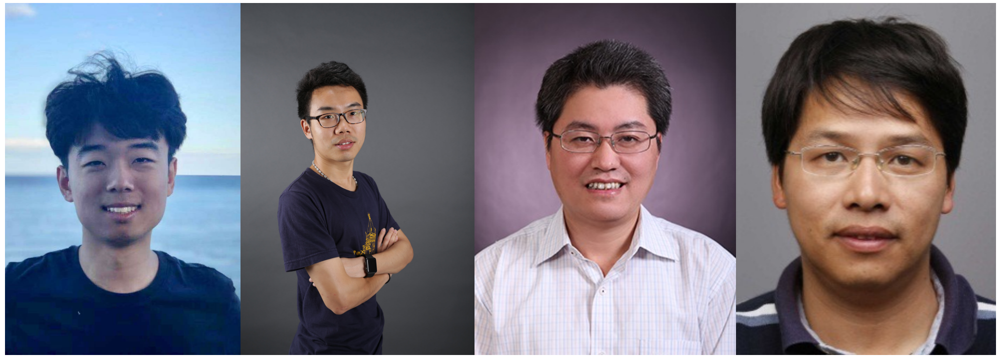
Dissecting the Diffusion Process in Linear Graph Convolutional Networks

Yifei Wang, Yisen Wang, Jiansheng Yang, Zhouchen Lin (Peking University)



NeurIPS 2021

https://yifeiwang77.github.io

Background

- Linear GCNs achieve comparable performance to nonlinear ones
- SGC (Simple Graph Convolution)
  - Given input **X**, label **Y**, adjacency matrix **A**, SGC predicts with

$$\hat{\mathbf{Y}}_{\mathrm{SGC}} = \mathrm{softmax}\left(\mathbf{S}^K \mathbf{X} \boldsymbol{\Theta}\right)$$

    - **1) K propagation steps (core)**

$$\mathbf{X}^{(k)} \leftarrow \mathbf{S}\mathbf{X}^{(k-1)}, \text{ where } \mathbf{S} = \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}} \quad \Longrightarrow \quad \mathbf{X}^{(K)} = \mathbf{S}^K \mathbf{X}$$

    - 2) linear classification

$$\hat{\mathbf{Y}}_{\mathrm{SGC}} = \mathrm{softmax}\left(\mathbf{X}^{(K)} \boldsymbol{\Theta}\right)$$

  - advantages: memory and parameter efficiency (preprocessed features)
  - disadvantages: over-smoothing, inferior performance

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.

Equivalence between SGC and Graph Heat Equation

- Key Insight from a continuous perspective
  - SGC's propagation = a (coarse) **discretization** of the graph diffusion equation
- Graph Heat Equation (GHE)

$$\begin{cases} \frac{d\mathbf{X}_t}{dt} & = -\mathbf{L}\mathbf{X}_t \\ \mathbf{X}_0 & = \mathbf{X} \end{cases}$$

  - where $\mathbf{L} = \mathbf{I} - \mathbf{S}$ is the graph Laplacian

Equivalence between SGC and Graph Heat Equation

- Key Insight from a continuous perspective
  - SGC's propagation = a (coarse) **discretization** of the graph diffusion equation
- Graph Heat Equation (GHE)

$$\begin{cases} \frac{d\mathbf{X}_t}{dt} & = -\mathbf{L}\mathbf{X}_t \\ \mathbf{X}_0 & = \mathbf{X} \end{cases}$$

  - where $\mathbf{L} = \mathbf{I} - \mathbf{S}$ is the graph Laplacian
- Discretization
  - Applying the forward Euler method with time interval Δt

Euler: $$\hat{\mathbf{X}}_{t+\Delta t} = \hat{\mathbf{X}}_t - \Delta t\mathbf{L}\hat{\mathbf{X}}_t = \hat{\mathbf{X}}_t - \Delta t(\mathbf{I}-\mathbf{S})\hat{\mathbf{X}}_t = [(1-\Delta t)\mathbf{I} + \Delta t\mathbf{S}]\hat{\mathbf{X}}_t$$

