Appendix: Enhancing Multi-view Graph Neural Network with Cross-view Confluent Message Passing

Anonymous Authors

ABSTRACT

In this appendix, we first provide detailed derivations mentioned in the main paper for better understanding. Then, we present experimental settings, including dataset statistics and hyperparameter selection. Finally, we supplement some experimental results and a training algorithm.

A DERIVATIONS

Here, we first provide detailed derivations about graph Laplacian regularization:

$$\begin{aligned}
&\frac{1}{2}\sum_{ij} \mathbf{A}_{ij}||\frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}}||_2^2 \\
=&\frac{1}{2}\sum_{ij} \mathbf{A}_{ij}(\frac{\mathbf{z}_i^\top \mathbf{z}_i}{d_i} - 2\frac{\mathbf{z}_i^\top \mathbf{z}_j}{\sqrt{d_i d_j}} + \frac{\mathbf{z}_j^\top \mathbf{z}_j}{d_j}) \\
=&\frac{1}{2}\sum_{ij} \mathbf{A}_{ij}\frac{\mathbf{z}_i^\top \mathbf{z}_i}{d_i} - \sum_{ij} \mathbf{A}_{ij}\frac{\mathbf{z}_i^\top \mathbf{z}_j}{\sqrt{d_i d_j}} + \frac{1}{2}\sum_{ij} \mathbf{A}_{ij}\frac{\mathbf{z}_j^\top \mathbf{z}_j}{d_j} \\
=&\frac{1}{2}\sum_{i} \mathbf{z}_i^\top \mathbf{z}_i - \sum_{ij} \frac{\mathbf{A}_{ij}}{\sqrt{d_i d_j}}\mathbf{z}_i^\top \mathbf{z}_j + \frac{1}{2}\sum_{j} \mathbf{z}_j^\top \mathbf{z}_j \\
=&\sum_{i} \mathbf{z}_i^\top \mathbf{z}_i - \sum_{ij} (d_i d_j)^{-\frac{1}{2}} \mathbf{A}_{ij}\mathbf{z}_i^\top \mathbf{z}_j \\
=&\mathrm{Tr}\left(\mathbf{Z}^\top(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{Z}\right) \\
=&\mathrm{Tr}\left(\mathbf{Z}^\top\hat{\mathbf{L}}\mathbf{Z}\right)
\end{aligned} \tag{1}$$

We further elaborate on how vanilla GCN and CGCN are induced by optimization problems.

Theorem A.1. *Given $\mathbf{Z}^{(0)} = \mathbf{X}$, the message passing of the vanilla GCN [1]*

$$\mathbf{Z}^{(l+1)} = \hat{\mathbf{A}}\mathbf{Z}^{(l)} \tag{2}$$

is optimizing the following objective

$$\min_{\mathbf{Z}} \sum_{ij} \mathbf{A}_{ij}||\frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}}||_2^2. \tag{3}$$

Proof. According to Eq. (1), Problem (3) is equal to

$$\min_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}) := \mathrm{Tr}(\mathbf{Z}^\top\hat{\mathbf{L}}\mathbf{Z}). \tag{4}$$

Taking derivative of $\mathcal{L}(\mathbf{Z})$ w.r.t $\mathbf{Z}$, we have

$$\hat{\mathbf{L}}\mathbf{Z} = 0 \ \Rightarrow (\mathbf{I} - \hat{\mathbf{A}})\mathbf{Z} = 0 \ \Rightarrow \mathbf{Z} = \hat{\mathbf{A}}\mathbf{Z}, \tag{5}$$

which can be regarded as a limit distribution $\mathbf{Z}_{lim} = \hat{\mathbf{A}}\mathbf{Z}_{lim}$. Therefore we solve Problem (3) in an iterative form

$$\mathbf{Z}^{(l+1)} = \hat{\mathbf{A}}\mathbf{Z}^{(l)}, \tag{6}$$

that is $\mathbf{Z}^{(l)} = \hat{\mathbf{A}}^{(l)}\mathbf{X}$. Eq. (6) is to approximate the limit with $l \to \infty$. □

