Graph Neural Networks

Keyulu Xu

MIT

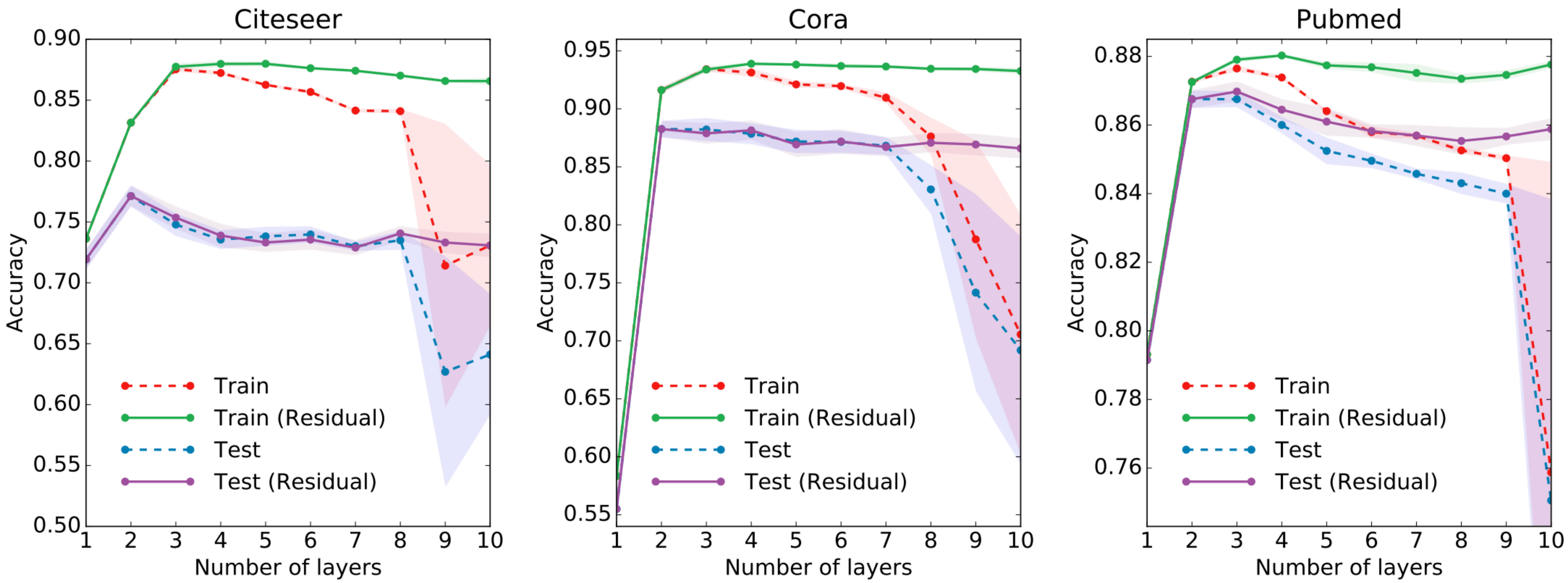
Outline

Representation learning on graphs with jumping knowledge networks. *ICML 2018*

Optimization of graph neural networks: implicit acceleration by skip connections and more depth. *ICML 2021*

GraphNorm: a principled apprach to accelerating graph neural networks. *ICML 2021*

Puzzle of the underperformance of deeper GNNs

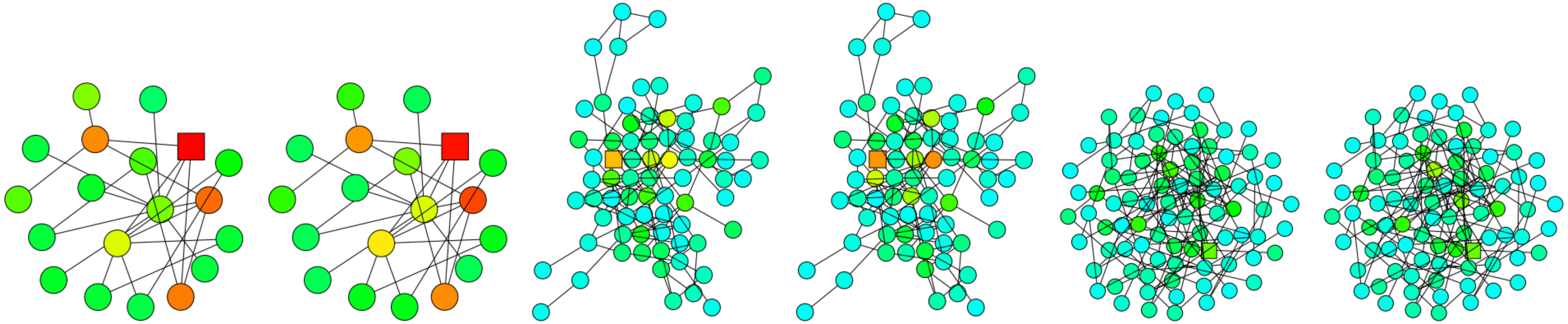


Influence function and graph structure

Theorem (XLTSKJ'18)
Influence distribution of learned node representations of a k-layer GCN is equal to that of a k-step random walk distribution.