SGC propagation:

$$\mathbf{X}^{(k)} \leftarrow \mathbf{S}\mathbf{X}^{(k-1)}$$

  - Thus, SGC is the Euler discretization of GHE with step size $\Delta t = 1$

Revealing SGC's Fundamental Limitations

- Limitations
  - 1. Oversmoothing (asymptotic)
    - SGC will oversmooth with increasing propagation steps $K = T \to \infty$
    - We provide a continuous characterization of this phenomenon

Theorem 1 (Oversmoothing from a spectral view). *Assume that the eigendecomposition of the Laplacian matrix as* $\mathbf{L} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$*, with eigenvalues* λ_i *and eigenvectors* $\mathbf{u}_i$*. Then, the heat equation (Eq. (4)) admits a closed-form solution at time* t*, known as the heat kernel* $\mathbf{H}_t = e^{-t\mathbf{L}} = \sum_{i=1}^{n} e^{-\lambda_i t} \mathbf{u}_i \mathbf{u}_i^\top$*. As* $t \to \infty$*,* $\mathbf{H}_t$ *asymptotically converges to a non-informative equilibrium as* $t \to \infty$*, due to the non-trivial (i.e., positive) eigenvalues vanishing:*

$$\lim_{t\to\infty} e^{-\lambda_i t} = \begin{cases} 0, & \text{if } \lambda_i > 0 \\ 1, & \text{if } \lambda_i = 0 \end{cases}, \quad i = 1, \dots, n. \tag{7}$$

Revealing SGC's Fundamental Limitations

- Limitations
  - 1. Oversmoothing (asymptotic)
  - 2. Numerical Error
    - Consequence by adopting a fixed time interval $\Delta t = 1$

Theorem 2 (Numerical errors). *For the initial value problem in Eq. (4) with finite terminal time T, the numerical error of the forward Euler method in Eq. (5) with K steps can be upper bounded by*

$$\left\|\mathbf{e}_T^{(K)}\right\| \leq \frac{T\|\mathbf{L}\|\|\mathbf{X}_0\|}{2K}\left(e^{T\|\mathbf{L}\|}-1\right). \quad (8)$$

- As T=K, the upper bound reduces to $c \cdot \left(e^{T\|\mathbf{L}\|}-1\right)$
- The numerical error increases exponentially with more propagation steps K=T

Revealing SGC's Fundamental Limitations

- Limitations
  - 1. Oversmoothing (asymptotic)
  - 2. Numerical Error
  - 3. Learning Risks
    - The two above issues will finally lead to a large learning risk

Theorem 3 (Learning risks). *Consider a simple linear regression problem* $(\mathbf{X}, \mathbf{Y})$ *on graph, where the observed input features* $\mathbf{X}$ *are generated by corrupting the ground truth features* $\mathbf{X}_c$ *with the following inverse graph diffusion with time* T^* :

$$\frac{d\widetilde{\mathbf{X}}_t}{dt} = \mathbf{L}\widetilde{\mathbf{X}}_t, \quad \text{where } \widetilde{\mathbf{X}}_0 = \mathbf{X}_c \text{ and } \widetilde{\mathbf{X}}_{T^*} = \mathbf{X}. \tag{9}$$

Denote the population risk with ground truth features as $R(\mathbf{W}) = \mathbb{E}\left\|\mathbf{Y} - \mathbf{X}_c\mathbf{W}\right\|^2$ *and that of Euler method applied input* $\mathbf{X}$ *(Eq. (5)) as* $\hat{R}(\mathbf{W}) = \mathbb{E}\left\|\mathbf{Y} - \left[\mathbf{S}^{(\Delta t)}\right]^K \mathbf{X}\mathbf{W}\right\|^2$. *Supposing that* $\mathbb{E}\|\mathbf{X}_c\|^2 = M < \infty$*, we have the following upper bound:*

$$\hat{R}(\mathbf{W}) \leq R(\mathbf{W}) + \|\mathbf{W}\|^2 \left(M \left\| e^{T^*\mathbf{L}} \right\|^2 \left\| e^{-T^*\mathbf{L}} - e^{-\hat{T}\mathbf{L}} \right\|^2 + \mathbb{E} \left\| \mathbf{e}_{T^*}^{(K)} \right\|^2 \right). \tag{10}$$

