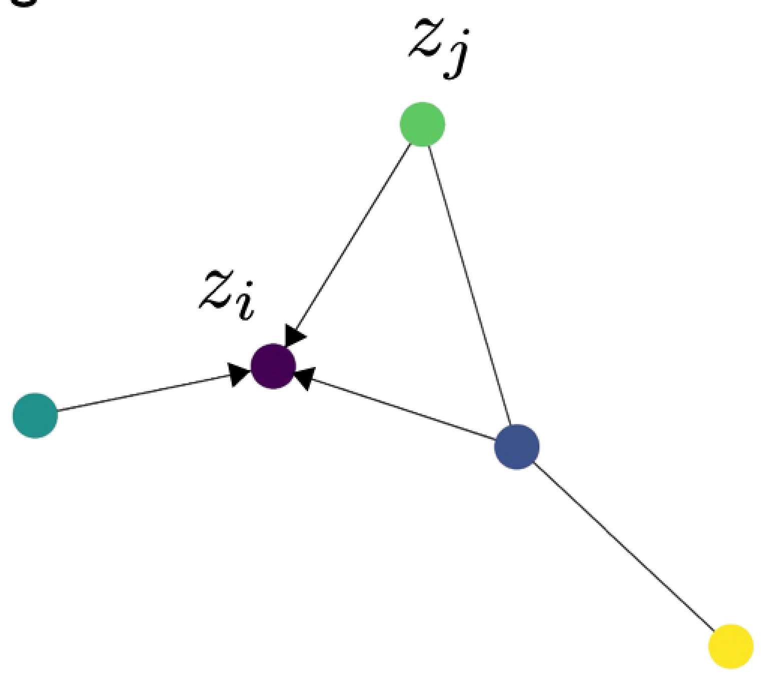
Not too little, not too much: a theoretical analysis of graph (over)smoothing

Khushee Kapoor

Message Passing Neural Networks

Graph Neural Networks (GNNs) work mostly by **Message-Passing**:

$$z_i^{(k)} = \mathrm{AGG}_{\theta_k}\left(z_i^{(k-1)}, \{z_j^{(k-1)}\}_{j \in \mathcal{N}_i}\right)$$

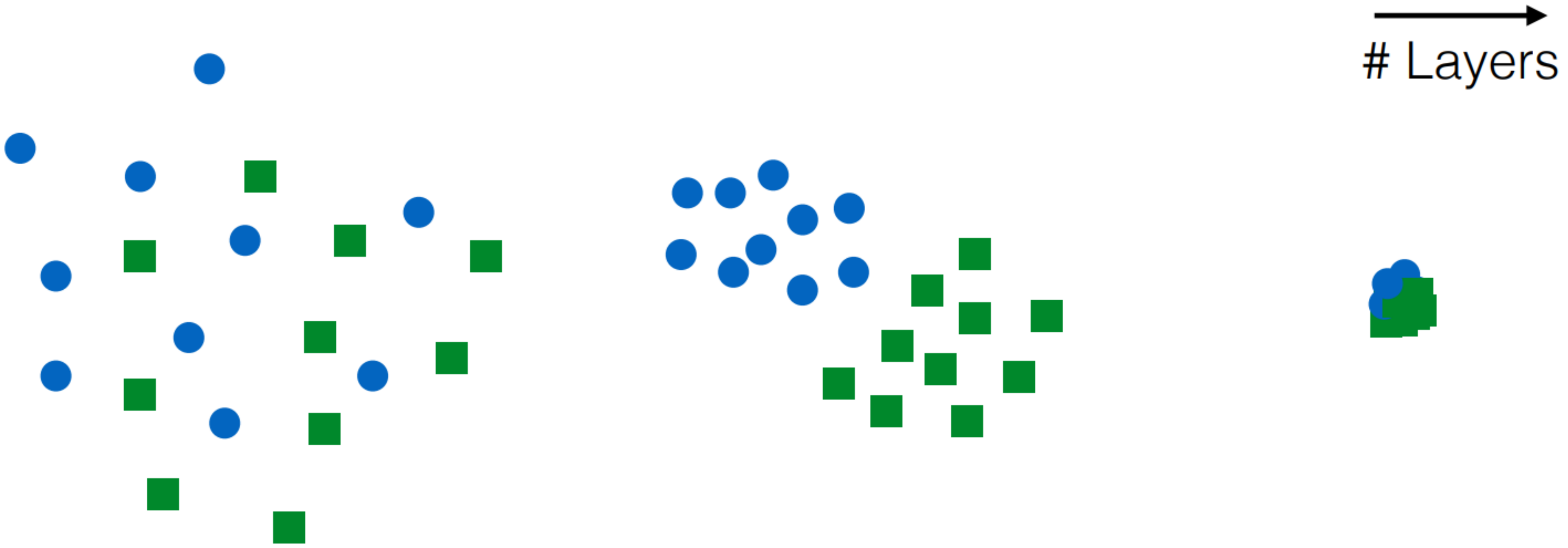


Here we use classic **mean aggregation**:

$$z_i^{(k)} = \frac{1}{\sum_j a_{ij}} \sum_j a_{ij} \Psi_{\theta_k}\left(z_j^{(k-1)}\right)$$

Note that this is just $Z^{(k)} = L\Psi_{\theta_k}(Z^{(k-1)})$ with $L = D^{-1}A$