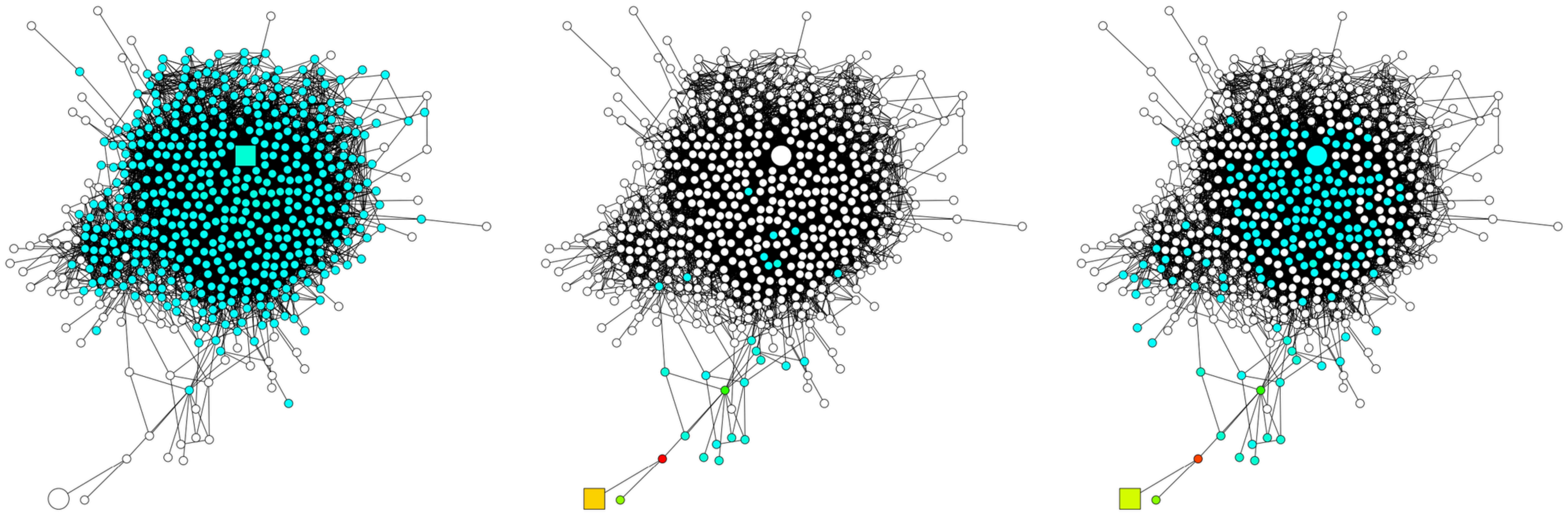
$$I(x,y) = e^T \left[\frac{\partial h_x^{(k)}}{\partial h_y^{(0)}} \right] e \quad I_x(y) = \frac{I(x,y)}{\sum_z I(x,z)}$$

* *Assumption: randomized activation or linearization*



(a) 2 layer GCN (b) 2 step r.w. (c) 4 layer GCN (d) 4 step r.w. (e) 6 layer GCN (f) 6 step r.w.

Optimal depth with respect to graph structure



(a) 4 steps at core (b) 4 steps at tree (c) 5 steps at tree

XLTSKJ'18

Optimal depth depends on the subgraph structure (expander vs. tree).

JK-Net: adaptively select the depth via skip connections.

Theory of GNNs

Optimization

Can gradient descent find a global minimum for GNNs?
What affects the speed of convergence?

Expressive Power

(Xu et al. 2019, Sato et al 2020, Chen et al 2019, 2020, Maron et al 2019, Keriven et al 2019, Loukas 2020, Balcilar et al 2021, Morris et al 2020, Azizian et al 2021, Vignac et al 2020)

Generalization

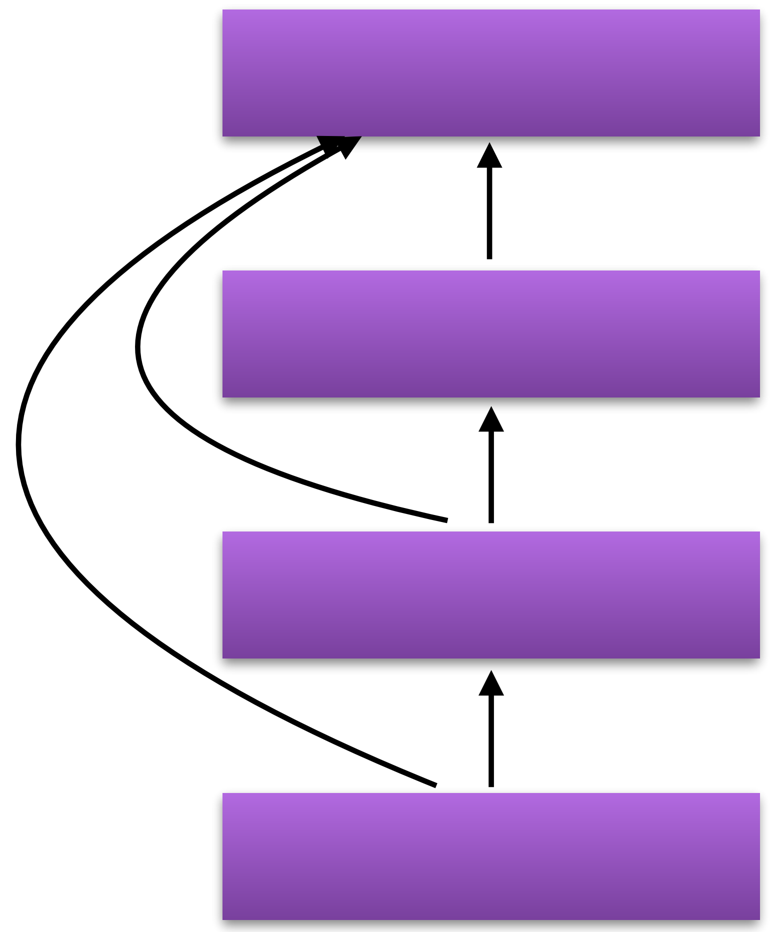
(interpolation & extrapolation)