To minimize the risk, we need

1) the optimal terminal time
2) minimized numerical errors

Revealing SGC's Fundamental Limitations

- Limitations
  - 1. Oversmoothing (asymptotic)
  - 2. Numerical Error
  - 3. Learning Risks
    - The two above issues will finally lead to a large learning risk

Theorem 3 (Learning risks). *Consider a simple linear regression problem* $(\mathbf{X}, \mathbf{Y})$ *on graph, where the observed input features* $\mathbf{X}$ *are generated by corrupting the ground truth features* $\mathbf{X}_c$ *with the following inverse graph diffusion with time* T^* :

$$\frac{d\widetilde{\mathbf{X}}_t}{dt} = \mathbf{L}\widetilde{\mathbf{X}}_t, \quad \text{where } \widetilde{\mathbf{X}}_0 = \mathbf{X}_c \text{ and } \widetilde{\mathbf{X}}_{T^*} = \mathbf{X}. \tag{9}$$

Denote the population risk with ground truth features as $R(\mathbf{W}) = \mathbb{E}\,\|\mathbf{Y} - \mathbf{X}_c\mathbf{W}\|^2$ *and that of Euler method applied input* $\mathbf{X}$ *(Eq. (5)) as* $\hat{R}(\mathbf{W}) = \mathbb{E}\left\|\mathbf{Y} - \left[\mathbf{S}^{(\Delta t)}\right]^K \mathbf{X}\mathbf{W}\right\|^2$. *Supposing that* $\mathbb{E}\|\mathbf{X}_c\|^2 = M < \infty$, *we have the following upper bound:*

$$\hat{R}(\mathbf{W}) \leq R(\mathbf{W}) + \|\mathbf{W}\|^2 \left(M \left\| e^{T^*\mathbf{L}} \right\|^2 \left\| e^{-T^*\mathbf{L}} - e^{-\hat{T}\mathbf{L}} \right\|^2 + \mathbb{E}\left\| \mathbf{e}^{(K)}_{T^*} \right\|^2 \right). \tag{10}$$

To minimize the risk, we need
1) the optimal terminal time ✗
2) minimized numerical errors

Ideal: real-value
SGC: integer

Revealing SGC's Fundamental Limitations

- Limitations
  - 1. Oversmoothing (asymptotic)
  - 2. Numerical Error
  - 3. Learning Risks
    - The two above issues will finally lead to a large learning risk

Theorem 3 (Learning risks). *Consider a simple linear regression problem* $(\mathbf{X}, \mathbf{Y})$ *on graph, where the observed input features* $\mathbf{X}$ *are generated by corrupting the ground truth features* $\mathbf{X}_c$ *with the following inverse graph diffusion with time* T^* :

$$\frac{d\widetilde{\mathbf{X}}_t}{dt} = \mathbf{L}\widetilde{\mathbf{X}}_t, \quad \text{where } \widetilde{\mathbf{X}}_0 = \mathbf{X}_c \text{ and } \widetilde{\mathbf{X}}_{T^*} = \mathbf{X}. \tag{9}$$

Denote the population risk with ground truth features as $R(\mathbf{W}) = \mathbb{E}\left\|\mathbf{Y} - \mathbf{X}_c\mathbf{W}\right\|^2$ *and that of Euler method applied input* $\mathbf{X}$ *(Eq. (5)) as* $\hat{R}(\mathbf{W}) = \mathbb{E}\left\|\mathbf{Y} - \left[\mathbf{S}^{(\Delta t)}\right]^K \mathbf{X}\mathbf{W}\right\|^2$. *Supposing that* $\mathbb{E}\|\mathbf{X}_c\|^2 = M < \infty$, *we have the following upper bound:*

$$\hat{R}(\mathbf{W}) \leq R(\mathbf{W}) + \|\mathbf{W}\|^2 \left(M \left\|e^{T^*\mathbf{L}}\right\|^2 \left\|e^{-T^*\mathbf{L}} - e^{-\hat{T}\mathbf{L}}\right\|^2 + \mathbb{E}\left\|\mathbf{e}_{T^*}^{(K)}\right\|^2\right). \tag{10}$$

To minimize the risk, we need
1) the optimal terminal time ✗
2) minimized numerical errors ✗

Ideal: $\boldsymbol{\Delta t \to 0}$
SGC: fixed step size $\Delta t = 1$

A Simple Fix to All These Limitations!

