Interpreting and Unifying Graph Neural Networks with An Optimization Framework

Meiqi Zhu\textsuperscript{1}, Xiao Wang\textsuperscript{1}, Chuan Shi\textsuperscript{1}, Houye Ji\textsuperscript{1}, Peng Cui\textsuperscript{2}

\textsuperscript{1}Beijing University of Posts and Telecommunications
\textsuperscript{2}Tsinghua University
The propagation mechanism is the most fundamental part of GNNs.

The Propagation Process

\[ Z = \text{PROPAGATE}(X; \mathcal{G}; K) = \left( \text{Trans}\left( \text{Agg}\{\mathcal{G}; Z^{(k-1)}\} \right) \right)_K \]

- \( \text{Agg}\{\mathcal{G}; Z^{(k-1)}\} \)
- \( \text{Trans}() \)
- \( \langle \rangle_K \)

SGC, APPNP...

GCN, GIN...

Rethinking the Propagation

- Is there a **unified framework** that essentially governs the **propagation mechanisms** of different GNNs? If so, what is it?

- Can it bring **new insights** for new GNNs **designing**?
The Unified Framework
The Unified Optimization Framework

\[ O = \min_Z \left\{ \zeta \| F_1 Z - F_2 H \|^2_F + \xi \text{tr}(Z^T \tilde{L} Z) \right\} \]

- **Feature**
  - **Flexible Feature Fitting Term**
  - **Graph Laplacian Regularization Term**

- **Flexible Graph Convolutional Kernels**
  - \( F_1, F_2 \rightarrow I, \hat{A}, \hat{L} \)
  - \( \zeta = 0 \) or \( \zeta = 1 \)

- **\( \xi \)** is a non-negative coefficient

\[ O_{reg} = \frac{\xi}{2} \sum_{i,j} \hat{A}_{i,j} \| Z_i - Z_j \|^2 = \xi \text{tr}(Z^T \tilde{L} Z) \]
**Interpreting PPNP and APPNP.**

**01 PPNP**

\[
Z = \text{PROPAGATE}(X; \mathcal{G}; \infty)_{\text{ppnp}} \\
\quad = \alpha (I - (1 - \alpha) \hat{A})^{-1} H, \quad \text{and} \quad H = f_\theta(X),
\]

**02 APPNP**

\[
Z = \text{PROPAGATE}(X; \mathcal{G}; K)_{\text{appnp}} \\
\quad = \left((1 - \alpha) \hat{A} Z^{(k-1)} + \alpha H\right)_K, \quad \text{and} \quad Z^{(0)} = H = f_\theta(X).
\]

**Theorem 3.3.** With \( F_1 = F_2 = I, \zeta = 1, \bar{\xi} = 1/\alpha - 1, \alpha \in (0, 1] \) in Eq. (3), the propagation process of PPNP/APPNP optimizes the following objective:

\[
O = \min_Z \left\{ \|Z - H\|_F^2 + \xi \text{tr}(Z^T \tilde{L} Z) \right\},
\]

where \( H = f_\theta(X) \).

**Proof.**

\[
\frac{\partial \left\{ \|Z - H\|_F^2 + \xi \text{tr}(Z^T \tilde{L} Z) \right\}}{\partial Z} = 0 \quad \Rightarrow \quad Z - H + \bar{\xi} \tilde{L} Z = 0.
\]

\[
Z = \left\{ I + (1/\alpha - 1)(I - \hat{A}) \right\}^{-1} H = \alpha (I - (1 - \alpha) \hat{A})^{-1} H.
\]
Interpreting GCN and SGC.

**GCN**

\[
Z = \text{PROPAGATE}(X; G; K)_{\text{gc}} = \hat{\Delta} \sigma (\cdots \sigma(\hat{\Delta}XW^{(0)}) \cdots )W^{(K-1)}.
\]

**SGC**

\[
Z = \text{PROPAGATE}(X; G; K)_{\text{sg}} = \hat{\Delta} \cdots \hat{\Delta}XW^{(0)} \cdots W^{(K-1)} = \hat{\Delta}^K XW^*.
\]

**Theorem 3.1.** With $\zeta = 0$ and $\xi = 1$ in Eq. (3), the propagation process of SGC/GCN optimizes the following graph regularization term:

\[
O = \min_Z \{tr(Z^T \tilde{L}Z)\},
\]

where $Z$ is initialized as $XW^*$.