(Scarselli et al. 2018, Verma et al 2019, Du et al 2019, Garg et al 2020, Xu et al 2020, 2021)

Analysis of gradient dynamics

Linearized GNNs with and without skip connections *(non-convex)*:

$$f(X, W, B) = \sum_{l=0}^{H} W_{(l)} X_{(l)},$$
$$X_{(l)} = B_{(l)} X_{(l-1)} S.$$



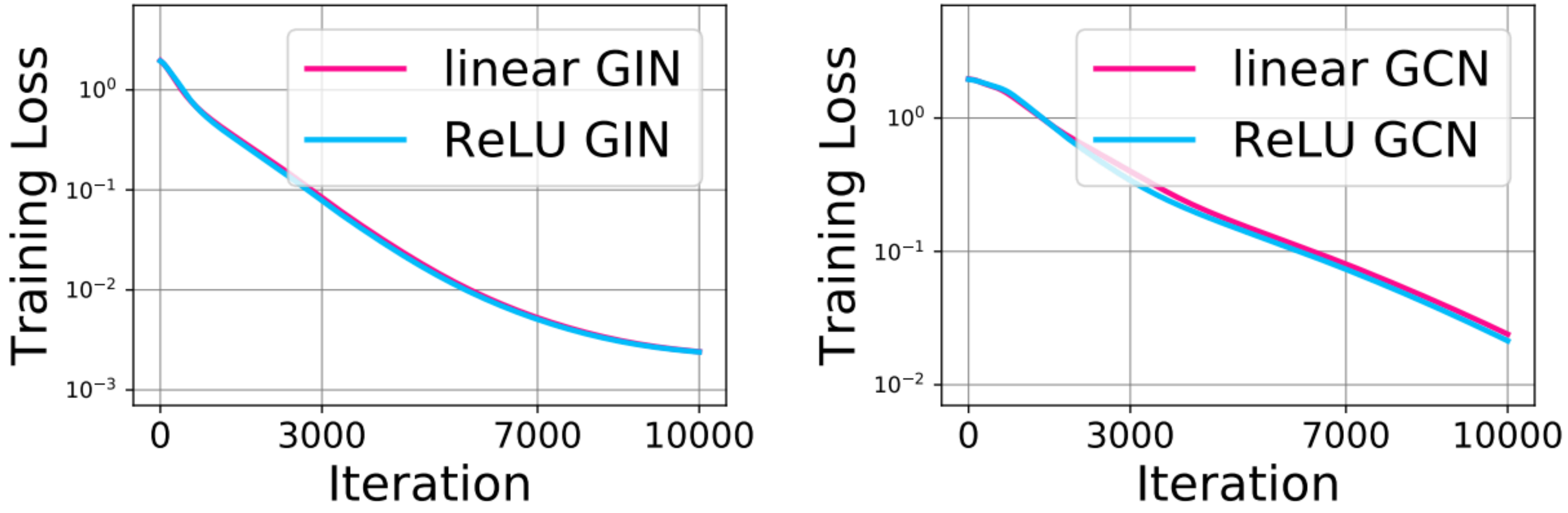
(Xu et al 2018)

Trajectory of gradient descent (flow) training:

$$\frac{d}{dt} W_t = -\frac{\partial L}{\partial W}(W_t, B_t), \quad \frac{d}{dt} B_t = -\frac{\partial L}{\partial B}(W_t, B_t)$$

Assumptions for analysis

Linear activation *(Saxe et al 2014, Kawaguchi 2016, Arora et al 2018, 2019, Bartlett et al 2019)*



Other common assumptions: over-parameterization (e.g. GNTK)

Difference: Convergence to global minimum with NTK assumes we can overfit the training data

(Jacot et al 2018, Li & Liang 2018, Du et al 2019, Arora et al 2019, Allen-Zhu et al 2019)

Global convergence

Theorem (XZJK'21)
Gradient descent training of a linearized GNN, with or without skip connections, converges to a *global minimum* at a linear rate.

$$L(W_T, B_T) - L^*_{1:H} \leq (L(W_0, B_0) - L^*_{1:H})e^{-\boxed{4\lambda_T^{(1:H)}}\boxed{\sigma^2_{\min}((G_H)_{*\mathcal{I}})T}}$$

global minimum

time-dependent on weight matrices

graph

Convergence rate

$$L(W_T, B_T) - L^*_{1:H} \le (L(W_0, B_0) - L^*_{1:H}) e^{-4\lambda_T^{(1:H)} \sigma^2_{\min}((G_H)_{*\mathcal{I}})T}$$

$$\lambda_T^{(H)} := \inf_{t\in[0,T]} \lambda_{\min}((\bar{B}_t^{(1:H)})^\top \bar{B}_t^{(1:H)})$$