Decoupling T (terminal time) and **K** (propagation steps)

- We take K and T as two free parameters
  - 1. Flexibly choose T (real-valued) for an optimal tradeoff of smoothing
  - 2. Given a **fixed** optimal T*, we can increase K for better precision without oversmoothing

A Simple Fix to All These Limitations!

Decoupling T (terminal time) and **K** (propagation steps)

- We take K and T as two free parameters
  - 1. Flexibly choose T (real-valued) for an optimal tradeoff of smoothing
  - 2. Given a **fixed** optimal T*, we can increase K for better precision without oversmoothing
- Decoupled Graph Convolution (DGC)

$$\hat{\mathbf{Y}}_{\mathrm{DGC}} = \mathrm{softmax}\left(\hat{\mathbf{X}}_T \boldsymbol{\Theta}\right), \text{ where } \hat{\mathbf{X}}_T = \mathrm{ode}_{\mathrm{int}}(\mathbf{X}, \Delta t, K)$$

  - where $\mathrm{ode\_int}(\mathrm{X}, \Delta\mathrm{t}, \mathrm{K})$ denotes the numerical intergration with step size Δt for K steps

A Simple Fix to All These Limitations!

Decoupling T (terminal time) and **K** (propagation steps)

- We take K and T as two free parameters
  - 1. Flexibly choose T (real-valued) for an optimal tradeoff of smoothing
  - 2. Given a **fixed** optimal T*, we can increase K for better precision without oversmoothing
- Decoupled Graph Convolution (DGC)

$$\hat{\mathbf{Y}}_{\mathrm{DGC}} = \operatorname{softmax}\left(\hat{\mathbf{X}}_T \boldsymbol{\Theta}\right), \text{ where } \hat{\mathbf{X}}_T = \mathrm{ode}_{\mathrm{int}}(\mathbf{X}, \Delta t, K)$$

  - where $\mathrm{ode\_int}(\mathrm{X}, \Delta \mathrm{t}, \mathrm{K})$ denotes the numerical intergration with step size Δt for K steps
- DGC-Euler with forward Euler scheme and step size $\Delta \mathrm{t} = \mathrm{T}/\mathrm{K}$

$$\hat{\mathbf{X}}_T = \left[\mathbf{S}^{(T/K)}\right]^K \mathbf{X}, \text{ where } \mathbf{S}^{(T/K)} = (1 - T/K) \cdot \mathbf{I} + (T/K) \cdot \mathbf{S}$$

A Simple Fix to All These Limitations!

Decoupling T (terminal time) and **K** (propagation steps)

- We take K and T as two free parameters
  - 1. Flexibly choose T (real-valued) for an optimal tradeoff of smoothing
  - 2. Given a **fixed** optimal T*, we can increase K for better precision without oversmoothing
- Decoupled Graph Convolution (DGC)

$$\hat{\mathbf{Y}}_{\mathrm{DGC}} = \operatorname{softmax}\left(\hat{\mathbf{X}}_T \boldsymbol{\Theta}\right), \text{ where } \hat{\mathbf{X}}_T = \mathrm{ode}_{\mathrm{int}}(\mathbf{X}, \Delta t, K)$$