Theorem A.2. *the Cross-view Confulent Message Passing (CCMP) (with computed $[\mathbf{S}^v]^{(l)}$) of CGCN*

$$\mathbf{z}_i^{(l+1)} = \frac{\lambda_1}{1+\lambda_1}\sum_{i,j}\sum_{v}\mu_v^{(l)}[\hat{\mathbf{S}}_{ij}^v]^{(l)}\mathbf{z}_j^{(l)} + \frac{1}{1+\lambda_1}\mathbf{x}_i \tag{7}$$

is optimizing the following objective

$$\min_{\mathbf{Z}} \mathcal{L} := ||\mathbf{Z}-\mathbf{X}||_F^2 + \frac{\lambda_1}{2}\sum_{i,j}\sum_{v}\mu_v \mathbf{S}_{ij}^v||\bar{\mathbf{z}}_i - \bar{\mathbf{z}}_j||_2^2. \tag{8}$$

Proof. According to Eq. (1), Problem (8) is equal to

$$\min_{\mathbf{Z}} \mathcal{L} := ||\mathbf{Z}-\mathbf{X}||_F^2 + \lambda_1\sum_{v}\mu_v \mathrm{Tr}(\mathbf{Z}^\top\hat{\mathbf{L}}_{\mathbf{S}}^v\mathbf{Z}) \tag{9}$$

Taking derivative of Eq. (9) w.r.t $\mathbf{Z}$, we have

$$\mathbf{Z} + \lambda_1\sum_{v}\mu_v\hat{\mathbf{L}}_{\mathbf{S}}^v\mathbf{Z} - \mathbf{X} = 0 \ \Rightarrow \ (\mathbf{I} + \lambda_1\sum_{v}\mu_v\hat{\mathbf{L}}_{\mathbf{S}}^v)\mathbf{Z} = \mathbf{X}. \tag{10}$$

Since $\hat{\mathbf{L}}_{\mathbf{S}}^v$ is semi-definite positive, we further obtain

$$\mathbf{Z} = (\mathbf{I} + \lambda_1\sum_{v}\mu_v\hat{\mathbf{L}}_{\mathbf{S}}^v)^{-1}\mathbf{X} = \left(\mathbf{I} + \lambda_1(\mathbf{I} - \sum_{v}\mu_v\hat{\mathbf{S}}^v)\right)^{-1}\mathbf{X}, \tag{11}$$

where $\hat{\mathbf{L}}_{\mathbf{S}}^v = \mathbf{I} - \hat{\mathbf{S}}^v$. By simple algebra, it can be transformed to

$$\mathbf{Z} = \frac{1}{1+\lambda_1}(\mathbf{I} - \frac{\lambda_1}{1+\lambda_1}\sum_{v}\mu_v\hat{\mathbf{S}}^v)^{-1}\mathbf{X}. \tag{12}$$

Then $\left(\mathbf{I} - \frac{\lambda_1}{1+\lambda_1}\sum_v \mu_v\hat{\mathbf{S}}^v\right)^{-1}$ can be decomposed into Taylor series as

$$\begin{aligned}
&(\mathbf{I} - \frac{\lambda_1}{1+\lambda_1}\sum_{v}\mu_v\hat{\mathbf{S}}^v)^{-1} \\
&= \mathbf{I} + \frac{\lambda_1}{1+\lambda_1}\sum_{v}\mu_v\hat{\mathbf{S}}^v + \cdots + \left(\frac{\lambda_1}{1+\lambda_1}\right)^{(l)}\sum_{v}\mu_v[\hat{\mathbf{S}}^v]^{(l)} + \cdots,
\end{aligned} \tag{13}$$

that is $\left(\mathbf{I} - \frac{\lambda_1}{1+\lambda_1}\sum_v \mu_v\hat{\mathbf{S}}^v\right)^{-1} = \lim_{l\to\infty}\sum_{i=0}^{l}\sum_v\left(\frac{\lambda_1}{1+\lambda_1}\right)^i\mu_v[\hat{\mathbf{S}}^v]^{(i)}$, so we denote the calculated $\mathbf{Z}$ with the l-th order approximation of the matrix inversion as $\mathbf{Z}^{(l)} = \frac{1}{1+\lambda_1}\sum_{i=0}^{l}\sum_v\left(\frac{\lambda_1}{1+\lambda_1}\right)^i\mu_v[\hat{\mathbf{S}}^v]^{(i)}\mathbf{X}$. Thus, we have the iterative form

$$\mathbf{Z}^{(l+1)} = \frac{\lambda_1}{1+\lambda_1}\sum_{v}\mu_v[\hat{\mathbf{S}}^v]^{(l)}\mathbf{Z}^{(l)} + \frac{1}{1+\lambda_1}\mathbf{X}. \tag{14}$$

