

Bootstrapping Informative Graph Augmentation via A Meta Learning Approach

Hang Gao^{1,2*}, **Jiangmeng Li^{1,2*}**, **Wenwen Qiang^{1,2†}**, **Lingyu Si^{1,2}**, **Fuchun Sun³**,
Changwen Zheng²

¹University of Chinese Academy of Sciences

²Institute of Software Chinese Academy of Sciences

³Tsinghua University

{gaohang, jiangmeng2019, wenwen2018, lingyu, changwen}@iscas.ac.cn, fcsun@tsinghua.edu.cn

IJCAI 2022

Code: github.com/hang53/mega

2022. 06. 15 • ChongQing

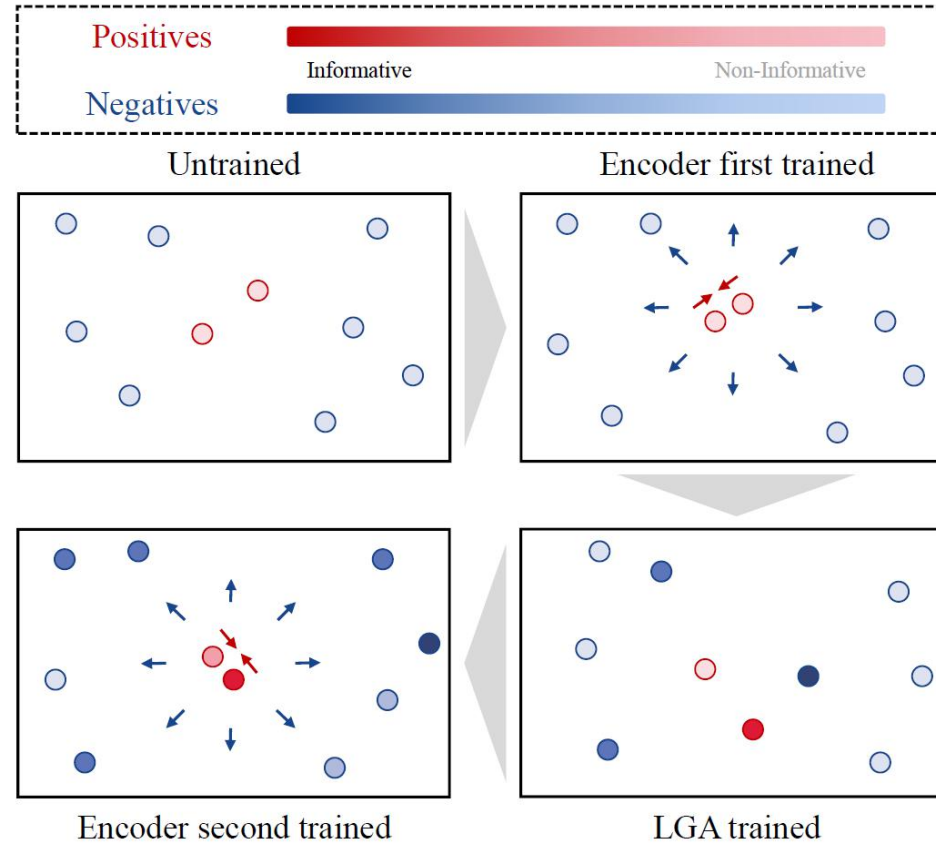


gesis
Leibniz-Institut
für Sozialwissenschaften



Reported by Chenghong Li

Introduction



Learnable Graph Augmenter (LGA)

Figure 1: An illustration example of the training process of MEGA. The figure shows the features in hidden space during training. Red points depicts the positive features and blue points depicts the negative features at the instance-level. The gradation of color denotes the informativeness at the feature-level.