Interpreting GC Operation.

**GC operation**

\[
Z = \text{PROPAGATE}(X; G; 1)_{\text{gc}} = \hat{\Delta}XW.
\]

**Theorem 3.2.** With $F_1 = F_2 = I$, $\zeta = 1$, $\xi = 1$ in Eq. (3), the 1-layer GC operation optimizes the following objective under first-order approximation:

\[
O = \min_Z \{\|Z - H\|_F^2 + tr(Z^T \tilde{L}Z)\},
\]

where $H = XW$ is the linear transformation on feature, $W$ is a trainable weight matrix.
Interpreting JKNet.

**01 JKNet**

\[ Z = \text{PROPAGATE}(X; G; K)_{JKNet} \]

\[ = \alpha_1 \hat{A}XW^* + \alpha_2 \hat{A}^2XW^* + \cdots + \alpha_K \hat{A}^KXW^* \]

\[ = \sum_{k=1}^{K} \alpha_k \hat{A}^kXW^*, \]

**Theorem 3.4.** With \( F_1 = I, F_2 = \hat{A}, \zeta = 1, \) and \( \xi \in (0, \infty) \) in Eq. (3), the propagation process of JKNet optimizes the following objective:

\[ O = \min_{Z} \left\{ \|Z - \hat{A}H\|_F^2 + \xi \text{tr}(Z^T \tilde{L}Z) \right\}, \]

where \( H = XW^* \) is the linear feature transformation after simplifications.

Interpreting DAGNN.

**02 DAGNN**

\[ Z = \text{PROPAGATE}(X; G; K)_{DAGNN} \]

\[ = s_0 H + s_1 \hat{A}H + s_2 \hat{A}^2H + \cdots + s_K \hat{A}^K H \]

\[ = \sum_{k=0}^{K} s_k \hat{A}^kH, \quad \text{and} \quad H = f_\theta(X). \]

**Theorem 3.5.** With \( F_1 = F_2 = I, \zeta = 1 \) and \( \xi \in (0, \infty) \) in Eq. (3), the propagation process of DAGNN optimizes the following objective:

\[ O = \min_{Z} \left\{ \|Z - H\|_F^2 + \xi \text{tr}(Z^T \tilde{L}Z) \right\}, \]

where \( H = f_\theta(X) \) is the non-linear transformation on feature matrix, the retention scores \( s_0, s_1, \cdots, s_K \) are approximated by \( \xi \in (0, \infty) \).
Overall Correspondences.

<table>
<thead>
<tr>
<th>Model</th>
<th>Characteristic</th>
<th>Propagation Mechanism</th>
<th>Corresponding Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN/SGC [13]</td>
<td>$K$-layer graph convolutions</td>
<td>$Z = \hat{A}^K X W^*$</td>
<td>$O = \min_{Z} \left{ \text{tr}(Z^T \hat{L} Z), Z^{(0)} = X W^* \right}$</td>
</tr>
<tr>
<td>GC Operation [13]</td>
<td>1-layer graph convolution</td>
<td>$Z = \hat{A} X W$</td>
<td>$O = \min_{Z} \left{ |Z - H|_F^2 + \text{tr}(Z^T \hat{L} Z) \right}, H = X W, (\text{first-order})$</td>
</tr>
<tr>
<td>PPNP/APPNP [14]</td>
<td>Personalized pagerank</td>
<td>$H = f_\theta(X), \begin{cases} \text{PPNP: } Z = \alpha(1 - \alpha \hat{A})^{-1} H \ \text{APPNP: } Z = (1 - \alpha \hat{A} Z^{(k-1)} + \alpha H)_K, Z^{(0)} = H \end{cases}$</td>
<td>$O = \min_{Z} \left{ |Z - H|_F^2 + (1/\alpha - 1)\text{tr}(Z^T \hat{L} Z) \right}$</td>
</tr>
<tr>
<td>JKNNet [38]</td>
<td>Jumping to the last layer</td>
<td>$Z = \sum_{k=1}^{K} \alpha_k \hat{A}^k X W^*$</td>
<td>$O = \min_{Z} \left{ |Z - \hat{A} H|_F^2 + \xi \text{tr}(Z^T \hat{L} Z) \right}, H = X W^*$</td>
</tr>
<tr>
<td>DAGNN [17]</td>
<td>Adaptively incorporating different layers</td>
<td>$H = f_\theta(X), Z = \sum_{k=0}^{K} s_k \hat{A}^k H$</td>
<td>$O = \min_{Z} \left{ |Z - H|_F^2 + \xi \text{tr}(Z^T \hat{L} Z) \right}$</td>
</tr>
</tbody>
</table>