  - where $\mathrm{ode\_int}(\mathrm{X}, \Delta\mathrm{t}, \mathrm{K})$ denotes the numerical intergration with step size Δt for K steps
- DGC-Euler with forward Euler scheme and step size $\Delta \mathrm{t} = \mathrm{T}/\mathrm{K}$
- DGC-RK with the 4th-order Runge-Kutta (RK) method $\hat{\mathbf{X}}_{t+\Delta t} = \hat{\mathbf{X}}_t + \frac{1}{6}\Delta t(\mathbf{R}_1 + 2\mathbf{R}_2 + 2\mathbf{R}_3 + \mathbf{R}_4) \triangleq \mathbf{S}_{\mathrm{RK}}^{(\Delta t)} \hat{\mathbf{X}}_t$

Verifying the Benefits of DGC

- Theoretical Benefits
  - Comparing SGC to DGC

Aspects	SGC (Wu et al. 2019)	DGC-Euler (ours)
Asymptotic	Over-smoothing as T=K	**A fixed T with optimal tradeoff**
Numerical error	Exponentially large when K increases	**With fixed T, increasing K leads to smaller numerical error**
Learning Risk	Deviation from optimal T + Large numerical error	**Reach optimal real-valued T + minimized numerical error with large K**

Verifying the Benefits of DGC

- Theoretical Benefits
- Empirical Evidence

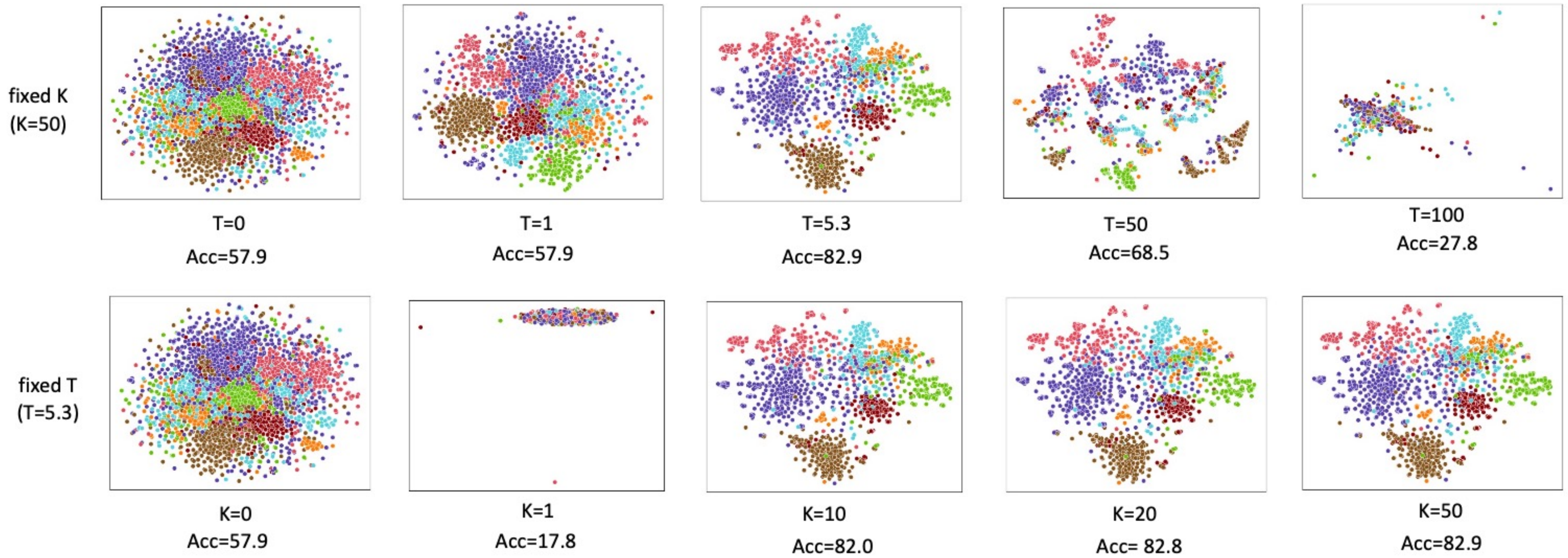


Figure 1: Input feature visualization of our DGC-Euler model with t-SNE [19] on the Cora dataset. Each point represents a node in the graph and its color denotes the class of the node.

T: Either a smaller T or a larger T mixes the features up. **An optimal T** implies better separable features.