Method

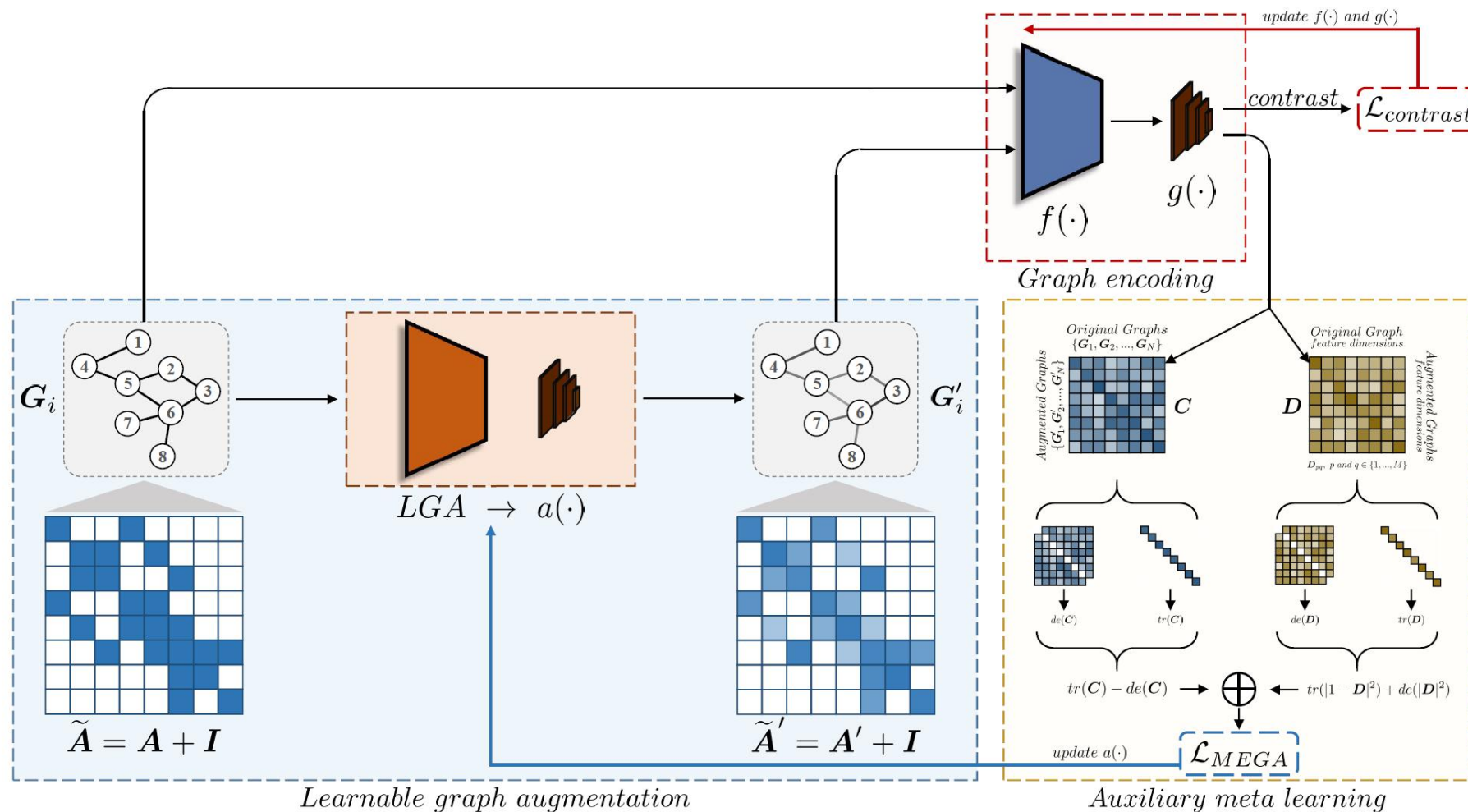