Discussion.

- **Understanding relationships** between different GNNs is much easier.
- The unified framework opens up **new insights** for designing novel GNNs.
New GNNs Design
**GNN-LF: GNN with Low-pass Filtering Kernel**

**Theorem 4.1.** With \( F_1 = F_2 = \{ \mu I + (1 - \mu)\hat{A} \}^{1/2}, \mu \in [1/2, 1], \) \( \zeta = 1 \) and \( \xi = 1/\alpha - 1, \alpha \in (0, 2/3) \) in Eq. (3), the propagation process considering flexible low-pass filtering kernel on feature is:

\[
O = \min \{ \| \{ \mu I + (1 - \mu)\hat{A} \}^{1/2} (Z - H) \|_F^2 + \xi \text{tr}(Z^T \hat{L}Z) \},
\]

where \( H = f_0(X) \).

**Objective**

**Closed Solution.**

\[
Z = \text{PROPAGATE}(X; G; \infty)_{\text{LF-closed}} = \left\{ \mu + 1/\alpha - 1 \right\}I + \left\{ 2 - \mu - 1/\alpha \right\}\hat{A}^{-1} \{ \mu I + (1 - \mu)\hat{A} \}H,
\]

and \( H = f_0(X) \).

**Iterative Approximation.**

\[
Z^{(0)} = \frac{\mu}{1 + \alpha \mu - \alpha} H + \frac{1 - \mu}{1 + \alpha \mu - \alpha} \hat{A} H, \quad \text{and} \quad H = f_0(X).
\]

**Model**

**Theorem 4.2.** With \( K \to \infty \), deep GNN-LF-iter converges to GNN-LF-closed with the same propagation result as Eq. (31).
**GNN-HF: GNN with High-pass Filtering Kernel**

**Objective**

\[
O = \min_{Z} \left\| \frac{1}{2}(Z - H) \right\|_F^2 + \xi \text{tr}(Z^T \tilde{L} Z),
\]

where \( H = f_\theta(X) \).

**Closed Solution.**

\[
Z = \text{PROPAGATE}(X; G; \infty)_{HF-closed}
= \{(\beta + 1/\alpha)I + (1 - \beta - 1/\alpha)\hat{\Delta}\}^{-1}(I + \beta\hat{L})H,
\]

and \( H = f_\theta(X) \).

**Iterative Approximation.**

\[
Z_{\text{iter}} = \text{PROPAGATE}(X; G; K)_{HF-iter}
= \left\{ \frac{\alpha\beta - \alpha + 1}{\alpha\beta + 1} \hat{Z}^{(k-1)} + \frac{\alpha}{\alpha\beta + 1} H + \frac{\alpha\beta}{\alpha\beta + 1} \hat{L}H \right\}_K,
\]

\[
Z^{(0)} = \frac{1}{\alpha\beta + 1} H + \frac{\beta}{\alpha\beta + 1} \hat{L}H, \quad \text{and} \quad H = f_\theta(X).
\]

**Theorem 4.4.** When \( K \to \infty \), deep GNN-HF-iter converges to GNN-HF-closed with the same propagation result as Eq. (39).
**Spectral Expressive Power Analysis**

K-order polynomial filter on graph signal \( \mathbf{X} \in \mathbb{R}^{n \times f} \):

\[
\left( \sum_{k=0}^{K} \theta_k \hat{L}^k \right) \mathbf{X}
\]

\( \hat{L} \) is the normalized Laplacian matrix.