Writing it in node-level form gives the CCMP:

$$\mathbf{z}_i^{(l+1)} = \frac{\lambda_1}{1+\lambda_1}\sum_{i,j}\sum_{v}\mu_v^{(l)}[\hat{\mathbf{S}}_{ij}^v]^{(l)}\mathbf{z}_j^{(l)} + \frac{1}{1+\lambda_1}\mathbf{x}_i, \tag{15}$$

which completes the proof. □

Table 1: Detailed statistics of single-view graph.

Datasets	Nodes	Edges	Classes	Features	Traning Nodes
Cora	2,708	5,429	7	1,433	Citation network
Citeseer	3,327	4,732	6	3,703	Citation network
Pubmed	19,717	44,324	3	500	Citation network
ACM-S	3,025	13,128	3	1,870	Paper network
BlogCatalog	5,196	171,743	6	8,189	Social network
UAI	3,067	28,311	19	4,973	Webpage network

Table 2: Detailed statistics of multi-relational graphs.

Datasets	Nodes	Features	Views	Classes
ACM	3,025	1,870	2	3
DBLP	4,057	334	3	4
IMDB	4,780	1,232	3	3
YELP	2,614	82	3	3

Table 3: Detailed statistics of multi-attribute and multi-modality graphs.

Datasets	Nodes	Features	Views	Classes
MNIST	10,000	48	3	10
HW	2,000	1,715	6	10
Animals	10,158	8,192	2	50
BDGP	2,500	1,829	2	5
ESP-Game	11,032	200	2	7
MIRFlickr	12,154	200	2	7

B DETAILED EXPERIMENTAL SETTINGS

B.1 Datasets

In this paper, four types of datasets are adopted, including six single-view datasets (Cora, Citeseer, Pubmed, ACM, BlogCatalog, UAI), four multi-relational graphs (ACM, DBLP, IMDB, YELP), three multi-attribute graphs (MINST, HW, Animals), and three multi-modality graphs (BDGP, ESP-Game, MIRFlickr). The detailed statistics of these datasets are illustrated in Tables 1, 2 and 3, respectively. Note that the number of features in Tables 3 is the sum of feature dimensions of all views.

B.2 Hyperparameter Settings

For CGCN, the hyperparameter configuration is as follows: learning rate = 0.01, dropout = 0.5, weight decay = 5E-4, number of layers = 2, and fixed hidden unit size of 32. The values of hyperparameter α ($\alpha = \frac{1}{1+\lambda_1}$) are searched from the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, hyperparameter λ_2 is searched from $\{50, 60, 70, 80, 90\}$, and hyperparameter γ is searched from $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}\}$. For other baseline methods, we adopt the default parameter settings provided by the authors' implementations.

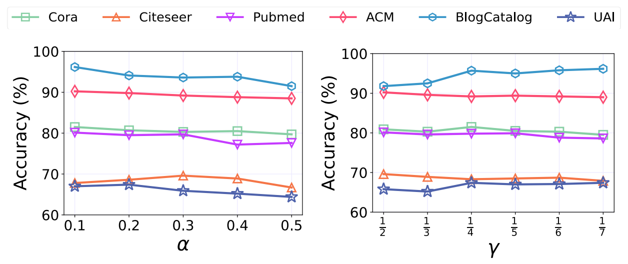
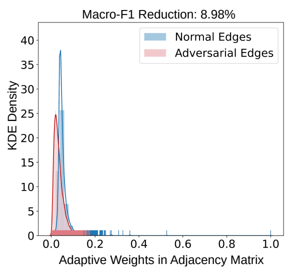
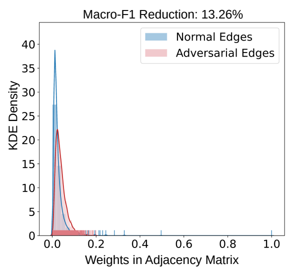


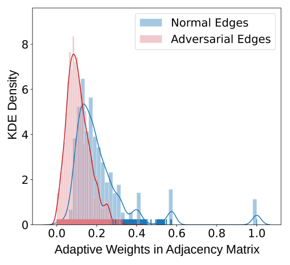
Figure 1: Parameter sensitivity on six single-view graph datasets.