$$B_{(l)} B_{(l-1)} \cdots B_{(1)}$$

GNN matrix

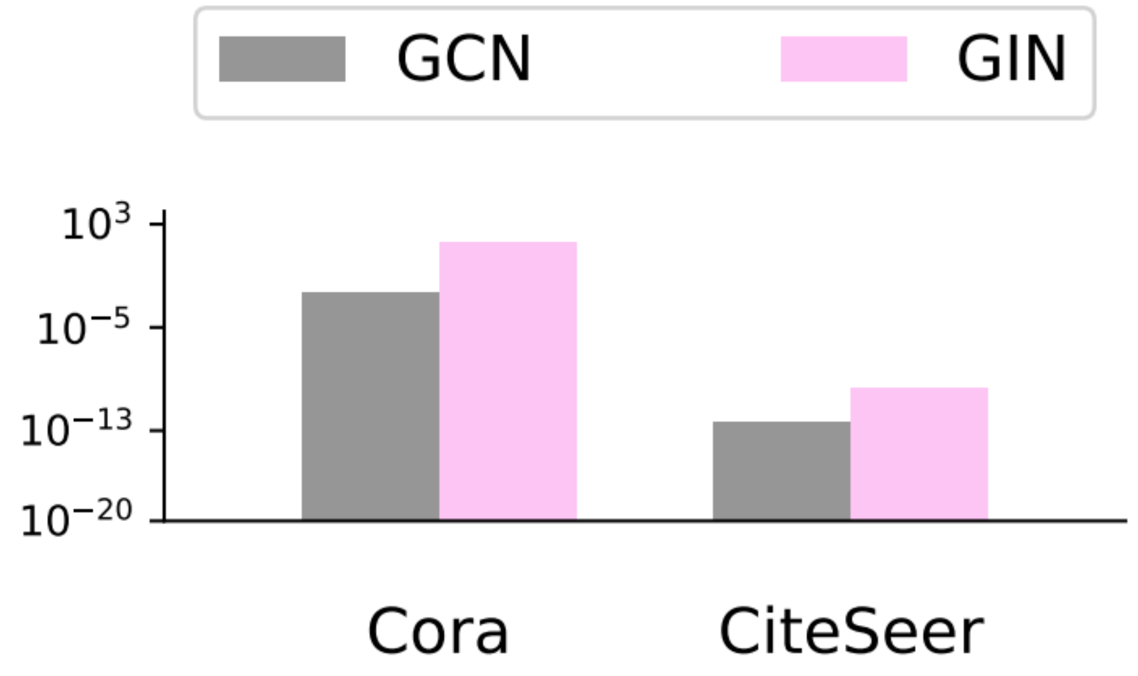
Without skip connections: $G_H := XS^H$

With skip connections: $G_H := [X^\top, (XS)^\top, \dots, (XS^H)^\top]^\top$

Conditions for global convergence

$$e^{-\boxed{4\lambda_T^{(1:H)}}\boxed{\sigma^2_{\min}((G_H)_{*\mathcal{I}})T}}$$

> 0?



(a) Graph $\sigma^2_{\min}(X(S^H)_{*\mathcal{I}})$

Lemma (XZJK'21)
Time-dependent condition is satisfied if initialization is good, i.e., loss is small at initialization.

What affects training speed

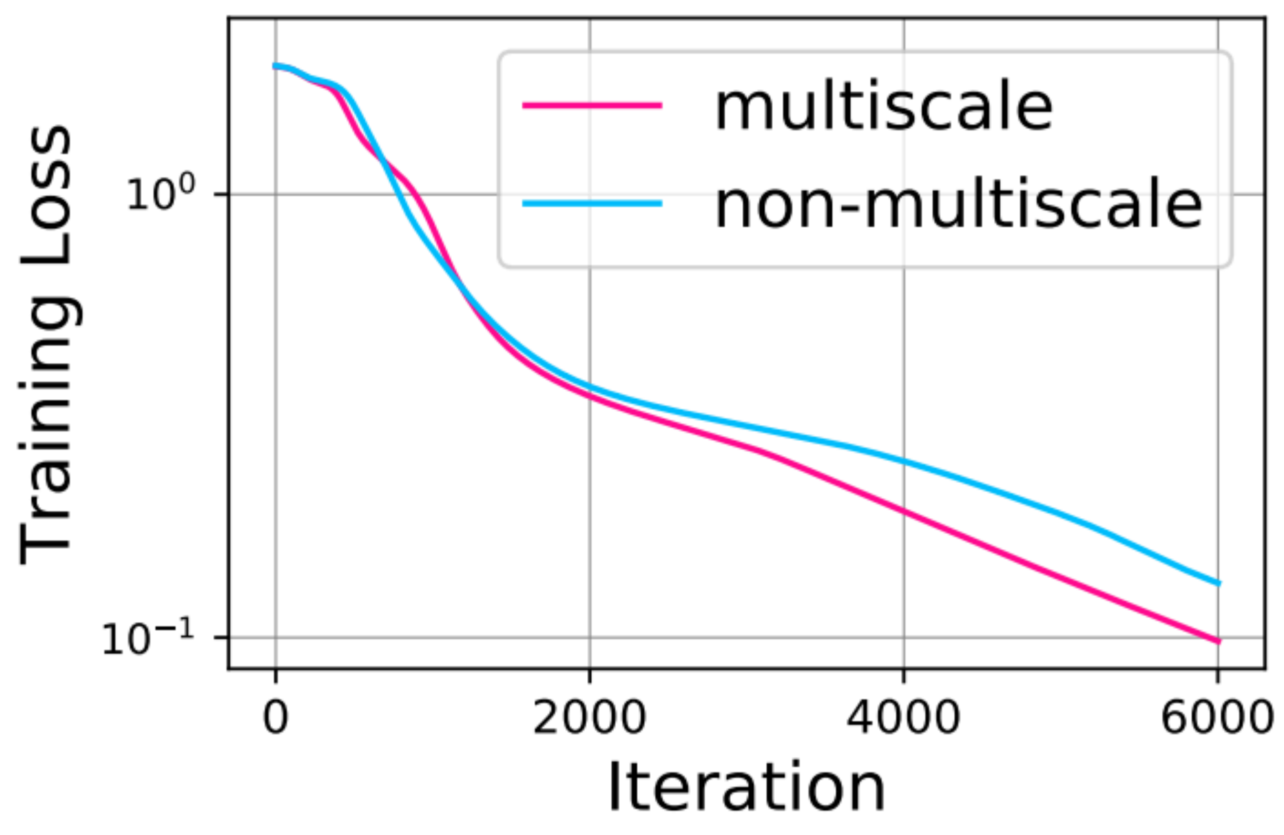
$$e^{-4\lambda_T^{(1:H)}\sigma^2_{\min}((G_H)_{*\mathcal{I}})T}$$