**GNN-LF/HF**

1) Filter coefficients for \( \hat{L}^0 \):

\[
\theta_0 = \frac{\alpha \mu (1 + \alpha \mu - 2 \alpha)}{(1 + \alpha \mu - \alpha)^2} + \frac{(\alpha - \alpha \mu) (1 + \alpha \mu - 2 \alpha)^{K-1}}{(1 + \alpha \mu - \alpha)^K} + \sum_{j=1}^{K-1} \delta_j \left( \begin{array}{c} K \\ j \end{array} \right).
\]

2) Filter coefficients for \( \hat{L}^1 \), \( k \in [1, K-1] \):

\[
\theta_k = \sum_{j=k}^{K} \delta_j (-1)^j \left( \begin{array}{c} K \\ j \end{array} \right).
\]

3) Filter coefficients for \( \hat{L}^K \):

\[
\theta_K = \frac{(\alpha - \alpha \mu) (1 + \alpha \mu - 2 \alpha)^{K-1}}{(1 + \alpha \mu - \alpha)^K} (-1)^K \left( \begin{array}{c} K \\ K \end{array} \right).
\]

**SGC**

\[
\mathbf{Z}^{(K)} = \hat{\mathbf{A}}^K \mathbf{X} = (I - \hat{L})^K \mathbf{X}.
\]

\[
\theta_k = (-1)^k \binom{K}{k}
\]

**PPNP/APPNP**

\[
\mathbf{Z}^{(K)} = \alpha \sum_{i=0}^{K-1} (1 - \alpha)^i (I - \hat{L})^i \mathbf{X}.
\]

\[
\theta_k = \alpha \sum_{i=k}^{K-1} (1 - \alpha)^i (-1)^k \binom{i}{k}
\]

Over-smoothing → Expressive Power

- **Fixed Constant**
- **Flexible**
Experiments
Datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classes</th>
<th>Nodes</th>
<th>Edges</th>
<th>Features</th>
<th>Train/Val/Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora</td>
<td>7</td>
<td>2708</td>
<td>5429</td>
<td>1433</td>
<td>140/500/1000</td>
</tr>
<tr>
<td>Citeseer</td>
<td>6</td>
<td>3327</td>
<td>4732</td>
<td>3703</td>
<td>120/500/1000</td>
</tr>
<tr>
<td>Pubmed</td>
<td>3</td>
<td>19717</td>
<td>44338</td>
<td>500</td>
<td>60/500/1000</td>
</tr>
<tr>
<td>ACM</td>
<td>3</td>
<td>3025</td>
<td>13128</td>
<td>1870</td>
<td>60/500/1000</td>
</tr>
<tr>
<td>Wiki-CS</td>
<td>10</td>
<td>11701</td>
<td>216123</td>
<td>300</td>
<td>200/500/1000</td>
</tr>
<tr>
<td>MS Academic</td>
<td>15</td>
<td>18333</td>
<td>81894</td>
<td>6805</td>
<td>300/500/1000</td>
</tr>
</tbody>
</table>

Metrics.

\[ \text{ACC} \pm \text{uncertainties} \]

BaseLines.