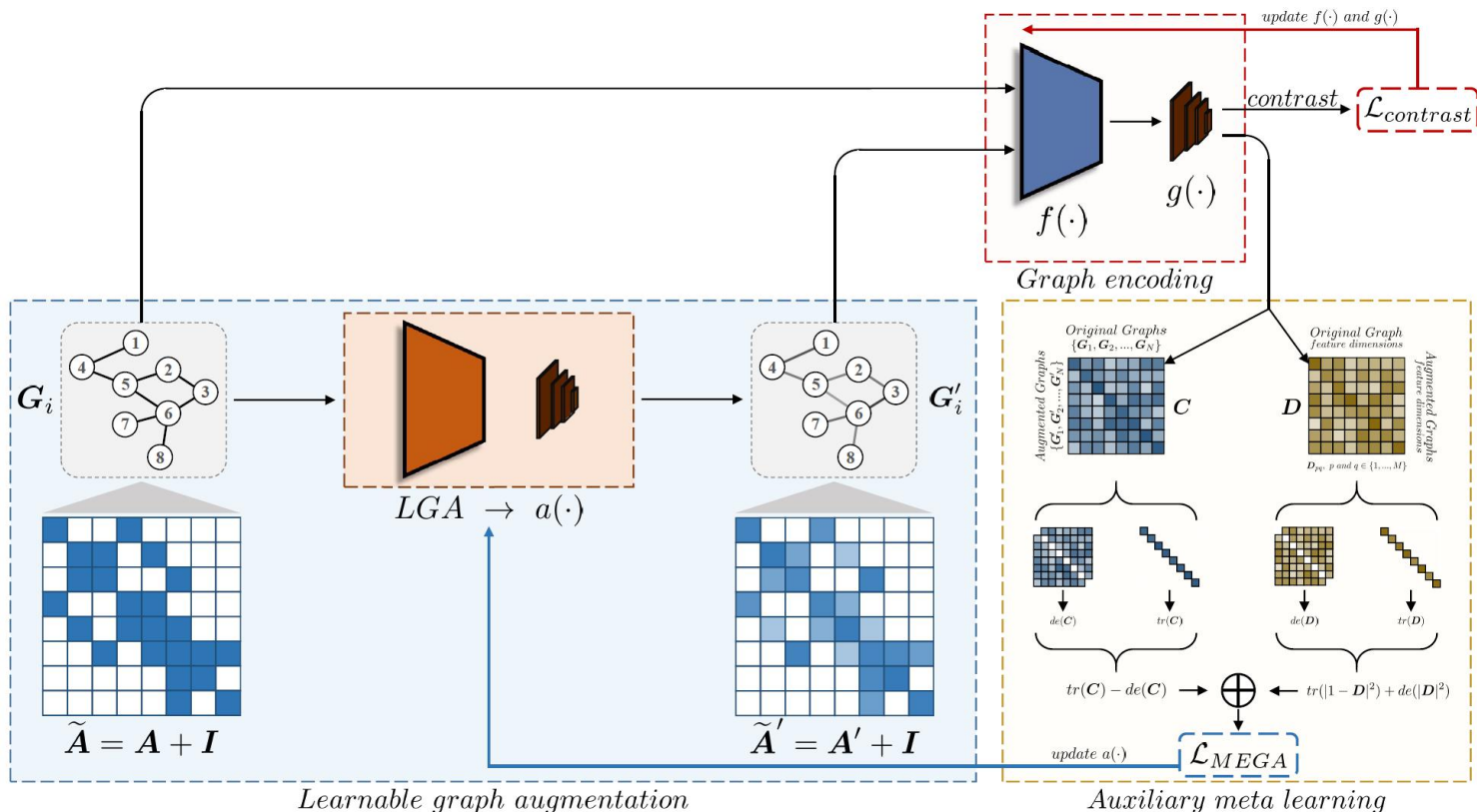