Smoothing Effect of Graph Convolution

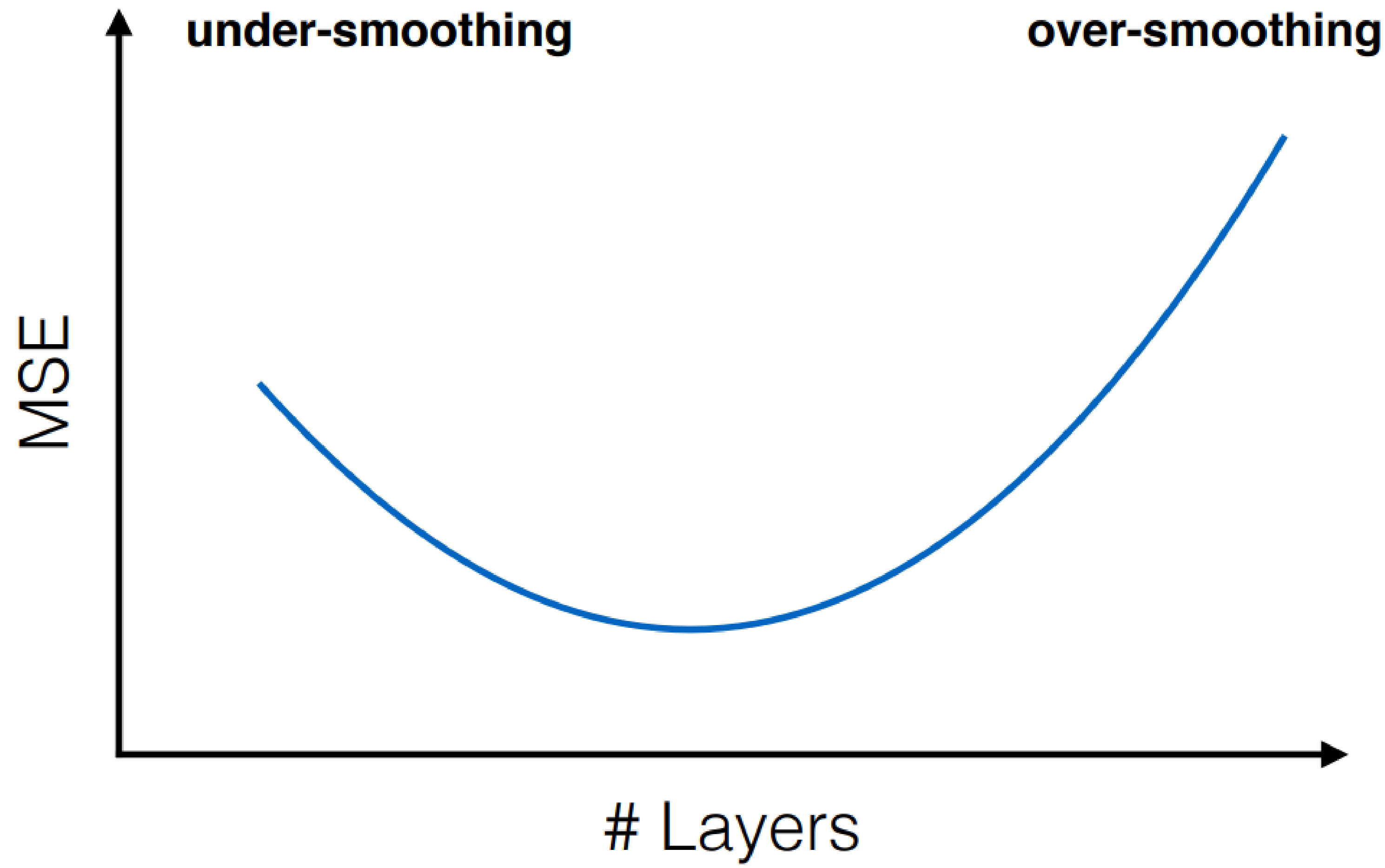


Not enough smoothing

Right amount of smoothing

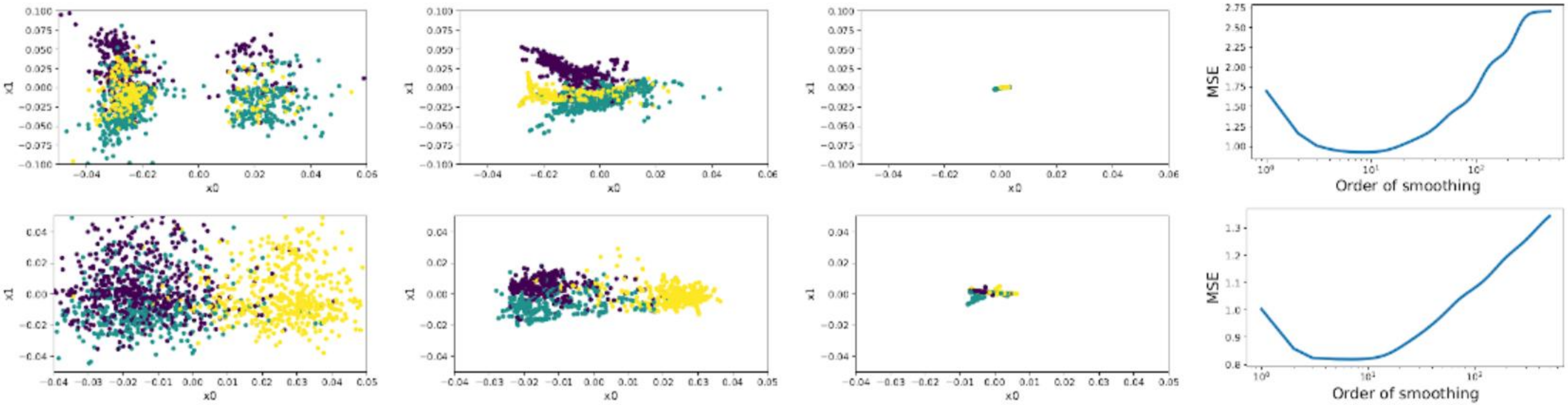
Too much smoothing

Smoothing Effect of Graph Convolution



Oversmoothing vs Sufficient Depth

- Oversmoothing restricts GNNs depth: node representations often **provably** converge to a constant... But **some** smoothing helps in practice!