01 Traditional graph learning methods
   - MLP, LP

02 Spectral methods
   - ChebNet, GCN

03 Spatial methods
   - SGC, GAT, GraphSAGE, PPNP

04 Deep GNN methods
   - JKNet, APPNP, IncepGCN
# Node Classification

<table>
<thead>
<tr>
<th>Model</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
<th>ACM</th>
<th>Wiki-CS</th>
<th>MS Academic</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>57.79±0.11</td>
<td>61.20±0.08</td>
<td>73.23±0.05</td>
<td>77.39±0.11</td>
<td>65.66±0.20</td>
<td>87.79±0.42</td>
</tr>
<tr>
<td>LP</td>
<td>71.50±0.00</td>
<td>50.80±0.00</td>
<td>72.70±0.00</td>
<td>63.30±0.00</td>
<td>34.90±0.00</td>
<td>74.10±0.00</td>
</tr>
<tr>
<td>ChebNet</td>
<td>79.92±0.18</td>
<td>70.90±0.37</td>
<td>76.98±0.16</td>
<td>79.53±1.24</td>
<td>63.24±1.43</td>
<td>90.76±0.73</td>
</tr>
<tr>
<td>GAT</td>
<td>82.48±0.31</td>
<td>72.08±0.41</td>
<td>79.08±0.22</td>
<td>88.24±0.38</td>
<td>74.27±0.63</td>
<td>91.58±0.25</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>82.14±0.25</td>
<td>71.80±0.36</td>
<td>79.20±0.27</td>
<td>87.57±0.65</td>
<td>73.17±0.41</td>
<td>91.53±0.15</td>
</tr>
<tr>
<td>IncepGCN</td>
<td>81.94±0.94</td>
<td>69.66±0.29</td>
<td>78.88±0.35</td>
<td>87.75±0.61</td>
<td>60.54±1.06</td>
<td>75.45±0.49</td>
</tr>
<tr>
<td>GCN</td>
<td>82.41±0.25</td>
<td>70.72±0.36</td>
<td>79.40±0.15</td>
<td>88.38±0.51</td>
<td>71.97±0.51</td>
<td>92.17±0.11</td>
</tr>
<tr>
<td>SGC</td>
<td>81.90±0.23</td>
<td>72.21±0.22</td>
<td>78.30±0.14</td>
<td>87.56±0.34</td>
<td>72.43±0.28</td>
<td>88.35±0.36</td>
</tr>
<tr>
<td>PPNP</td>
<td>83.34±0.20</td>
<td>71.73±0.30</td>
<td>80.06±0.20</td>
<td>89.12±0.17</td>
<td>74.53±0.36</td>
<td>92.27±0.23</td>
</tr>
<tr>
<td>APPNP</td>
<td>83.32±0.42</td>
<td>71.67±0.48</td>
<td>80.05±0.27</td>
<td>89.04±0.21</td>
<td>74.30±0.50</td>
<td>92.25±0.18</td>
</tr>
<tr>
<td>JKNNet</td>
<td>81.19±0.49</td>
<td>70.69±0.88</td>
<td>78.60±0.25</td>
<td>88.11±0.36</td>
<td>60.90±0.92</td>
<td>87.26±0.23</td>
</tr>
<tr>
<td>GNN-LF-closed</td>
<td>83.70±0.14</td>
<td>71.98±0.33</td>
<td>80.34±0.18</td>
<td>89.43±0.20</td>
<td>75.50±0.56</td>
<td>92.79±0.15</td>
</tr>
<tr>
<td>GNN-LF-iter</td>
<td>83.53±0.24</td>
<td>71.92±0.24</td>
<td>80.33±0.20</td>
<td>89.37±0.40</td>
<td>75.35±0.24</td>
<td>92.69±0.20</td>
</tr>
<tr>
<td>GNN-HF-closed</td>
<td>83.96±0.22</td>
<td>72.30±0.28</td>
<td>80.41±0.25</td>
<td>89.46±0.30</td>
<td>74.92±0.45</td>
<td>92.47±0.23</td>
</tr>
<tr>
<td>GNN-HF-iter</td>
<td>83.79±0.29</td>
<td>72.03±0.36</td>
<td>80.54±0.25</td>
<td>89.59±0.31</td>
<td>74.90±0.37</td>
<td>92.51±0.16</td>
</tr>
</tbody>
</table>
Propagation Depth Analysis

(a) Cora
(b) Citeseer
(c) Pubmed
Model Analysis

(a) GNN-LF

(b) GNN-HF
Conclusions
A unified optimization framework.

We propose a unified objective optimization framework and theoretically prove that this framework is able to unify a series of GNNs propagation mechanisms.

A new insight for designing GNNs.

Within the proposed optimization framework, we design two novel deep GNNs with flexible low-frequency and high-frequency filters which can well alleviate over-smoothing.

Extensive experiments verify the feasibility.

Our extensive experiments clearly show that the proposed two GNNs outperform the state-of-the-art methods. This further verifies the feasibility for designing GNNs under the unified framework.
Resources

公众号：北邮GAMMA Lab
公众号：图与推荐

石川老师主页：http://shichuan.org/

图与推荐交流QQ群
Thanks!
Q&A