Figure 2: MEGA's architecture. MEGA uses LGA to generate augmented graph, which and the original graph are encoded together. In one iteration, the encoder and projection head are trained by back-propagating $\mathcal{L}_{contrast}$, and in the next iteration, the LGA is trained by performing the second-derivative technique on \mathcal{L}_{MEGA} . The encoder is trained until convergence.

Method



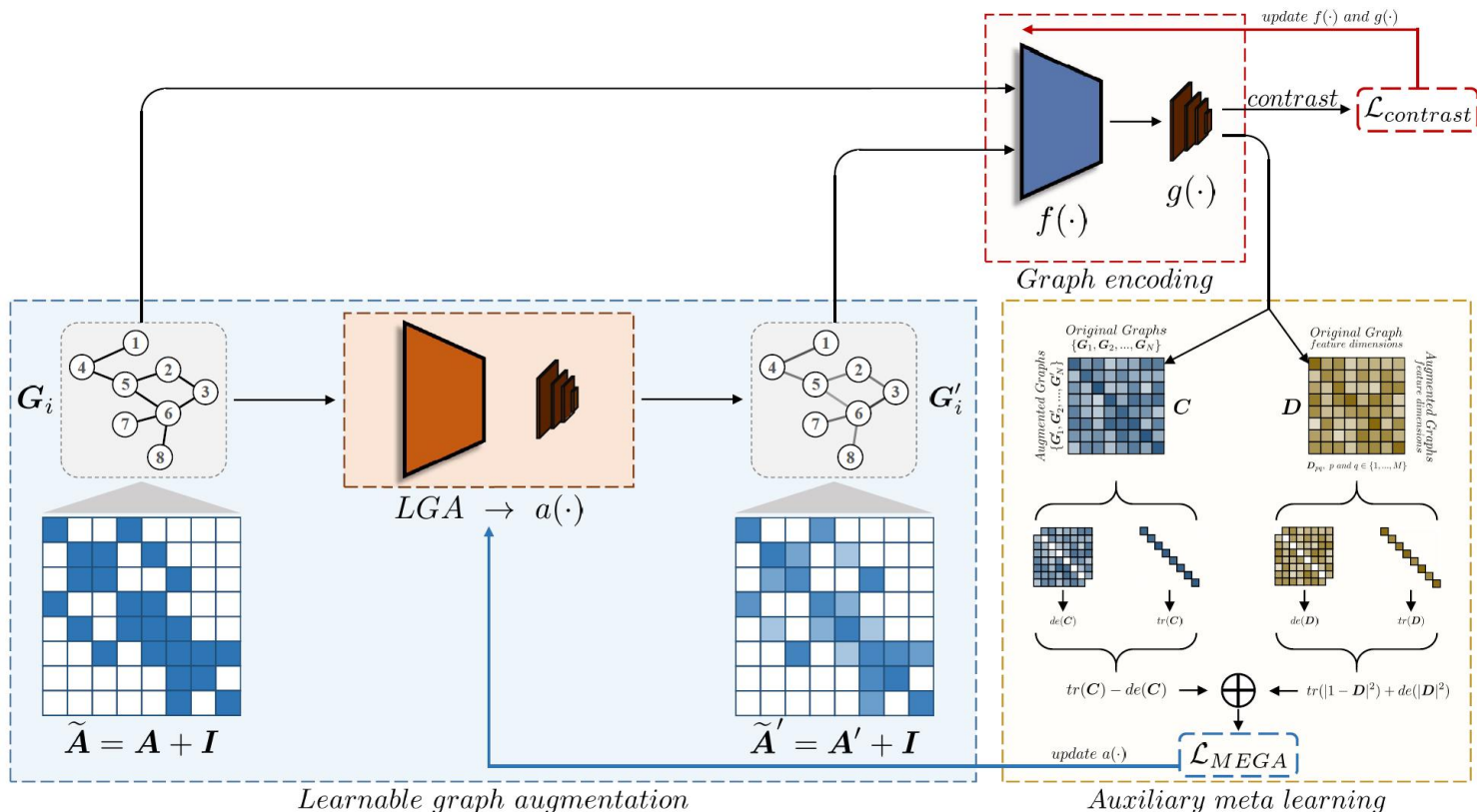
$$\mathbf{H}_v^{(k+1)} = \text{combine}^{(k)} \left(\mathbf{H}_v^k, \text{aggregate}^{(k)} \left(\mathbf{H}_u^k, \forall u \in \mathcal{N}(v) \right) \right), \quad (1)$$

$$\mathbf{h}_i = \text{readout} \left(\mathbf{H}_v, v \in \mathbf{V}_i \right). \quad (2)$$

$$\mathcal{L}_{contrast} = -\log \frac{\exp \left(\frac{\langle \mathbf{z}^+ \rangle}{\tau} \right)}{\exp \left(\frac{\langle \mathbf{z}^+ \rangle}{\tau} \right) + \sum \exp \left(\frac{\langle \mathbf{z}^- \rangle}{\tau} \right)} \quad (3)$$