graph structure, feature normalization, initialization,

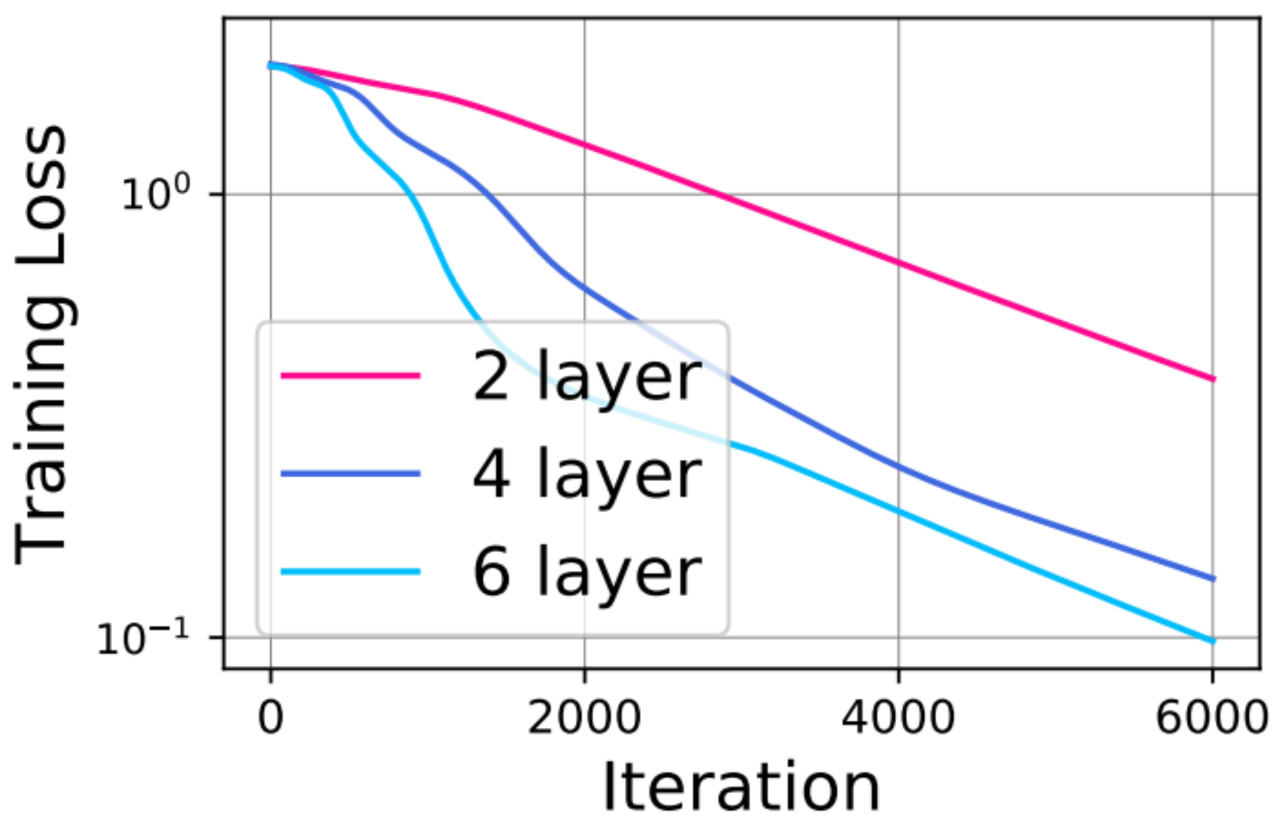
GNN architecture, labels…

Implicit acceleration

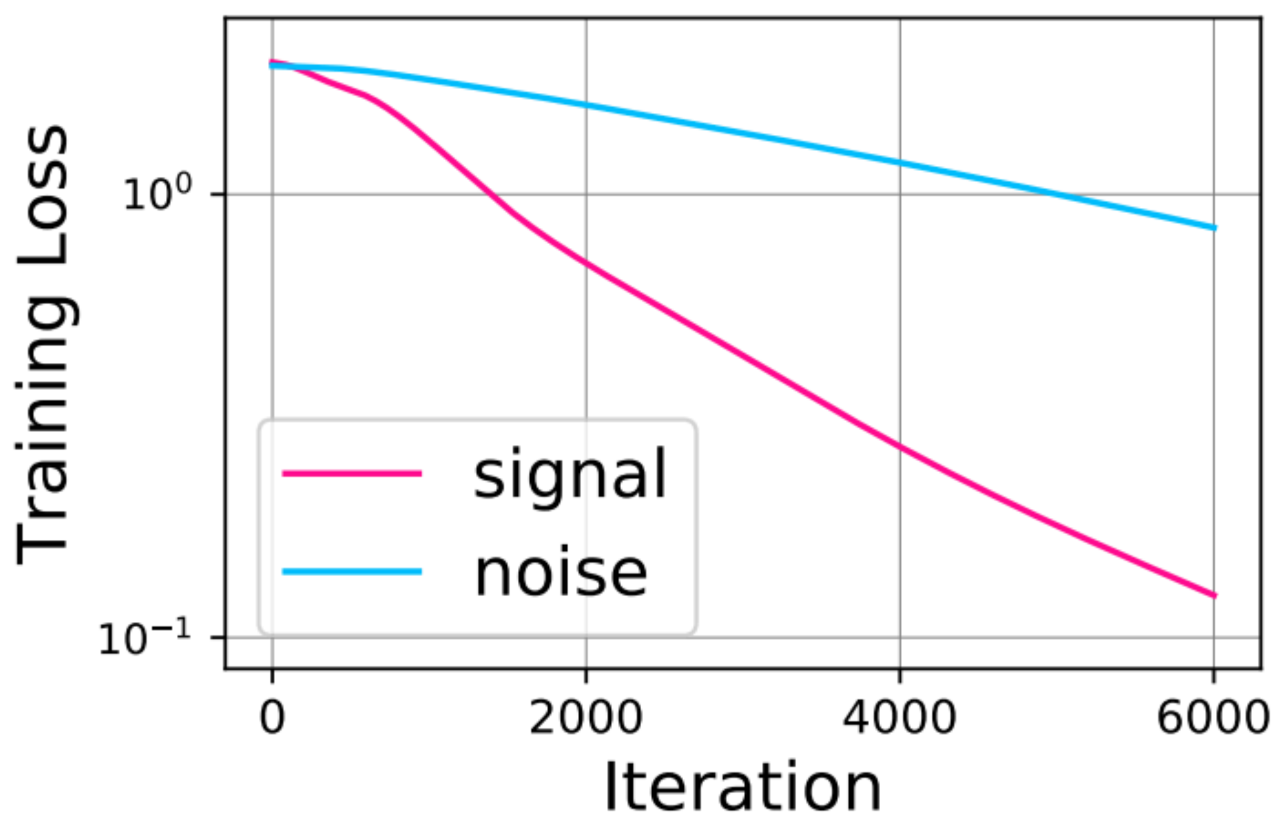
Theorem (XZJK'21)
GNN training is implicitly accelerated by skip connections, depth, and/or a good label distribution.



(a) Multiscale vs. non-multiscale.



(b) Depth.



(c) Signal vs. noise.

How depth and labels affect training speed

Deeper GNNs train faster:

$$\frac{d}{dt}L(W_t, B_t) = \underbrace{-\sum_{l=0}^{H} \underbrace{\left\| \mathrm{vec}\left[V_t (X(S^l)_{*\mathcal{I}})^\top\right]\right\|_{F_{(l),t}}^2}_{\geq 0}}_{\text{further improvement as depth } H \text{ increases}} \underbrace{-\sum_{i=1}^{H} \underbrace{\left\| \sum_{l=i}^{H} J_{(i,l),t} \,\mathrm{vec}\left[V_t (X(S^l)_{*\mathcal{I}})^\top\right]\right\|_2^2}_{\geq 0}}_{\text{further improvement as depth } H \text{ increases}}.$$

Faster if labels are more correlated with graph features:

$\left\| \mathrm{vec}\left[V_t (X(S^l)_{*\mathcal{I}})^\top\right]\right\|_{F_{(l),t}}^2$ larger if Y more correlated with $X(S^l)_{*\mathcal{I}}$

$$V_t := \frac{\partial L(W_t, B_t)}{\partial \hat{Y}_t}$$

Implication for deep GNNs

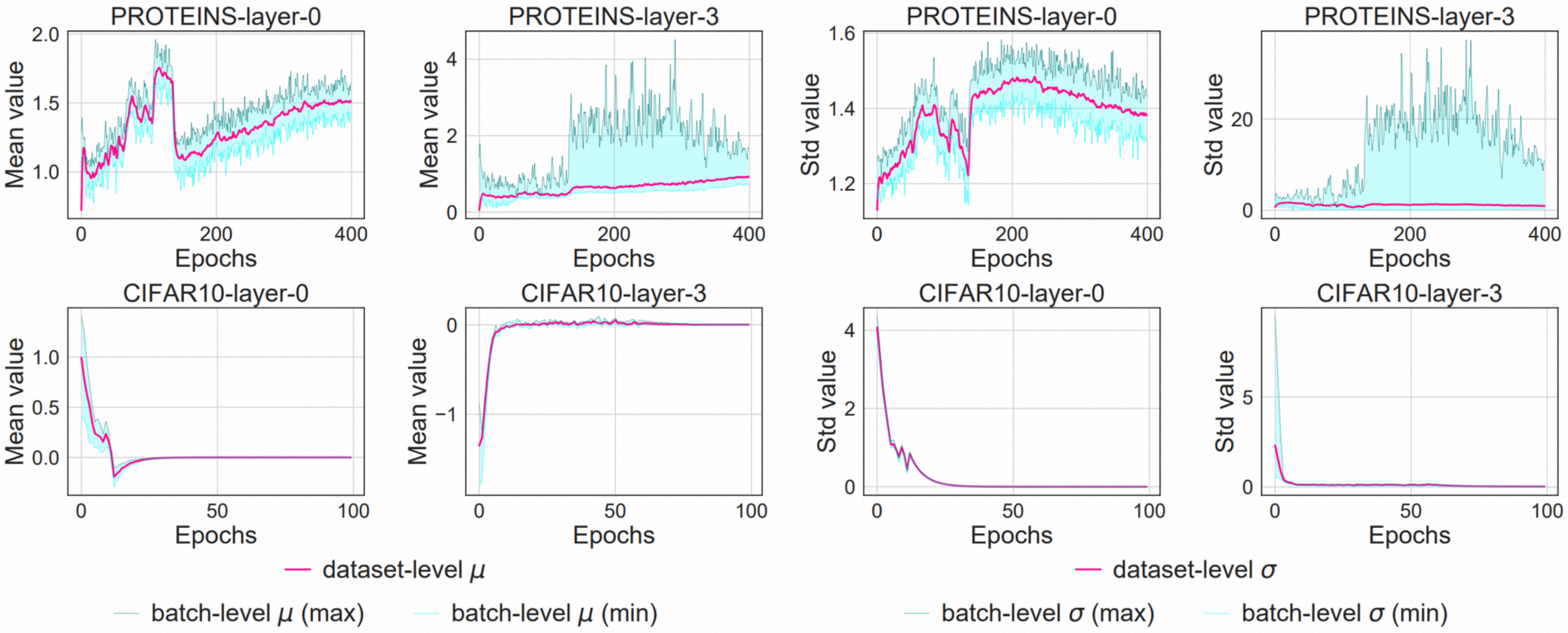
Node prediction over-smoothing: Deeper GNNs without skip connections may not have better global minimum