*Learning on increasingly smoothed features on **Cora** and **Pubmed**, illustrated by principal components.*

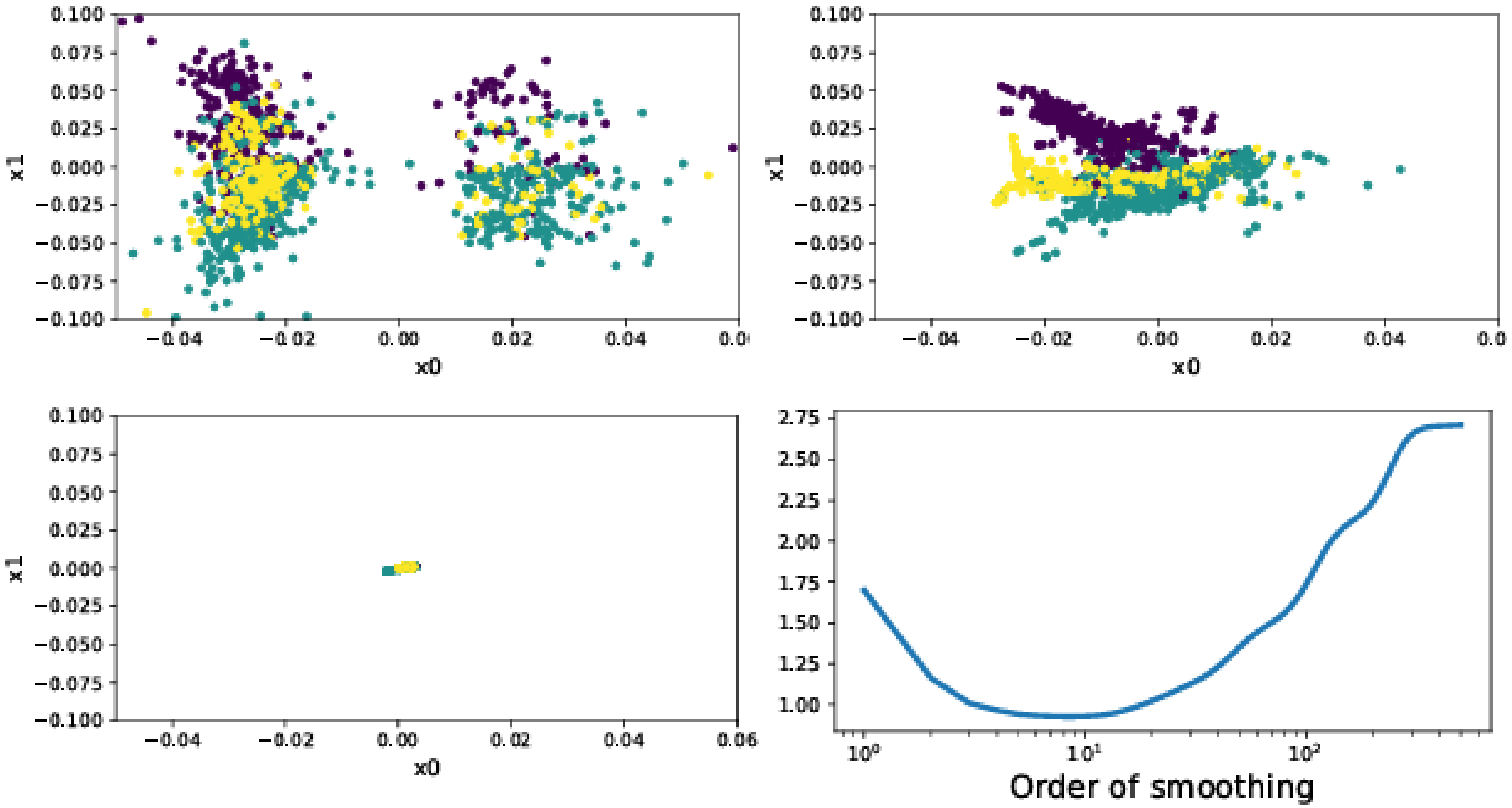
Overmsmoothing vs Sufficient Depth

- Theoretical analyses of GNNs are generally **"in the deep limit"**
  - **Sufficiently deep** GNNs can be as powerful as 1-WL [1]
  - "Well-engineered" GNNs can model useful **diffusion processes** [2]
- Of course, only valid when oversmoothing does **not** occur...
  - i.e., not with usual aggregation functions like average over neighbors

Overssmoothing vs Sufficient Depth

Oversmoothing is a well-studied phenomenon "preventing" GNNs from being "too deep" in practice. E.g., for mean aggregation: $L^k Z \xrightarrow[k\to\infty]{} c1_n$

Cora



Overversmoothing vs Sufficient Depth

But... most analyses showing the power of GNNs **take the limit** $k \to \infty$ **!**

(*not* for mean aggregation, obviously)

- sufficiently deep GNNs are "**Weisfeiler-Lehman**" powerful *[Xu et al. 2019]*
- some GNNs model a **diffusion process** that separates well data, etc

[Bodnar et al. 2022]

