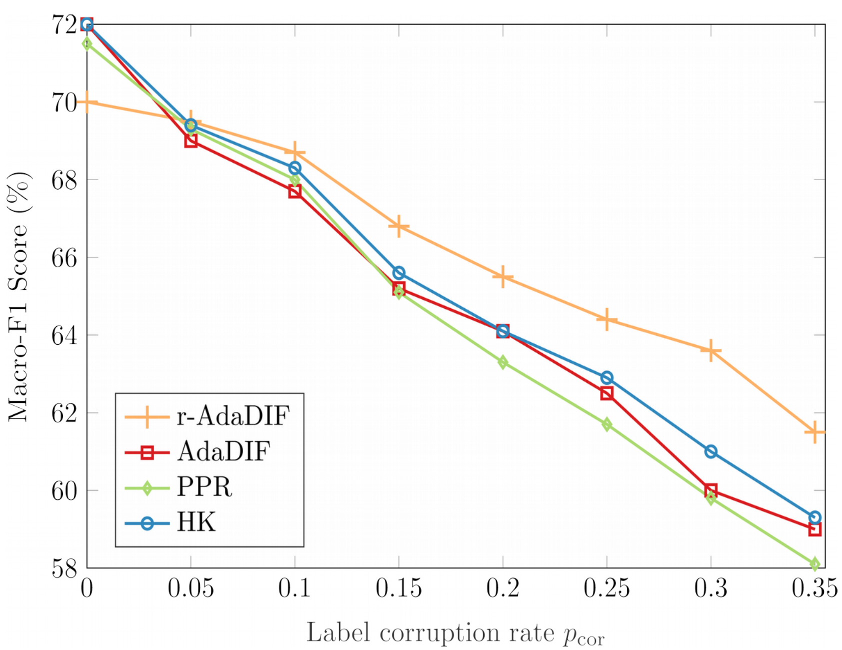
$$\mathbf{Z} = \text{SoftThres}_{\lambda_o}(\mathbf{X})$$

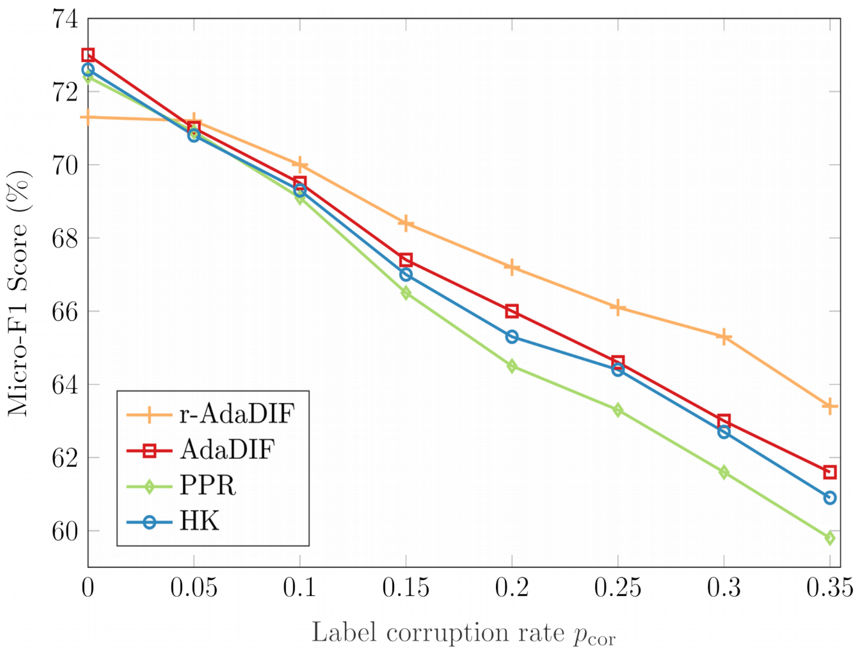
$$\mathbf{z}_i = \|\mathbf{x}_i\|_2[1 - \lambda_o/(2\|\mathbf{x}_i\|_2)]_+$$

- Alternating minimization converges to stationary point
- Remove outliers $\mathcal{S} := \{i \in \mathcal{L} : \|[\hat{\mathbf{O}}]_{i,:}\|_2 > 0\}$ from $\mathcal{L}$ and predict $\mathcal{U}$ using $\{\hat{\boldsymbol{\theta}}_c\}_{c\in\mathcal{Y}}$