K: With fixed optimal T , too large step size Δt (small K) leads to feature collapse, and **large K** makes features separable!

Experiments

- Performance on Semi-supervised Node Classification

Table 2: Test accuracy (%) of semi-supervised node classification on citation networks.

Type	Method	Cora	Citeseer	Pubmed
Non-linear	GCN [8]	81.5	70.3	79.0
	GAT [20]	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3
	GraphSAGE [6]	82.2	71.4	75.8
	JKNet [26]	81.1	69.8	78.1
	APPNP [9]	83.3	71.8	80.1
	GWWN [25]	82.8	71.7	79.1
	GraphHeat [24]	83.7	72.5	80.5
	CGNN [23]	84.2 ± 0.6	71.8 ± 0.7	76.8 ± 0.6
	GCDE [16]	83.8 ± 0.5	72.5 ± 0.5	79.9 ± 0.3
Linear	Label Propagation [29]	45.3	68.0	63.0
	DeepWalk [15]	70.7 ± 0.6	51.4 ± 0.5	76.8 ± 0.6
	SGC [22]	81.0 ± 0.0	71.9 ± 0.1	78.9 ± 0.0
	SGC-PairNorm [28]	81.1	70.6	78.2
	DGC (ours)	$\mathbf{83.5} \pm \mathbf{0.0}$	$\mathbf{74.5} \pm \mathbf{0.2}$	$\mathbf{80.2} \pm \mathbf{0.1}$

Experiments

- Performance on Semi-supervised Node Classification

Table 2: Test accuracy (%) of semi-supervised node classification on citation networks.

Type	Method	Cora	Citeseer	Pubmed
Non-linear	GCN [8]	81.5	70.3	79.0
	GAT [20]	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3
	GraphSAGE [6]	82.2	71.4	75.8
	JKNet [26]	81.1	69.8	78.1
	APPNP [9]	83.3	71.8	80.1
	GWWN [25]	[illegible]	71.7	79.1
	[illegible]	[illegible]	72.5	80.5
	[illegible]	[illegible]	71.8 ± 0.7	76.8 ± 0.6
	[illegible]	[illegible]	72.5 ± 0.5	79.9 ± 0.3
Linear	[illegible] [29]	45.3	68.0	63.0
	DeepWalk [15]	70.7 ± 0.6	51.4 ± 0.5	76.8 ± 0.6
	SGC [22]	81.0 ± 0.0	71.9 ± 0.1	78.9 ± 0.0
	SGC-PairNorm [28]	81.1	70.6	78.2
	DGC (ours)	**83.5** ± **0.0**	**74.5** ± **0.2**	**80.2** ± **0.1**

Improves SGC by a large margin

Experiments

- Performance on Semi-supervised No[illegible]

Comparable to SOTA nonlinear GCNs!

Table 2: Test accuracy (%) of [illegible] classification on citation networks.

Type	Method	Cora	Citeseer	Pubmed
Non-linear	GCN [8]	81.5	70.3	79.0
	GAT [20]	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3
	GraphSAGE [6]	82.2	71.4	75.8
	JKNet [26]	81.1	69.8	78.1
	APPNP [9]	83.3	71.8	80.1
	GWWN [25]	[illegible]	71.7	79.1
	[illegible]	[illegible]	72.5	80.5
	[illegible]	[illegible]	71.8 ± 0.7	76.8 ± 0.6
	[illegible]	[illegible]	72.5 ± 0.5	79.9 ± 0.3
Linear	[illegible] [29]	45.3	68.0	63.0
	DeepWalk [15]	70.7 ± 0.6	51.4 ± 0.5	76.8 ± 0.6
	SGC [22]	81.0 ± 0.0	71.9 ± 0.1	78.9 ± 0.0
	SGC-PairNorm [28]	81.1	70.6	78.2
	DGC (ours)	**83.5** ± **0.0**	**74.5** ± **0.2**	**80.2** ± **0.1**

Improves SGC by a large margin

Experiments

- Performance on Semi-supervised Node Classification
- Performance on Fully-supervised Node Classification

Table 3: Test accuracy (%) of fully-supervised node classification on citation networks.