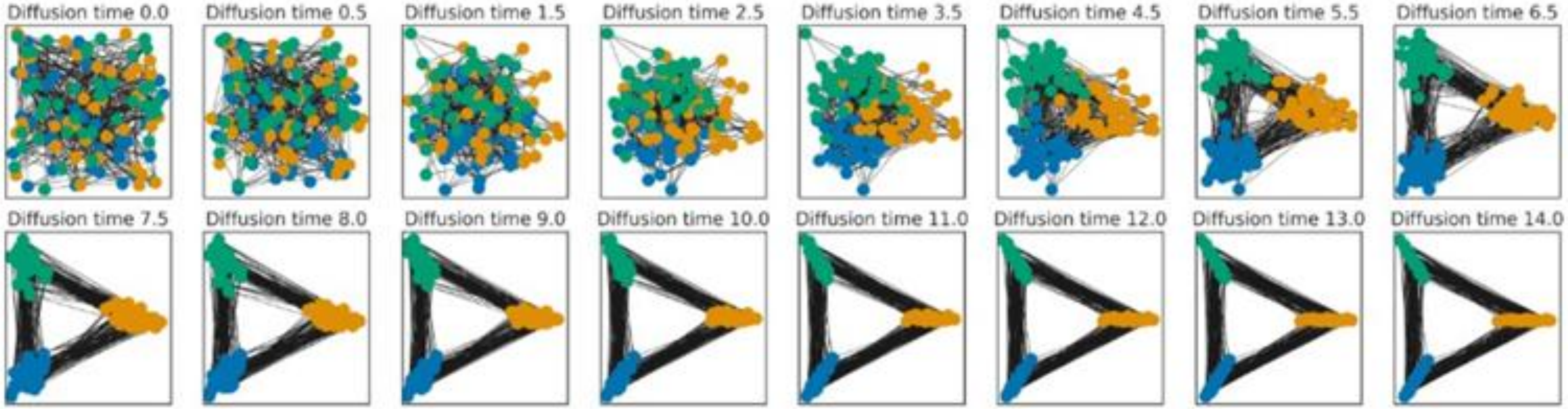
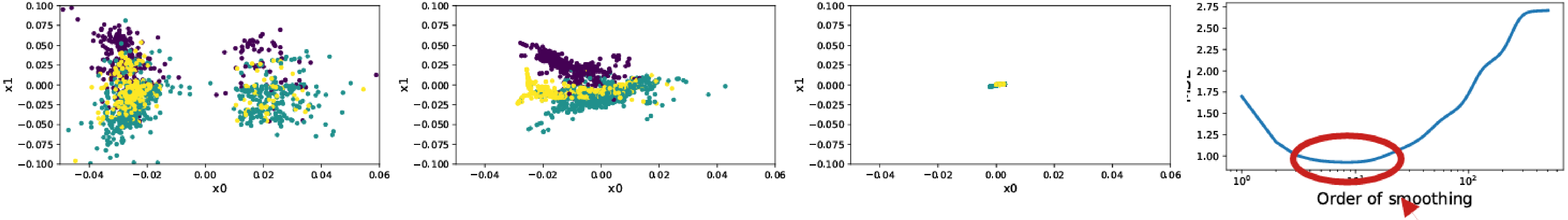


Figure 7. Sheaf diffusion process disentangling the $C = 3$ classes over time. The nodes are coloured by their class.

Oversmoothing vs Sufficient Depth

Can "good smoothing" and oversmoothing co-exist? Why?



Middle regime?

Take-home message: smoothing collapses node features,
but not everything collapses at the same speed

Latent Space Random Graph Model

Each node i is associated with a **latent variable** $x_i \in \mathbb{R}^d$ with $d \gg p$.

A complete graph with **edge weights**

$$a_{ij} = W(x_i, x_j) = \epsilon + e^{-\frac{1}{2}\|x_i - x_j\|_2^2}$$

Latent variables and node labels (x_i, y_i) are drawn iid from some distribution.

Node features are a linear projection of the latent variables to a lower dimension:

$$z_i = M^T x_i$$

for some unknown $M \in \mathbb{R}^{d \times p}$ and $M^T M = I$.

Over-smoothing as $k \to \infty$

$L = D^{-1}A$ is a stochastic matrix,

$$L^k \to 1_n \bar{d}^T, \text{ where } \bar{d}_i = \frac{\text{degree of node } i}{\text{sum of all degrees}}.$$

Therefore $Z^{(k)} = L^k Z \to 1_n \bar{d}^T Z$, i.e., each node has identical representation.

Since the test nodes have identical representations, the predictions will be the same: $\hat{y}_{te}^{(k)} = Z^{(k)} \hat{\beta}^{(k)} \to c 1_{n_{te}}$ for some constant c.

Using the closed-form solution for the ridge regression can get

$$c = \frac{1}{n_{tr}} \left(\frac{\|Z^T \bar{d}\|_2^2}{\|Z^T \bar{d}\|_2^2 + \lambda} \right) \boxed{\sum_{i=1}^{n_{tr}} y_i}$$ average training labels

Model of Random Graph

We model **both phenomena** at once:

We give simple examples in which finite smoothing provably helps learning, before oversmoothing kicks in.

- Smoothing collapses features, but not everything collapse at the same speed
- **Some subspaces may collapse faster**, which helps regression
- **Communities collapse onto themselves**, which helps classification

- Unknown **latent variables** $x_i \in \mathbb{R}^d$, labels $y_i \in \mathbb{R}$:

$$(x_i, y_i) \overset{iid}{\sim} P, \quad 1 \leqslant i \leqslant n$$

- **Graph structure**: Gaussian kernel

$$a_{ij} = W(x_i, x_j), \quad W(x, x') = e^{-\|x-x'\|^2} + \epsilon$$

  - For simplicity: no "Bernoulli edges" & small $\epsilon > 0$

- **Node features**: **partial observation**

$$z_i = Mx_i \in \mathbb{R}^p, \quad p < d$$

  - Does **not** satisfy JL lemma: loss of information

Can smoothing (and only smoothing, here) help recover lost information before oversmoothing dominates?

Model of Random Graph

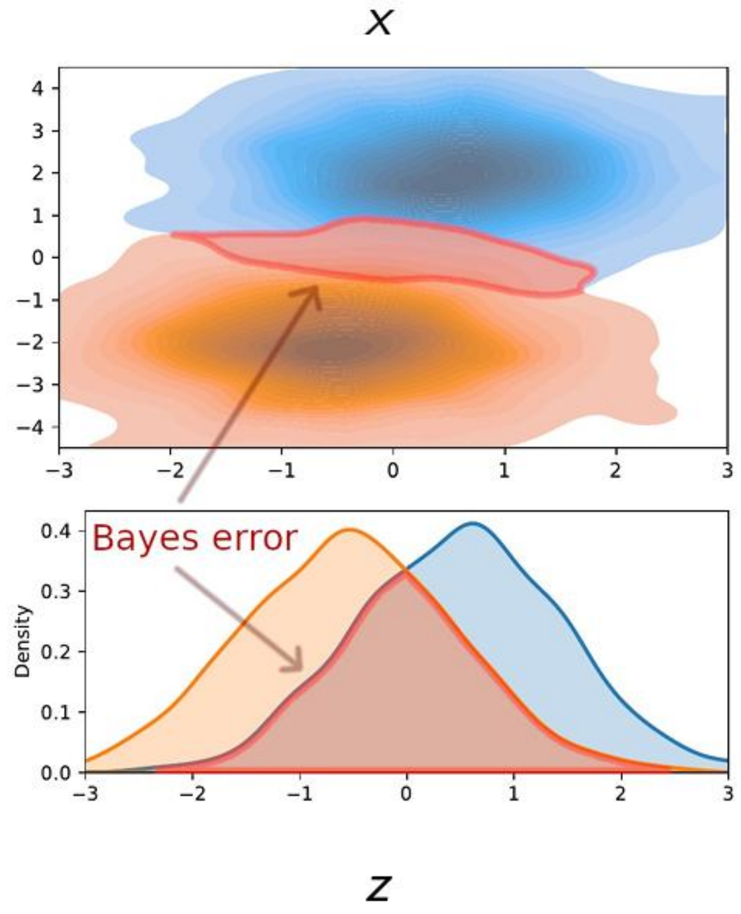
Random graph model:

