Adversarial Graph Contrastive Learning with Information Regularization

(WWW, 2022)

paper

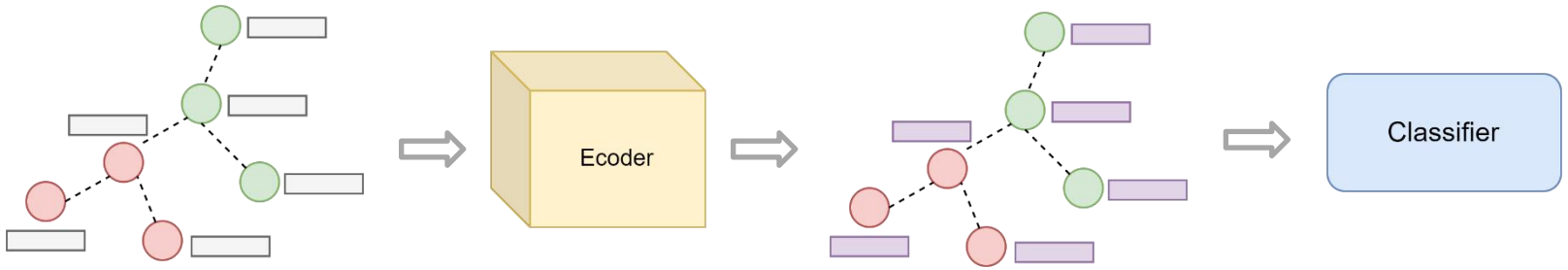
Graph Representation Learning

Given an attributed graph: $G = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, Where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$,

$\mathbf{X} \in \mathbb{R}^{n \times d}$, Adjacency matrix can be defined as : $\mathbf{A} \in \{0, 1\}^{n \times n}$

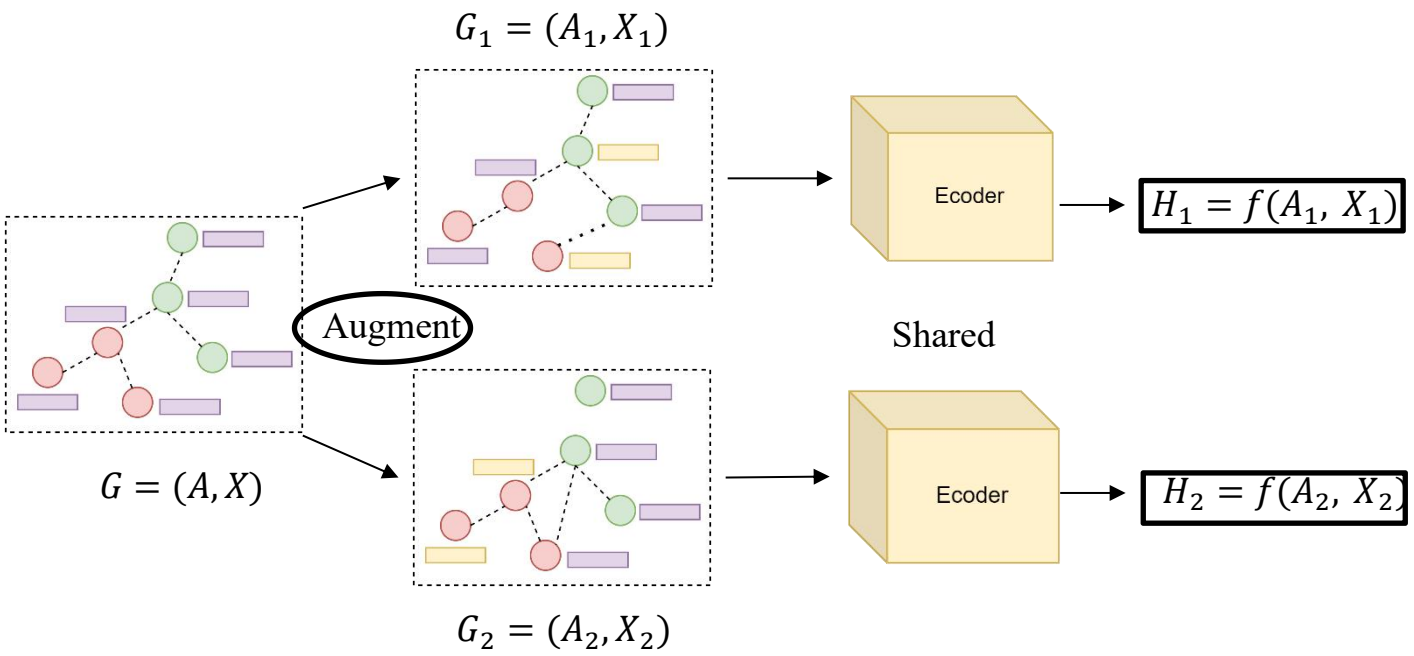
Objective is to learn an encoder $f : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d'}$,