Figure 2: MEGA's architecture. MEGA uses LGA to generate augmented graph, which and the original graph are encoded together. In one iteration, the encoder and projection head are trained by back-propagating $\mathcal{L}_{contrast}$, and in the next iteration, the LGA is trained by performing the second-derivative technique on \mathcal{L}_{MEGA} . The encoder is trained until convergence.

Method



$$\arg \min_{\sigma} \left(\mathcal{L}_{MEGA} \left(g_{\hat{\phi}} \left(f_{\hat{\phi}}(\mathbf{G}) \right), g_{\hat{\phi}} \left(f_{\hat{\phi}}(a_{\sigma}(\mathbf{G})) \right) \right) \right) \quad (4)$$

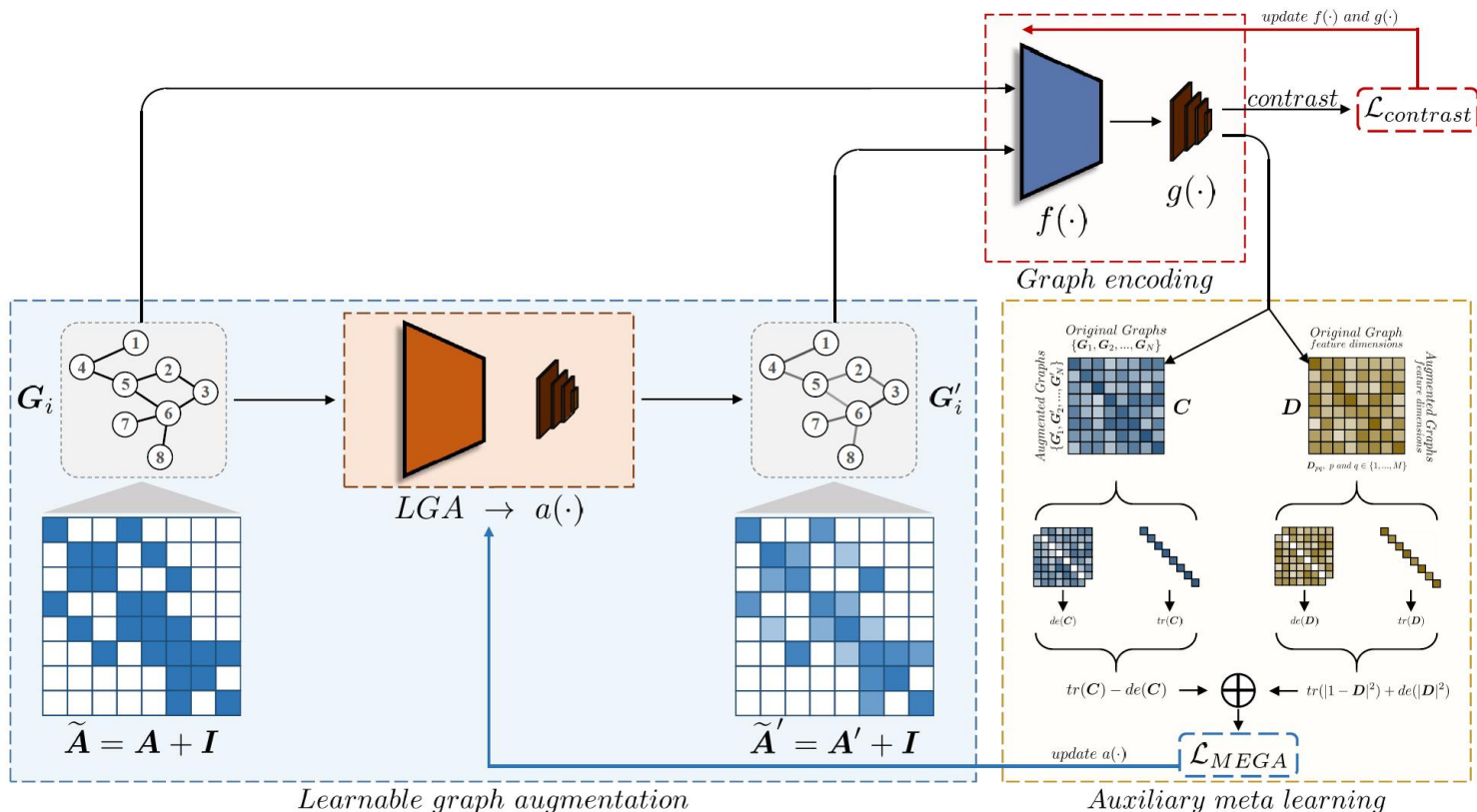
$$\hat{\phi} = \phi - \ell \nabla_{\phi} \left(\mathcal{L}_{contrast} \left(g_{\varphi} \left(f_{\phi}(\mathbf{G}) \right), g_{\varphi} \left(f_{\phi}(\mathbf{G}') \right) \right) \right) \quad (5)$$