$$\boxed{(x_i, y_i) \sim P, \;\; a_{ij} = W(x_i, x_j), \;\; z_i = Mx_i}$$

With $\quad M \in \mathbb{R}^{p \times d}, \quad p < d \qquad W(x, x') = e^{-\|x - x'\|^2} + \epsilon$

No Johnson-Lindenstrauss here. There **is** loss of information in the node features.

*Can **mean aggregation** recover some of the information **before oversmoothing occurs** ?*



Linear GCN

We will focus on **linear GCN** with a Mean Square Error (MSE) loss.

Given an input matrix $Z \in \mathbb{R}^{n \times p}$, the output after k rounds of mean aggregation is

$$Z^{(k)} = L^k Z$$

where $L = D^{-1}A$ is the normalized adjacency matrix.

Consider learning with MSE loss and ridge regularization

$$\min_{\beta} \frac{1}{2n_{tr}} \left\| y_{tr} - Z_{tr}^{(k)} \beta \right\|_2^2 + \lambda \|\beta\|_2^2$$

where subscript tr means training.

Test Risk

Denote

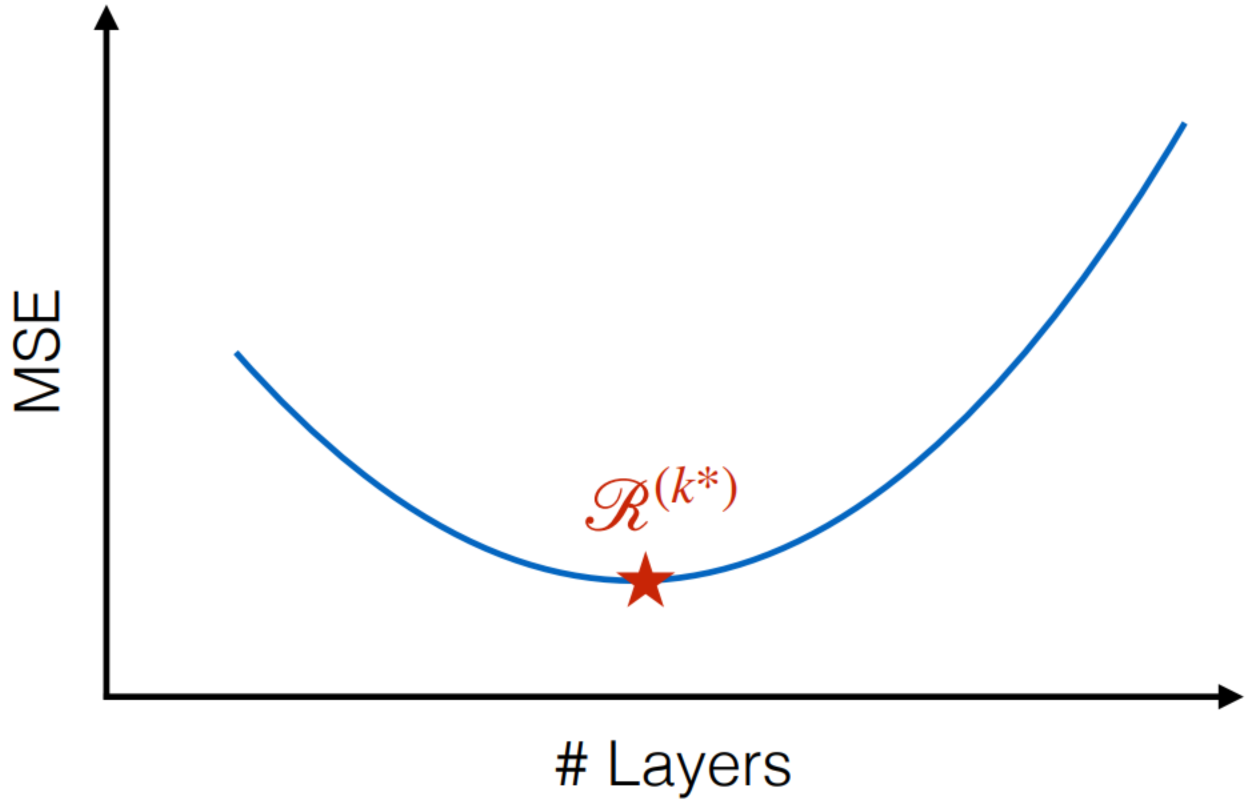
$$\hat{\beta}^{(k)} = \mathrm{argmin}_{\beta} \frac{1}{2n_{tr}} \left\| y_{tr} - Z_{tr}^{(k)}\beta \right\|_2^2 + \lambda\|\beta\|_2^2$$

The test risk is

$$\mathscr{R}^{(k)} = \frac{1}{n_{te}} \left\| y_{te} - Z_{te}^{(k)}\hat{\beta}^{(k)} \right\|_2^2$$