$\mathbf{H} = f(\mathbf{A}, \mathbf{X})$, where $\mathbf{H}[i, :] \in \mathbb{R}^{d'}$ is the embedding of node v_i.



(A, X) f $H = f(A, X)$

Contrastive Learning for GRL



$$L_k = -\log\left(\frac{e^{g(\mathbf{x},\mathbf{x}^+)}}{e^{g(\mathbf{x},\mathbf{x}^+)} + \sum_{i=1}^{k} e^{g(\mathbf{x},\mathbf{x}_i^-)}}\right).$$

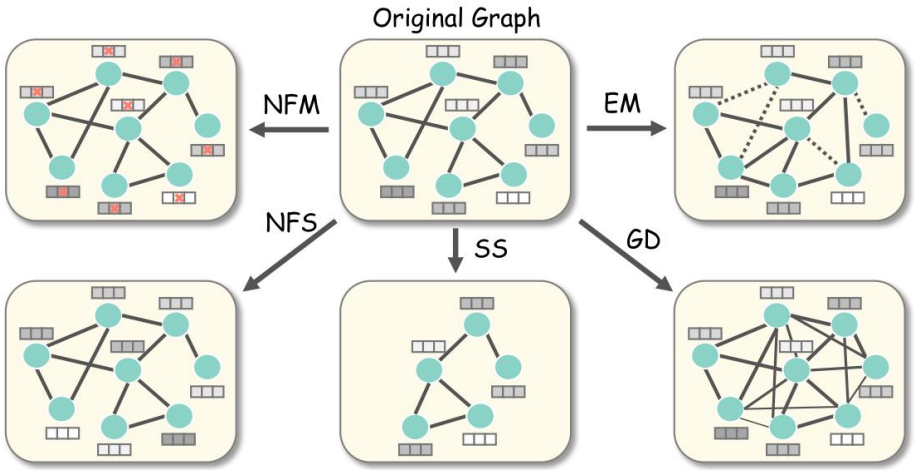
$$L_{\text{con}}(G_1, G_2) = \frac{1}{2n}\sum_{i=1}^{n}(l(\mathbf{u}_i, \mathbf{v}_i) + l(\mathbf{v}_i, \mathbf{u}_i)),$$

$$l(\mathbf{u}_i, \mathbf{v}_i) = -\log \frac{e^{\theta(\mathbf{u}_i,\mathbf{v}_i)/\tau}}{e^{\theta(\mathbf{u}_i,\mathbf{v}_i)/\tau} + \sum_{j\neq i} e^{\theta(\mathbf{u}_i,\mathbf{v}_j)/\tau} + \sum_{j\neq i} e^{\theta(\mathbf{u}_i,\mathbf{u}_j)/\tau}},$$

Motivation

Limitation of Existing Augment methods for GCL:

- Uncontrollable: The data augmentation on the graph, could be either too similar to or totally different from the original graph.
- Vulnerable : GNNs are known to be vulnerable to adversarial attacks