$$\hat{\varphi} = \varphi - \ell \nabla_{\varphi} \left(\mathcal{L}_{contrast} \left(g_{\varphi} \left(f_{\phi}(\mathbf{G}) \right), g_{\varphi} \left(f_{\phi}(\mathbf{G}') \right) \right) \right)$$

$$\sigma = \sigma - \ell' \nabla_{\sigma} \left(\mathcal{L}_{MEGA} \left(g_{\hat{\phi}} \left(f_{\hat{\phi}}(\mathbf{G}) \right), g_{\hat{\phi}} \left(f_{\hat{\phi}}(\widehat{\mathbf{G}}') \right) \right) \right) \quad (6)$$

Figure 2: MEGA's architecture. MEGA uses LGA to generate augmented graph, which and the original graph are encoded together. In one iteration, the encoder and projection head are trained by back-propagating $\mathcal{L}_{contrast}$, and in the next iteration, the LGA is trained by performing the second-derivative technique on \mathcal{L}_{MEGA} . The encoder is trained until convergence.

Method



$$\mathcal{L}_{MEGA} = \underbrace{tr(\mathbf{C}) - de(\mathbf{C})}_{instance\ term} + \lambda \underbrace{(tr(|\mathbf{1} - \mathbf{D}|^2) + de(|\mathbf{D}|^2))}_{feature\ term} \quad (7)$$

$$de(M) = \sum_i \sum_{j \neq i} M_{ij} \cdot |\cdot|^2$$

$$|M|^2 = M \times M$$

$$C_{ij} = \frac{z_i \cdot z'_j}{|z_i| \cdot |z'_j|} \quad (8)$$

$$D_{pq} = \frac{\sum_i (z_{i,p} \cdot z'_{i,q})}{\sqrt{\sum_i (z_{i,p})^2} \cdot \sqrt{\sum_i (z'_{i,q})^2}} \quad (9)$$

Figure 2: MEGA's architecture. MEGA uses LGA to generate augmented graph, which and the original graph are encoded together. In one iteration, the encoder and projection head are trained by back-propagating $\mathcal{L}_{contrast}$, and in the next iteration, the LGA is trained by performing the second-derivative technique on \mathcal{L}_{MEGA} . The encoder is trained until convergence.