- $\mathscr{R}^{(0)}$ is the test risk without any GCN layer
- $\mathscr{R}^{(\infty)}$ denotes the asymptotic test risk as $k \to \infty$
- **Over-smoothing:** $\mathscr{R}^{(0)} < \mathscr{R}^{(\infty)}$
- **Key result in this paper:** $\exists k^* \geq 1$ such that $\mathscr{R}^{(k^*)} < \min\{\mathscr{R}^{(0)}, \mathscr{R}^{(\infty)}\}$

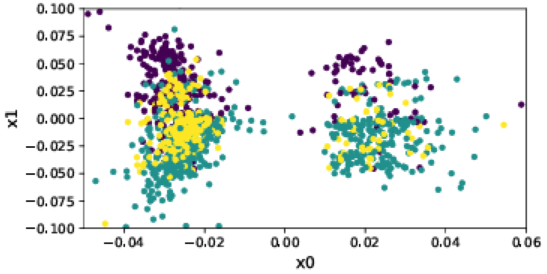
Theoretical Risk

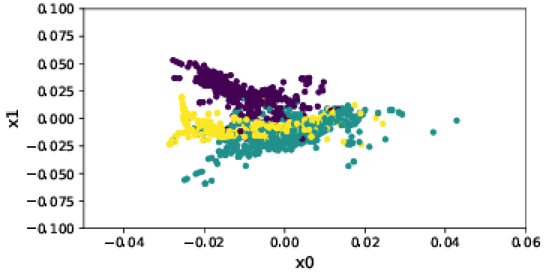


Settings: Ridge Regression and SSL

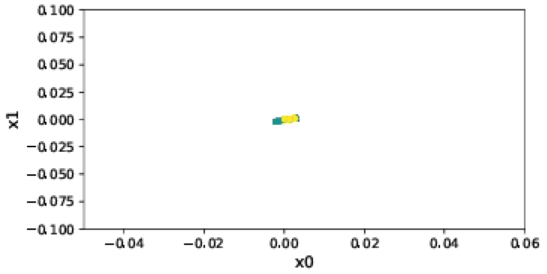
- **Linear GNN** *(also called SGC [Wu et al. 2019])*

$$\hat{Y} = Z^{(k)}\beta \text{ with } Z^{(k)} = L^k Z$$





- ***Semi-Supervised Learning*** $n_{tr}, n_{te} \sim n$



- ***Ridge Regression***

$$\beta^{(k)} = \arg\min_\beta \frac{1}{n_{tr}} \|Z^{(k)}_{tr}\beta - Y_{tr}\|^2 + \lambda\|\beta\|^2$$

- ***Test risk***

$$\mathcal{R}^{(k)} = \frac{1}{n_{te}} \|Y_{te} - Z^{(k)}_{te}\beta^{(k)}\|^2$$

Goal: show there is $k^\star$ s.t.

$$\mathcal{R}^{(k^\star)} < \min(\mathcal{R}^{(0)}, \mathcal{R}^{(\infty)})$$

Thm: Oversmoothing $Z^{(k)}_{te}\beta^{(k)} \xrightarrow[k\to\infty]{} C1_{n_{te}}$

Regression

Regression settings: $x \sim \mathcal{N}(0, \Sigma), \quad y = x^\top \beta^\star$

Thm: if $\Sigma, \beta^\star, M$ are "well-aligned" and n is large enough, $k^\star$ exists.

Intuition: $L^k X$ behaves "almost" as

$$\mathcal{N}(0, (\mathrm{Id} + \Sigma^{-1})^{-k}\Sigma)$$

- *The small eigenvalues shrink* ***faster*** *than the large ones* $\lambda_i \leftarrow \lambda_i/(1+1/\lambda_i)^k$
- *If well-aligned ("homophily"), smoothing helps*
- *If inversely aligned ("heterophily"), smoothing never helps*
- *Proof not that simple: for k>0,* ***dependent rows of Z***

Regression

Assume the latent variable $x \sim \mathcal{N}(0,\Sigma)$ and the label $y = x^T\beta^\star$.
We observe node features $z = M^T x \in \mathbb{R}^p$. Recall $x \in \mathbb{R}^d$ and $d \gg p$.

Example: $\mathscr{R}^{(\infty)}$

y follows a mean 0 normal distribution.

Recall that as $k \to \infty$, the predictions are $\hat{y}_{te}^{(k)} = c1_{n_{te}}$, where $c = c' \sum_{i=1}^{n_{tr}} y_i$.

In the infinite sample limit, as $n \to \infty$, we have $c \to \mathbb{E}[y] = 0$.
This means that, as $n, k \to \infty$, the predictions $\hat{y}_{te}^{(k)} \to 1_{n_{te}}\mathbb{E}[y] = 0$.
Consequently, as $n \to \infty$, $\mathscr{R}^{(\infty)} \to \mathbb{E}\,|y|^2 = \mathrm{Var}(y) = \beta^{\star T}\Sigma\beta^\star$.

What about $\mathscr{R}^{(0)}$ and more generally $\mathscr{R}^{(k)}$?

Regression

Given a psd matrix $S \in \mathbb{R}^{d\times d}$, define

$$R_{\mathrm{reg}}(S) = (\Sigma^{1/2}\beta^\star)^T\left(I - S^{1/2}M(\lambda I + M^TSM)^{-1}M^TS^{1/2}\right)^2(\Sigma^{1/2}\beta^\star) \in \mathbb{R}_+$$

[Theorem 3] $$\mathscr{R}^{(0)} = R_{\mathrm{reg}}(\Sigma) + \underbrace{\boxed{O\left(\frac{\mathrm{poly}(\|\Sigma\|, \|\beta^\star\|, d)}{\sqrt{n}}\right)}}_{\xrightarrow{n\to\infty} 0}$$

$$\Rightarrow \quad \mathscr{R}^{(0)} \approx R_{\mathrm{reg}}(\Sigma) \leq \mathrm{Var}(y) \approx \mathscr{R}^{(\infty)}$$