Type	Method	Cora	Citeseer	Pubmed
Non-linear	GCN [8]	85.77	73.58	88.13
	GAT [20]	86.37	74.32	87.62
	JK-MaxPool [26]	89.6	77.7	-
	JK-Concat [26]	89.1	78.3	-
	JK-LSTM [26]	85.8	74.7	-
	APPNP [9]	90.21	79.8	86.29
Linear	SGC [22]	85.82	78.08	83.27
	DGC (ours)	**88.2 ± 0.1**	**79.0 ± 0.2**	**88.7 ± 0.0**

Experiments

- Performance on Semi-supervised Node Classification
- Performance on Fully-supervised Node Classification
- Performance on Large Scale Datasets

Table 3: Test accuracy (%) of fully-supervised node classification on citation networks.

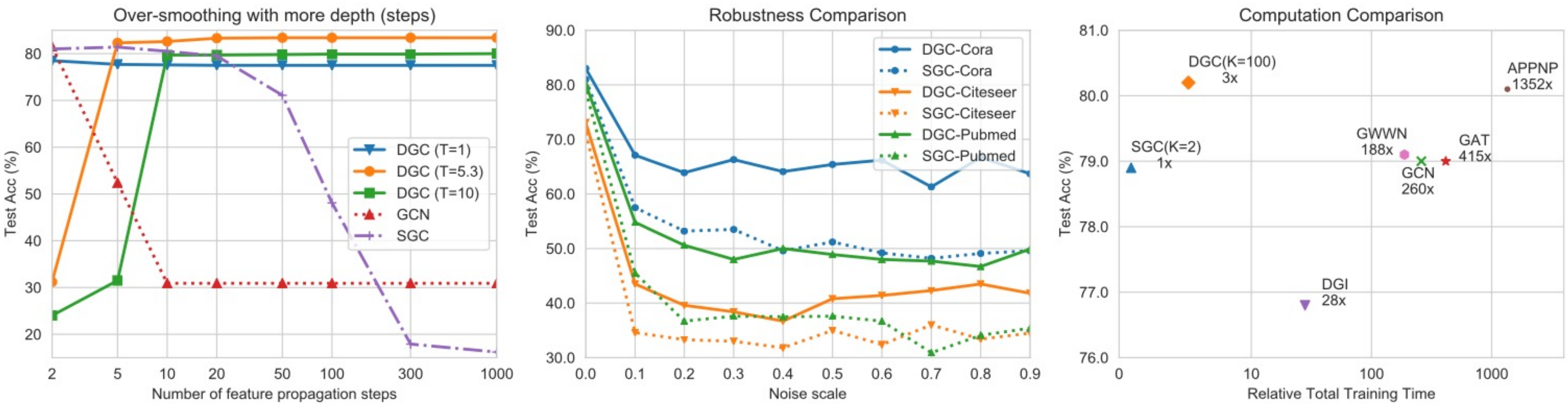
Type	Method	Cora	Citeseer	Pubmed
Non-linear	GCN [8]	85.77	73.58	88.13
	GAT [20]	86.37	74.32	87.62
	JK-MaxPool [26]	89.6	77.7	-
	JK-Concat [26]	89.1	78.3	-
	JK-LSTM [26]	85.8	74.7	-
	APPNP [9]	90.21	79.8	86.29
Linear	SGC [22]	85.82	78.08	83.27
	DGC (ours)	**88.2 ± 0.1**	**79.0 ± 0.2**	**88.7 ± 0.0**

Table 4: Test accuracy (%) comparison with inductive methods on on a large scale dataset, Reddit. Reported results are averaged over 10 runs. OOM: out of memory.