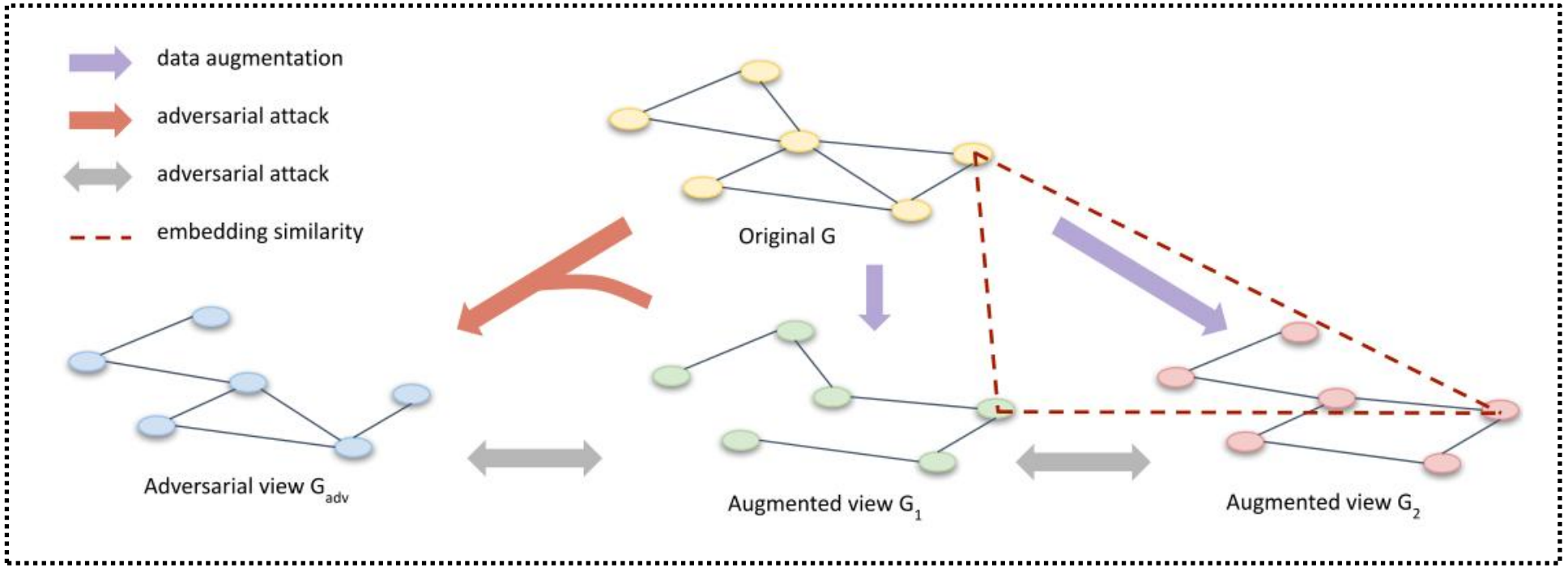
How to generate a new graph that is hard enough for the model to discriminate from the original one, and in the meanwhile also maintains the desired properties?

Introduce adversarial training into GCL.

On the one hand, since the perturbation is under the constraint, the adversarial sample still stays close enough to the original one.

On the other hand, the adversarial attack makes sure the adversarial sample is hard to discriminate from the other view by increasing the contrastive loss.

Overview of ARIEL



Adversarial Training

Projected Gradient Descent Attack

$$\Delta_t = \Pi_{\|\Delta\|_\infty \le \delta}(\Delta_{t-1} + \eta \cdot \mathrm{sgn}(\nabla_{\Delta_{t-1}} L(\mathbf{Z} + \Delta_{t-1})), \qquad (7)$$

where $\boldsymbol{L}()$ is loss function of input matrix $\boldsymbol{Z}$, $\boldsymbol{\Delta_t}$ is the perturbation matrix. $\mathbf{sgn}()$ is sign function, $\boldsymbol{\eta}$ is step size.

Adversarial Objective: $G_{\mathrm{adv}} = \arg\max_{G'} L_{\mathrm{con}}(G_1, G'), \qquad (8)$

where $G' = (A', X')$ is generated from the original graph G and satisfy:

$$\sum_{i,j} |\mathbf{A}'[i,j] - \mathbf{A}[i,j]| \le \Delta_{\mathbf{A}}, \qquad (9)$$

$$\sum_{i,j} |\mathbf{X}'[i,j] - \mathbf{X}[i,j]| \le \Delta_{\mathbf{X}}. \qquad (10)$$

Adversarial Training

Attack on Structure:

$$\mathbf{A}_{adv} = \mathbf{A} + \mathbf{C} \circ \mathbf{L}_{\mathbf{A}}, \qquad (11)$$

$$\mathbf{C} = \bar{\mathbf{A}} - \mathbf{A}, \qquad (12)$$

Where $\overline{\boldsymbol{A}} = \mathbf{1}_{n\times n} - \mathbf{I}_n - \mathbf{A}$, $\mathbf{L}_{\boldsymbol{A}} \in \{0,1\}^{n \times n}$

$\boldsymbol{L_A}$ is relaxed to its convex hull $\tilde{\boldsymbol{L}}_{\boldsymbol{A}} \in [0,1]^{n \times n}$

Attack on Feature matrix:

$$\mathbf{X}_{adv} = \mathbf{X} + \mathbf{L}_{\mathbf{X}}, \qquad (13)$$

Where $\boldsymbol{L_X} \in \mathbb{R}^{n\times d}$ is the perturbation on the feature matrix.