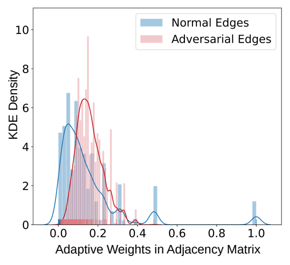
(a) w CCMP on view 1 of IMDB



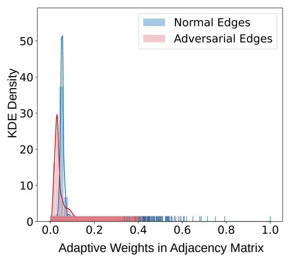
(b) w/o CCMP on view 1 of IMDB



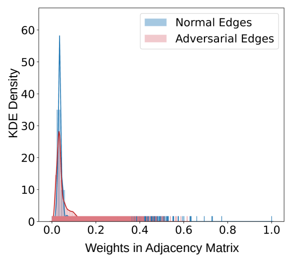
(c) w CCMP on view 2 of IMDB



(d) w/o CCMP on view 2 of IMDB



(e) w CCMP on view 3 of IMDB



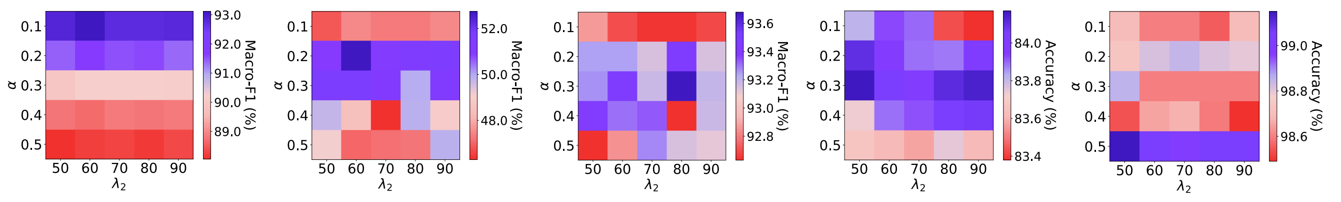
(f) w/o CCMP on view 3 of IMDB

Figure 2: Weight density distributions of normal and adversarial edges on the learned graph. We implement CGNN w and w/o CCMP on views 1-3 of IMDB datasets.

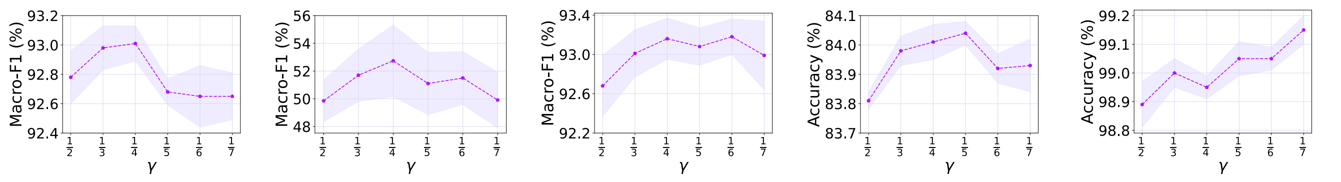
C SUPPLEMENTAL EXPERIMENTS

C.1 Robustness Analysis of Graph Structures

Attack Settings. In robustness analysis, we employ the state-of-the-art method Mettack to generate adversarial perturbations. Mettack [2] specifically targets graph structures during training, and the largest connected components of the graphs are utilized during the



(a) DBLP (b) IMDB (c) YELP (d) Animals (e) BDGP



(f) DBLP (g) IMDB (h) YELP (i) Animals (j) BDGP

Figure 3: Parameter sensitivity on five multi-view graph datasets. (a)-(e) Performance with different combinations of α and λ_2. (f)-(j) Performance curves as γ ranges in $\{\frac{1}{2}, \cdots, \frac{1}{7}\}$.

attack. The perturbation rate represents the ratio of changed edges. For instance, a 25% perturbation rate implies that the number of adversarial edges added is 25% of the original edge count.