Type	Method	Acc.
Non-linear	GCN [8]	OOM
	FastGCN [3]	93.7
	GraphSAGE-GCN [6]	93.0
	GraphSAGE-mean [6]	95.0
	GraphSAGE-LSTM [6]	95.4
	APPNP [9]	95.0
Linear	RandDGI [21]	93.3
	SGC [22]	94.9
	DGC (ours)	**95.8**

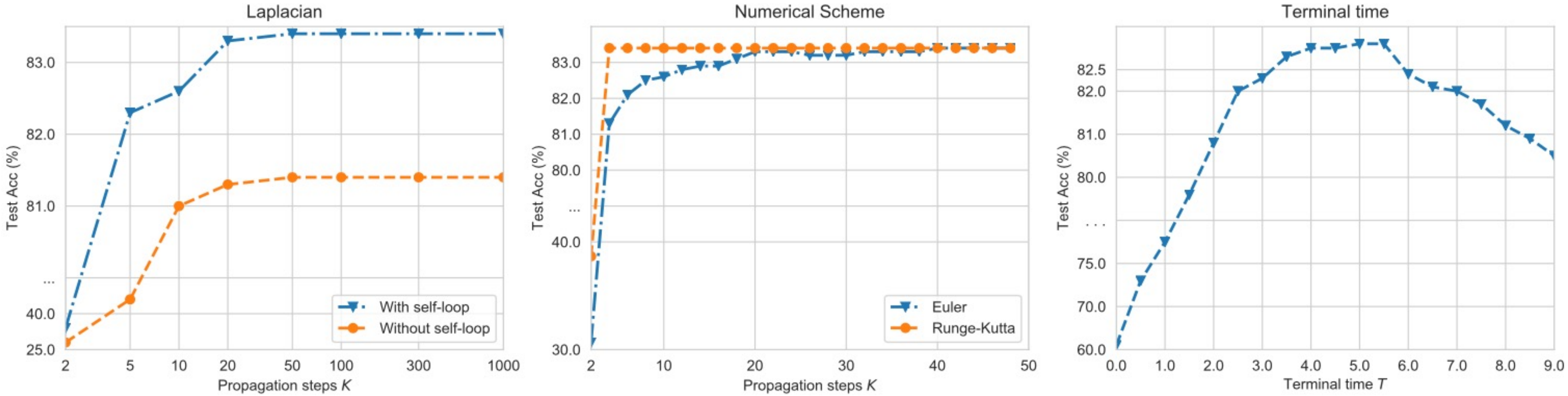
Experiments

- Empirical Understandings of DGC
  - Left: over-smoothing with increasing steps
  - Middle: robustness to feature noise
  - Right: computation time



Experiments

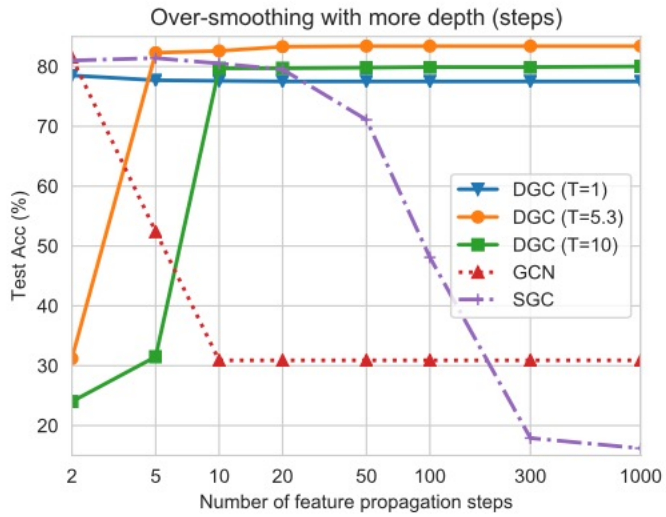
- Empirical Understandings of DGC
  - Left: graph Laplacian
  - Middle: numerical scheme
  - Right: terminal time



Takeaways

- The diffusion process can be understood through continuous PDEs
- This perspective inspires us to design more accurate and robust (linear) GCNs by simply decoupling T and K
- A properly designed linear GCN is comparable to SOTA nonlinear ones
- We should propose new alternatives that can truly benefit from nonlinear architectures

Thanks!



Q & A

Find more stuff about this work at https://yifeiwang77.github.io/
Contact:
yifei_wang AT pku.edu.cn; yisen.wang AT pku.edu.cn