Regression

[Theorem 2] $\exists k^* \geq 1, \mathscr{R}^{(k^*)} < \min\{\mathscr{R}^{(0)}, \mathscr{R}^{(\infty)}\}$

Follows from Theorems 3 and 4 and **Assumption 1:** $R_{\text{reg}}(\Sigma) > R_{\text{reg}}(\Sigma^{(1)})$

[Theorem 3]

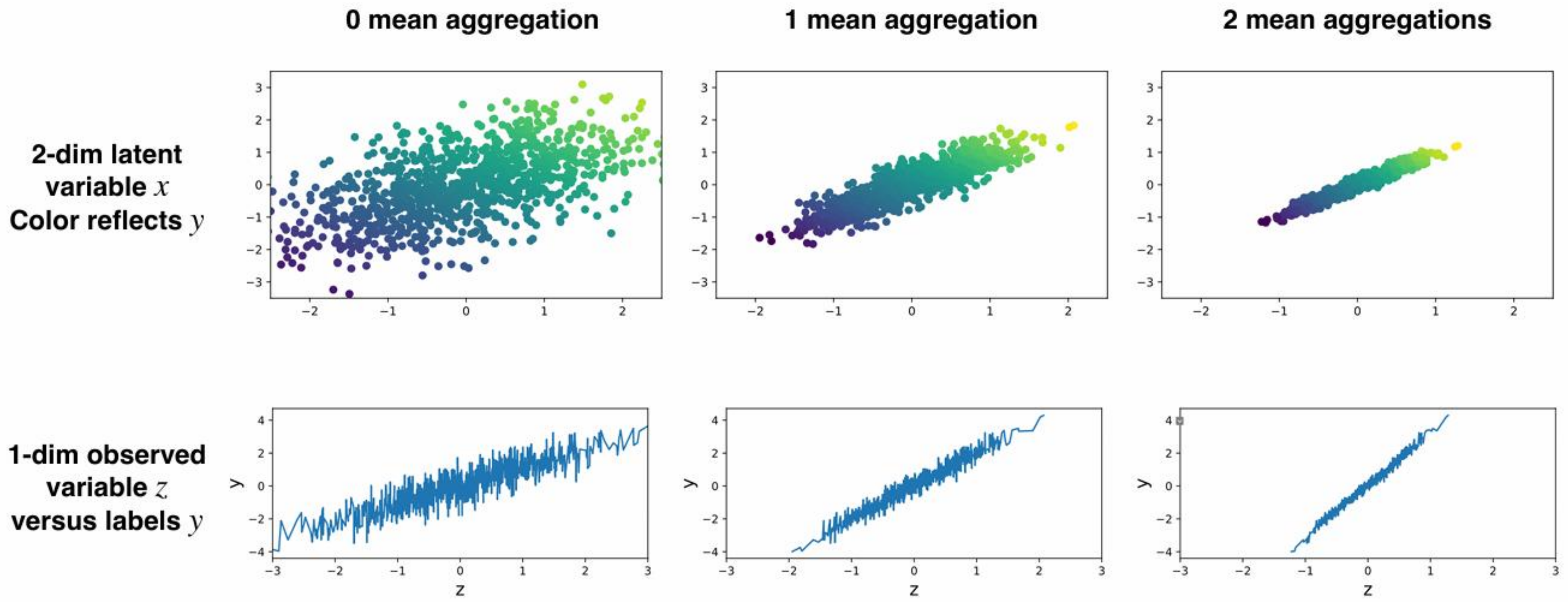
$$\mathscr{R}^{(0)} = R_{\text{reg}}(\Sigma) + \underbrace{\boxed{O\left(\frac{\text{poly}(\|\Sigma\|, \|\beta^\star\|, d)}{\sqrt{n}}\right)}}_{\xrightarrow{n \to \infty} 0}$$

[Theorem 4]

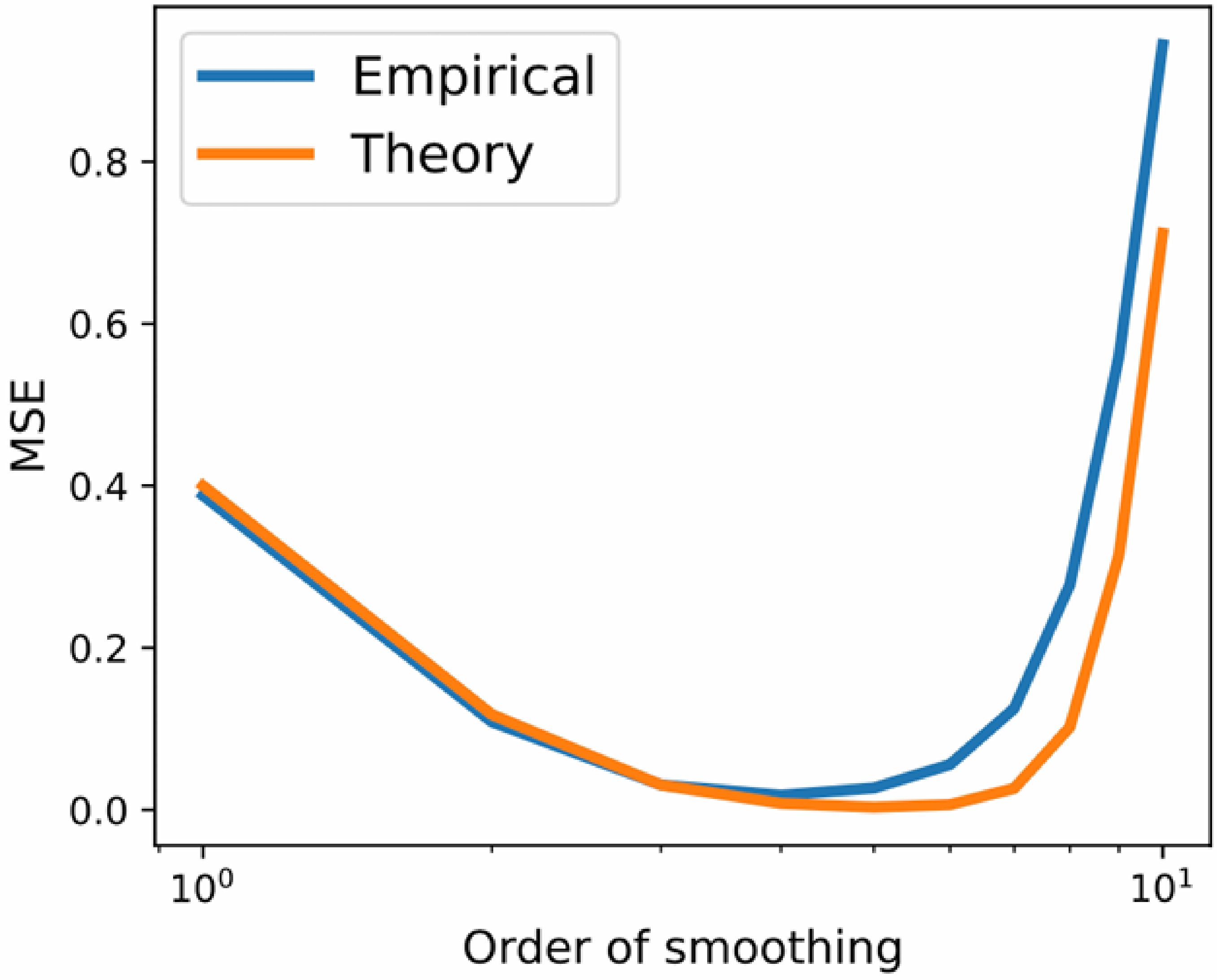
$$\mathscr{R}^{(1)} = R_{\text{reg}}(\boxed{\Sigma^{(1)}}) + \underbrace{\boxed{O\left(\frac{\text{poly}(\|\Sigma\|, \|\beta^\star\|, d, \epsilon^{-1})}{\sqrt{n}}\right) + O\left(\frac{C}{\epsilon^{1/5}}\right)}}_{\xrightarrow{n \to \infty} 0}$$