Adversarial Training

Update for adversarial perturbation:

$$\tilde{\mathbf{L}}_{\mathbf{A}}^{(t)} = \Pi_{\mathcal{S}_{\mathrm{A}}}[\tilde{\mathbf{L}}_{\mathbf{A}}^{(t-1)} + \alpha \cdot \mathbf{G}_{\mathbf{A}}^{(t)}], \tag{14}$$

$$\mathbf{L}_{\mathbf{X}}^{(t)} = \Pi_{\mathcal{S}_{\mathrm{X}}}[\mathbf{L}_{\mathbf{X}}^{(t-1)} + \beta \cdot \mathrm{sgn}(\mathbf{G}_{\mathbf{X}}^{(t)})], \tag{15}$$

$$\mathbf{G}_{\mathbf{A}}^{(t)} = \nabla_{\tilde{\mathbf{L}}_{\mathbf{A}}^{(t-1)}} L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}^{(t-1)}), \tag{16}$$

$$\mathbf{G}_{\mathbf{X}}^{(t)} = \nabla_{\mathbf{L}_{\mathbf{X}}^{(t-1)}} L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}^{(t-1)}), \tag{17}$$

$$G_{\mathrm{adv}}^{(t-1)} = \{\mathbf{A} + \mathbf{C} \circ \tilde{\mathbf{L}}_{\mathbf{A}}^{(t-1)}, \mathbf{X} + \mathbf{L}_{\mathbf{X}}^{(t-1)}\}$$

$$\mathcal{S}_{\mathrm{A}} = \{\tilde{\mathbf{L}}_{\mathbf{A}} | \sum_{i,j} \tilde{\mathbf{L}}_{\mathbf{A}} \leq \Delta_{\mathbf{A}}, \tilde{\mathbf{L}}_{\mathbf{A}} \in [0,1]^{n \times n}\}$$

$$\mathcal{S}_{\mathrm{X}} = \{\mathbf{L}_{\mathbf{X}} | \|\mathbf{L}_{\mathbf{X}}\|_{\infty} \leq \delta_{\mathbf{X}}, \mathbf{L}_{\mathbf{X}} \in \mathbb{R}^{n \times d}\}$$

The projection operation $\Pi_{\mathcal{S}_{\mathrm{X}}}$ simply clips L_X into the range $[-\delta_{\mathrm{X}}, \delta_{\mathrm{X}}]$

The projection operation $\Pi_{\mathcal{S}_{\mathrm{A}}}$ is calculated as

$$\Pi_{\mathcal{S}_{\mathrm{A}}}(\mathbf{Z}) = \begin{cases} P_{[0,1]}[\mathbf{Z} - \mu \mathbf{1}_{n \times n}], & \text{if } \mu > 0, \text{ and } \sum_{i,j} P_{[0,1]}[\mathbf{Z} - \mu \mathbf{1}_{n \times n}] = \Delta_{\mathbf{A}}, \\ P_{[0,1]}[\mathbf{Z}], & \text{if } \sum_{i,j} P_{[0,1]}[\mathbf{Z}] \leq \Delta_{\mathbf{A}}, \end{cases} \tag{18}$$

Adversarial Graph Contrastive Learning

$$L(G_1, G_2, G_{\mathrm{adv}}) = L_{\mathrm{con}}(G_1, G_2) + \epsilon_1 L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}), \tag{19}$$

Information Regularization

THEOREM3.1. *For two graph views G_1 and G_2 independently transformed from the graph G, the density ratio of their node embeddings H_1 and H_2 should satisfy $g(H_2[i,:], H_1[i,:]) \leq g(H_2[i,:], H[i,:])$ and $g(H_1[i,:], H_2[i,:]) \leq g(H_1[i,:], H[i,:])$, where H is the node embeddings of the original graph.*