Deeper GNNs with skip connections is better in terms of optimization

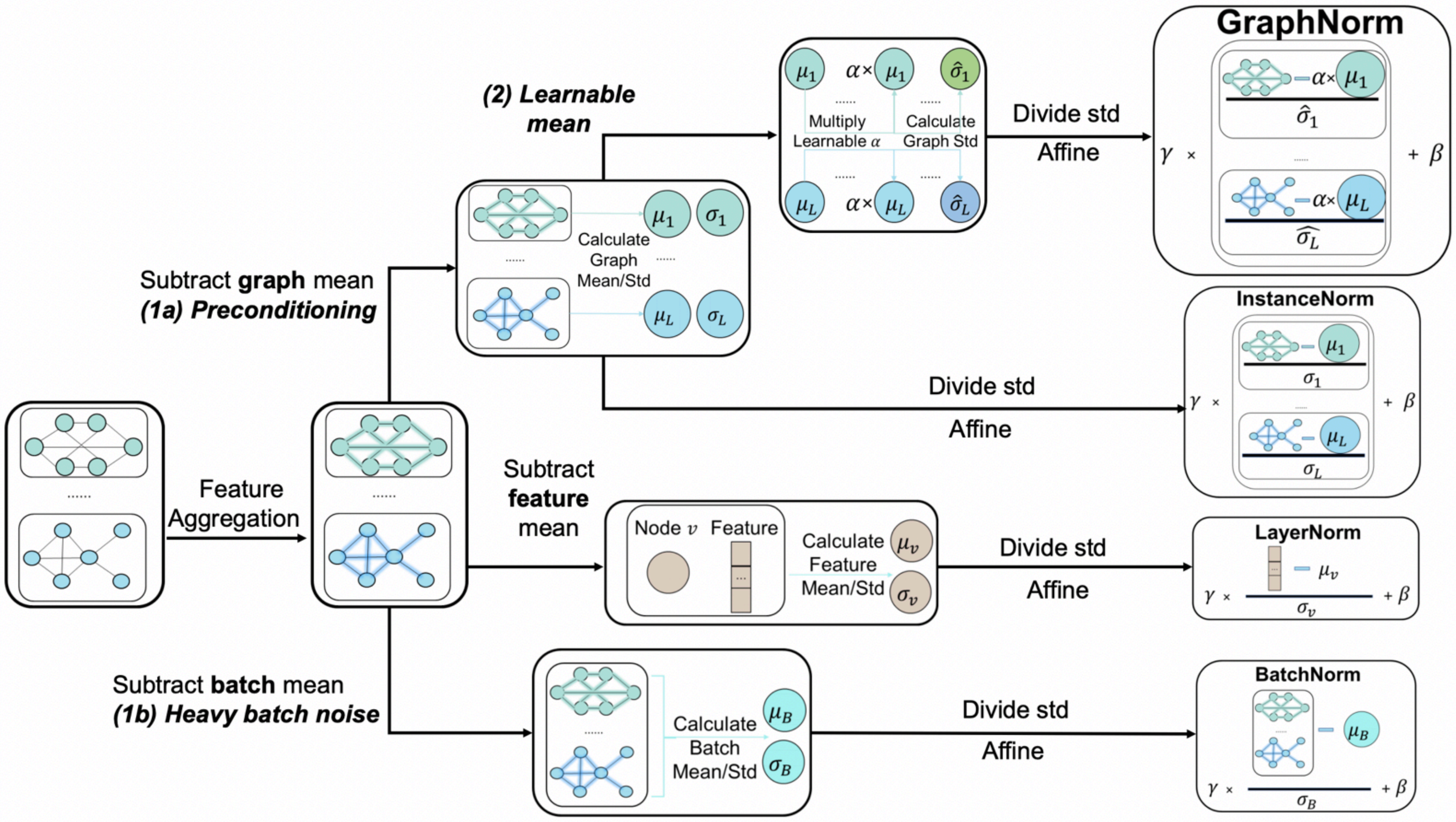
(1) Guaranteed to have smaller training loss