Experiments

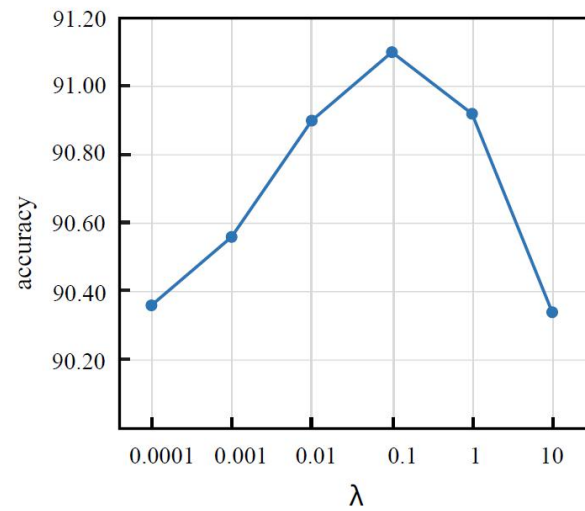
| Method | PROTEINS | MUTAG | DD | COLLAB | RDT-M5K | IMDB-B | IMDB-M |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| GIN RIU | 69.03±0.33 | 87.61±0.39 | 74.22±0.30 | 63.08±0.10 | 27.52±0.61 | 51.86±0.33 | 32.81±0.57 |
| InfoGraph | 72.57±0.65 | 87.71±1.77 | 75.23±0.39 | 70.35±0.64 | 51.11±0.55 | 71.11±0.88 | 48.66±0.67 |
| GraphCL | 72.86±1.01 | 88.29±1.31 | 74.70±0.70 | 71.26±0.55 | 53.05±0.40 | 70.80±0.77 | 48.49±0.63 |
| AD-GCL | 73.46±0.67 | 89.22±1.38 | 74.48±0.62 | 72.90±0.83 | 53.15±0.78 | 71.12±0.98 | 48.56±0.59 |
| MEGA-IL | 73.89±0.62 | 90.34±1.20 | 75.78±0.63 | 73.54±0.82 | 53.16±0.65 | 71.08±0.73 | 49.09±0.79 |
| MEGA | 74.20±0.73 | 91.10±1.34 | 75.56±0.63 | 73.96±0.73 | 54.32±0.79 | 71.95±0.98 | 49.52±0.62 |

Table 1: Performance of classification accuracy on datasets from TU Dataset (Averaged accuracy \pm std. over 10 runs). We highlight the best records in bold.

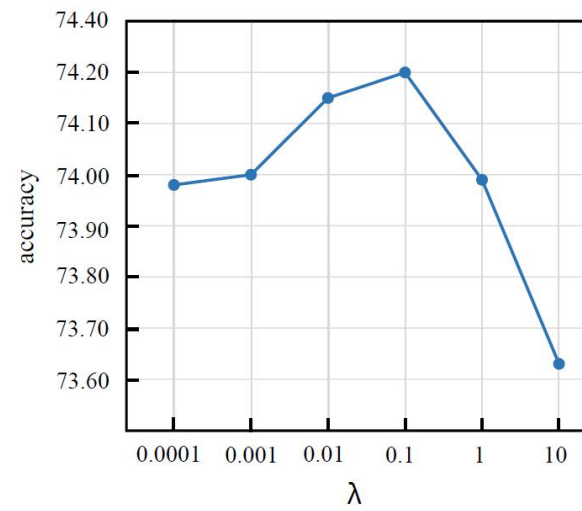
| Method | molesol | mollipo | molbbbp | moltox21 | molsider |
|----------------|---------------------------------------|--------------------|--|-------------------|-------------------|
| | Regression tasks (RMSE \downarrow) | | Classification tasks (ROC-AUC % \uparrow) | | |
| GIN RIU | 1.706±0.180 | 1.075±0.022 | 64.48±2.46 | 71.53±0.74 | 62.29±1.12 |
| InfoGraph | 1.344±0.178 | 1.005±0.023 | 66.33±2.79 | 69.74±0.57 | 60.54±0.90 |
| GraphCL | 1.272±0.089 | 0.910±0.016 | 68.22±1.89 | 72.40±1.01 | 61.76±1.11 |
| AD-GCL | 1.270±0.092 | 0.926±0.037 | 68.26±1.32 | 71.08±0.93 | 61.83±1.14 |
| MEGA-IL | 1.153±0.103 | 0.852±0.022 | 68.34±1.38 | 72.08±0.82 | 63.37±0.87 |
| MEGA | 1.121±0.092 | 0.831±0.018 | 69.71±1.56 | 72.45±0.67 | 62.92±0.76 |

Table 2: Performance of chemical molecules property prediction in OGB datasets. There are two kinds of tasks, regression tasks and classification tasks. We highlight the best records in bold.