$$\boxed{\Sigma^{(k)} = (I + \Sigma^{-1})^{-2k}\Sigma}$$

Regression


0 mean aggregation
1 mean aggregation
2 mean aggregations
2-dim latent variable x Color reflects y
1-dim observed variable z versus labels y
y
z

Regression

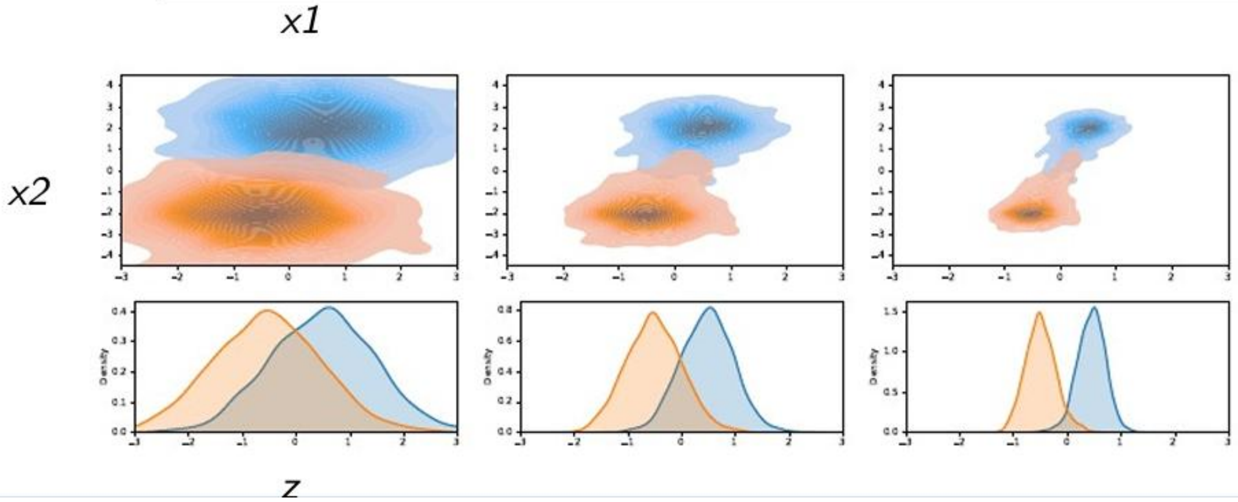

Empirical
Theory
MSE
0.0
0.2
0.4
0.6
0.8
10^0
10^1
Order of smoothing

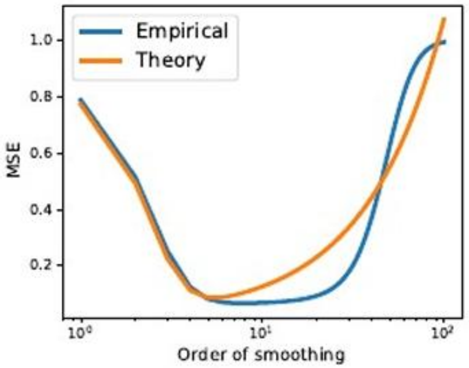
Classification

Classif. settings: $(x, y) \sim \frac{1}{2}\mathcal{N}(\mu, \mathrm{Id}) \otimes \{1\} + \frac{1}{2}\mathcal{N}(-\mu, \mathrm{Id}) \otimes \{-1\}$

Thm: if $\|\mu\|, n$ are large enough and $\|M\mu\| > 0$, $k^\star$ exists.

Intuition: *The communities (initially) concentrate faster than they get close to each other.*





- This can help separate node features z_i before oversmoothing happens

Conclusion

- **Beneficial smoothing and oversmoothing** generally coexist
- Theoretical analysis may give hints as to why
- There are links with the distinction homophily/heterophily

Outlooks:

- Can take inspiration to **combat oversmoothing less indiscriminatively**?
  - eg, smarter normalization in GNNs to fight oversmoothing
- How to better describe and exploit the interaction between labels, node features and graph structure?

Thank you!

Khushee Kapoor