(2) Converge faster

Normalization: why BatchNorm is less effective



Normalization on graphs

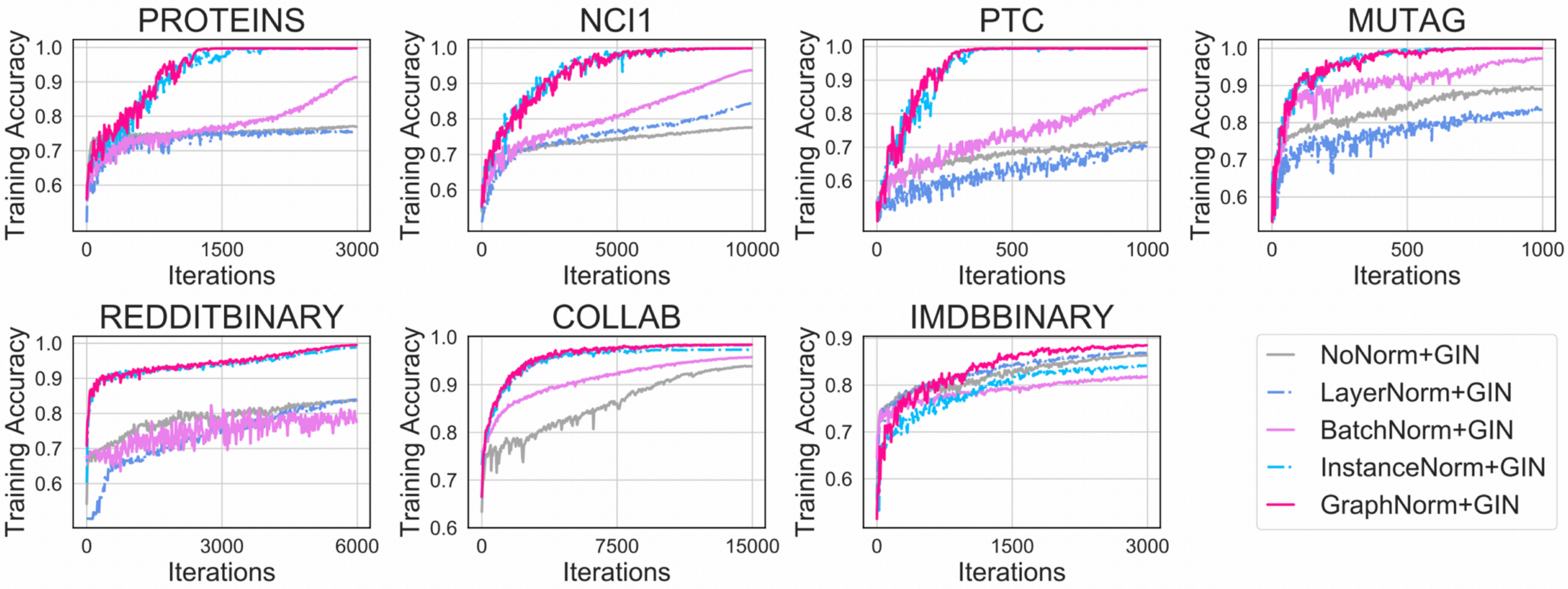


GraphNorm

$$\text{GraphNorm}\left(\hat{h}_{i,j}\right) = \gamma_j \cdot \frac{\hat{h}_{i,j} - \alpha_j \cdot \mu_j}{\hat{\sigma}_j} + \beta_j$$

$$\text{where } \mu_j = \frac{\sum_{i=1}^{n} \hat{h}_{i,j}}{n}, \hat{\sigma}_j^2 = \frac{\sum_{i=1}^{n} \left(\hat{h}_{i,j} - \alpha_j \cdot \mu_j\right)^2}{n}$$

GraphNorm accelerates training



GraphNorm improves generalization

Table 1. **Test performance** of GIN/GCN with various normalization methods on graph classification tasks.

Datasets	MUTAG	PTC	PROTEINS	NCI1	IMDB-B	RDT-B	COLLAB
# graphs	188	344	1113	4110	1000	2000	5000
# classes	2	2	2	2	2	2	2
Avg # nodes	17.9	25.5	39.1	29.8	19.8	429.6	74.5
WL SUBTREE (SHERVASHIDZE ET AL., 2011)	90.4 ± 5.7	59.9 ± 4.3	75.0 ± 3.1	$\mathbf{86.0 \pm 1.8}$	73.8 ± 3.9	81.0 ± 3.1	78.9 ± 1.9
DCNN (ATWOOD & TOWSLEY, 2016)	67.0	56.6	61.3	62.6	49.1	-	52.1
DGCNN (ZHANG ET AL., 2018)	85.8	58.6	75.5	74.4	70.0	-	73.7
AWL (IVANOV & BURNAEV, 2018)	87.9 ± 9.8	-	-	-	74.5 ± 5.9	87.9 ± 2.5	73.9 ± 1.9
GIN+LAYERNORM	82.4 ± 6.4	62.8 ± 9.3	76.2 ± 3.0	$78.3 \pm 1{,}7$	$74.5 \pm 4{,}4$	82.8 ± 7.7	80.1 ± 0.8
GIN+BATCHNORM ((XU ET AL., 2019))	89.4 ± 5.6	64.6 ± 7.0	76.2 ± 2.8	82.7 ± 1.7	75.1 ± 5.1	92.4 ± 2.5	$\mathbf{80.2 \pm 1.9}$
GIN+INSTANCENORM	90.5 ± 7.8	64.7 ± 5.9	76.5 ± 3.9	81.2 ± 1.8	74.8 ± 5.0	93.2 ± 1.7	80.0 ± 2.1
GIN+GraphNorm	$\mathbf{91.6 \pm 6.5}$	$\mathbf{64.9 \pm 7.5}$	$\mathbf{77.4 \pm 4.9}$	81.4 ± 2.4	$\mathbf{76.0 \pm 3.7}$	$\mathbf{93.5 \pm 2.1}$	$\mathbf{80.2 \pm 1.0}$