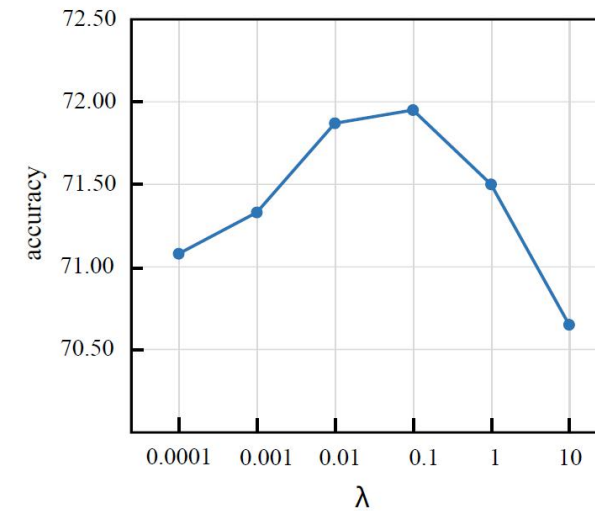
Experiments



(a) MUTAG



(b) PROTEINS



(c) IMDB-B

Figure 3: Results of MEGA's performance with a range of factor λ . We perform MEGA on three benchmark datasets: MUTAG, PROTEINS, and IMDB-B. The abscissa axis represents the value of λ , and the ordinate axis represents the accuracies.

Experiments

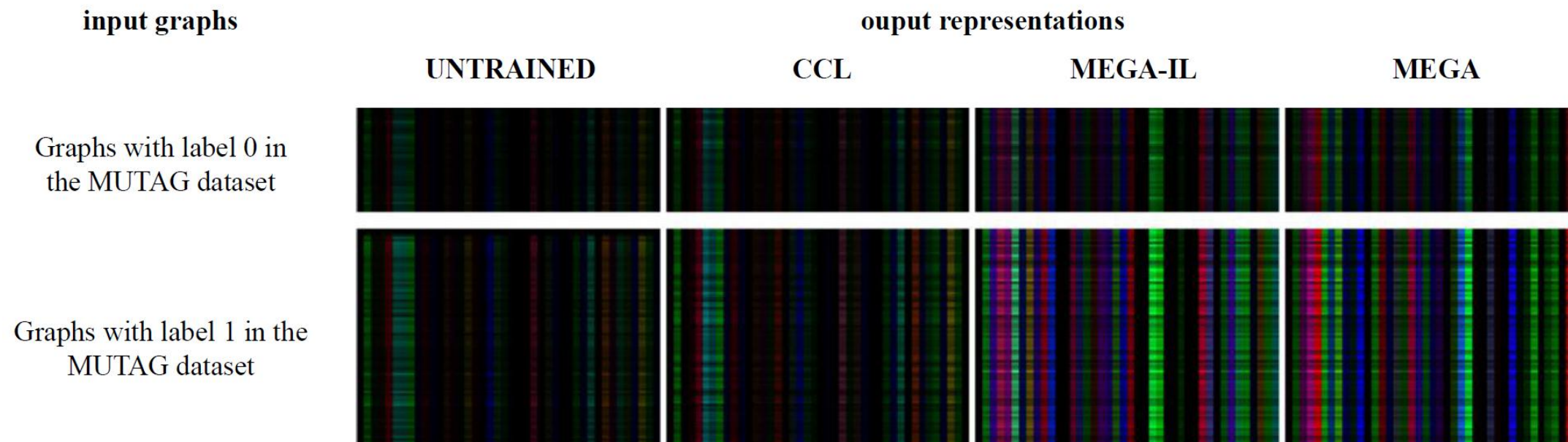


Figure 4: This figure shows the visualized output graph features on the MUTAG dataset. The graph features are projected into a color image in RGB format, where different colors represent different types of features. The abscissa axis represents the output feature dimensions of compared methods, and the ordinate axis represents graphs of different classes.



Thanks