Testing classification performance

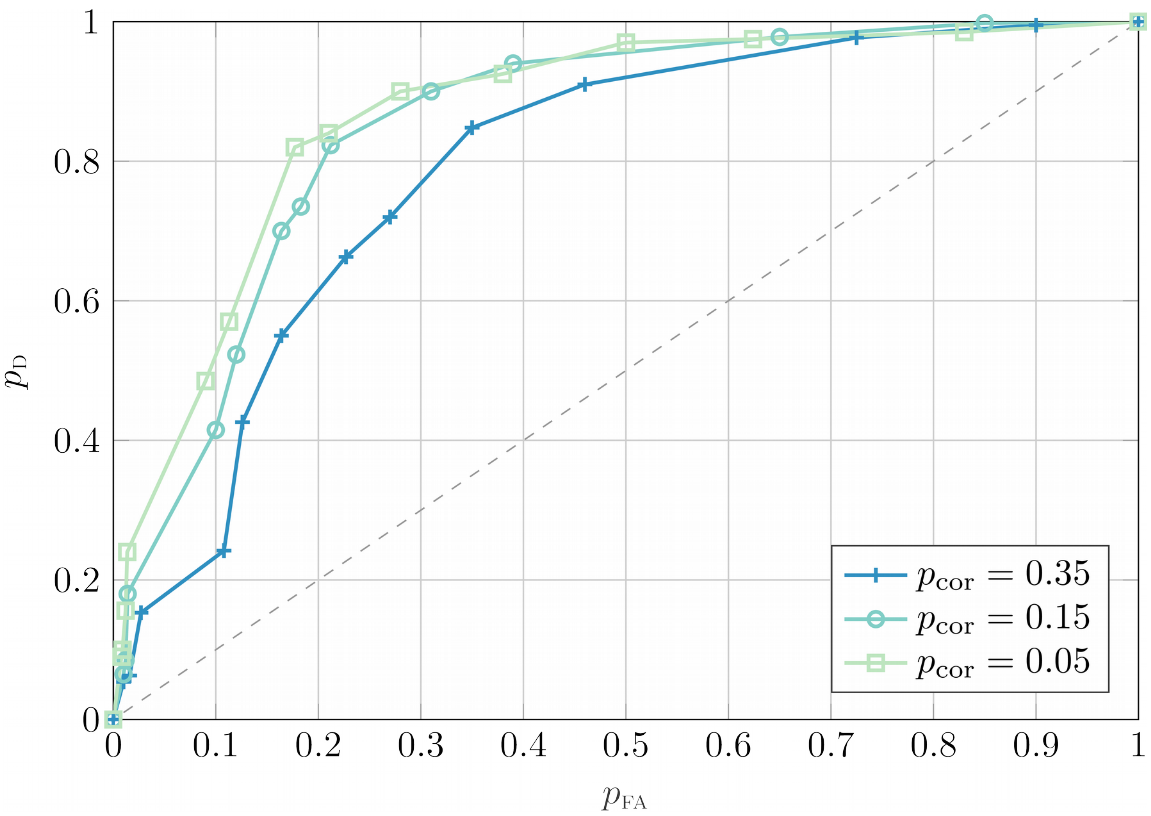
- Anomalies injected in Cora graph
  - Go through each entry $[\mathbf{y}_{\mathcal{L}}]_i = c$ of $\mathbf{y}_{\mathcal{L}}$
  - With probability p_{cor} draw a label $c' \sim \text{Unif}\{\mathcal{Y} \setminus c\}$
  - Replace $[\mathbf{y}_{\mathcal{L}}]_i \leftarrow c'$
- For fixed $\lambda_o > 0$, accuracy with $p_{\text{cor}} > 0$ improves as false samples are removed
  - Less accuracy for $p_{\text{cor}} = 0$ (no anomalies), only useful samples removed (false alarms)





Testing anomaly detection performance

- **ROC** curve: Probability of detection vs probability of false alarms
  - As expected, performance improves as p_{cor} decreases



Research outlook

- Investigate different losses and diverse regularizers
- Further boost accuracy with nonlinear diffusion models
- Effect reduced complexity and memory requirements via approximations
- Online AdaDIF for dynamic graphs

Thank you!