Robustness of Multi-view Datasets. We present additional results on multi-view datasets. Specifically, we apply Mettack to each view of the IMDB dataset with a 25% perturbation rate. Figure 2 illustrates the weight density distribution of normal and adversarial edges with and without CCMP across three views. The figure shows that adversarial edges are assigned lower weights than normal edges, and the reduction in Macro-F1 post-attack has been significantly mitigated. Similar to the single-view scenario, our proposed CCMP demonstrates robustness on multi-view datasets.

C.2 Parameter Sensitivity

In this subsection, we supplement the parameter analysis to both single-view graphs and the remaining multi-view graphs.

Single-view Graph. For single-view graphs, we investigate the sensitivity of hyperparameters α ($\alpha = \frac{1}{1+\lambda_1}$) and γ, as depicted in Figure 1. When varying α from 0.1 to 0.5, the performance of CGCN consistency improves, with peak performance observed at $\alpha = 0.1$ or 0.3. Similarly, varying γ from $\frac{1}{2}$ to $\frac{1}{7}$ shows results consistently at a promising level, with slight fluctuations across all datasets. These findings suggest that our proposed CGNN maintains stability across most datasets, underscoring the efficacy of our approach in addressing single-view semi-supervised classification tasks.

Multi-view Graphs. We present the parameter analysis of five multi-view datasets not included in the main paper. The performance of CGCN concerning (α,λ_2) is illustrated in Figure 3 (a)-(e). Optimal values are observed when α ranges from 0.1 to 0.3 for multi-relational datasets and the Animals dataset, while $\alpha = 0.5$ is optimal for the BDGP dataset. Our observation validates that the judicious introduction of the graph smoothing term can enhance the model. Optimal selection of the hyperparameter α leads to stable and consistently superior performance of λ_2, indicating the effectiveness of the regularization term $\|\boldsymbol{\mu}\|_2^2$ in balancing multiple views. The impact of γ on performance is shown in Figure 3 (f)-(j). When varying γ from $\frac{1}{2}$ to $\frac{1}{7}$, a marginal enhancement is noted at $\gamma = \frac{1}{4}$ or $\frac{1}{5}$, followed by a subsequent decline in the multi-relational datasets and Animals dataset. Conversely, the performance of the BDGP dataset consistently improves, reaching its peak at $\gamma = \frac{1}{7}$. Overall, the model consistently maintains a relatively satisfactory performance across variations in γ, highlighting the stability of the hyperparameter γ. It reflects the model's ability to capture information effectively through dynamic graph structures guided by edge-level coefficients.

D TRAINING ALGORITHM

Algorithm 1 shows the updating process of variables in the main paper.

Algorithm 1: CGCN

```
Input: Multi-view graph $\mathcal{G} = \{\mathcal{G}^v\}_{v=1}^V$, feature matrix $\mathbf{X}$,
       layer number $L$, iteration number $K$,
       hyperparameters $\lambda_1, \lambda_2, \gamma$.
Output: Node embedding $\mathbf{z}_{out}$.
1  Initialize $\mathbf{W}_1$ and $\mathbf{W}_2$;
2  $\mathbf{z}_i^{(0)} = \mathrm{ReLU}(\mathbf{x}_i \mathbf{W}_1)$;
3  for $l = 0 \rightarrow (L-1)$ do
4      for $k = 1 \rightarrow (K-1)$ do
5          Initialize $\boldsymbol{\mu}^1 = \{\frac{1}{V}, ..., \frac{1}{V}\}$;
6          Calculate $\mu_v^{k+1}$ with Eq. (12);
7          When convergence, terminate the loop to
8          obtain $\mu_v^{(l)} \leftarrow \mu_v^{k+1}$;
9      Calculate $[\mathbf{S}_{ij}^v]^{(l)}$ with Eq. (13);
10     Calculate $\mathbf{z}_i^{(l+1)}$ with Equ. (16);
11 $\mathbf{z}_{out} = \mathrm{softmax}(\mathbf{z}_i^{(L)} \mathbf{W}_2)$;
12 return Node embedding $\mathbf{z}_{out}$.
```

REFERENCES

[1] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[2] Daniel Zugner and Stephan Gunnemann. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *ICLR*.