The node embeddings of two views H_1, H_2 and of original graph H satisfy the Markov relation as follow:

$$H_1 \rightarrow H \rightarrow H_2$$

$$H_1[i,:] \rightarrow H[i,:] \rightarrow H_2[i,:]$$

Then, we get:

$$p(\mathbf{H}_2[i,:]|\mathbf{H}_1[i,:]) = p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:])p(\mathbf{H}[i,:]|\mathbf{H}_1[i,:]) \quad (24)$$

$$\leq p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:]), \quad (25)$$

$$\frac{p(\mathbf{H}_2[i,:]|\mathbf{H}_1[i,:])}{p(\mathbf{H}_2[i,:])} \leq \frac{p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:])}{p(\mathbf{H}_2[i,:])}. \quad (26)$$

Since $g(\mathbf{a},\mathbf{b}) \propto \frac{p(\mathbf{a}|\mathbf{b})}{p(\mathbf{a})}$, then $g(\mathbf{H}_2[i,:], \mathbf{H}_1[i,:]) \leq g(\mathbf{H}_2[i,:], \mathbf{H}[i,:])$.

Information Regularization

$$g(H_2[i,:],\ H_1[i,:]) \leq g(H_2[i,:],\ H[i,:])$$

$$g(H_1[i,:],\ H_2[i,:]) \leq g(H_1[i,:],\ H[i,:])$$

According to the previous definition, $g(a,\ b) = e^{\theta(a,b)/\tau}$, simply replace $g(\cdot,\ \cdot)$ with $\theta(\cdot,\ \cdot)$, then combine two upper bounds into one:

$$2 \cdot \theta(\mathbf{H}_1[i,:], \mathbf{H}_2[i,:]) \leq \theta(\mathbf{H}_2[i,:], \mathbf{H}[i,:]) + \theta(\mathbf{H}_1[i,:], \mathbf{H}[i,:]).$$

$$d_i = 2 \cdot \theta(\mathbf{H}_1[i,:], \mathbf{H}_2[i,:]) - (\theta(\mathbf{H}_2[i,:], \mathbf{H}[i,:]) + \theta(\mathbf{H}_1[i,:], \mathbf{H}[i,:]))$$

$$L_I(G_1, G_2, G) = \frac{1}{n} \sum_{i=1}^{n} \max\{d_i, 0\}.$$

$$L(G_1, G_2, G_{\text{adv}}) = L_{\text{con}}(G_1, G_2) + \epsilon_1 L_{\text{con}}(G_1, G_{\text{adv}}) + \epsilon_2 L_I(G_1, G_2, G), \qquad (30)$$

Training pseudocode

Algorithm 1 Algorithm of ARIEL

Input data: Graph $G = (\mathbf{A}, \mathbf{X})$
Input parameters: α, β, $\Delta_{\mathbf{A}}$, $\delta_{\mathbf{X}}$, ϵ_1, ϵ_2, γ and T
Randomly initialize the graph encoder f
for iteration $k = 0, 1, \cdots$ **do**
  Sample a subgraph G_s from G
  Generate two views G_1 and G_2 from G_s
  Generate the adversarial view G_{adv} according to Equations (15), (14)
  Update model f to minimize $L(G_1, G_2, G_{\text{adv}})$ in Equation (30)
  if $(k+1) \mod T = 0$ **then**
    Update $\epsilon_1 \leftarrow \gamma * \epsilon_1$
  end if
end for
return: Node embedding matrix $\mathbf{H} = f(\mathbf{A}, \mathbf{X})$

Experiments

RQ1. How effective is the proposed ArieL in comparison with previous graph contrastive learning methods on the node classification task?

Method	Cora	CiteSeer	Amazon-Computers	Amazon-Photo	Coauthor-CS	Coauthor-Physics
GCN	84.14 ± 0.68	69.02 ± 0.94	88.03 ± 1.41	92.65 ± 0.71	92.77 ± 0.19	95.76 ± 0.11
GAT	83.18 ± 1.17	69.48 ± 1.04	85.52 ± 2.05	91.35 ± 1.70	90.47 ± 0.35	94.82 ± 0.21
DeepWalk+features	79.82 ± 0.85	67.14 ± 0.81	86.23 ± 0.37	90.45 ± 0.45	85.02 ± 0.44	94.57 ± 0.20
DGI	84.24 ± 0.75	69.12 ± 1.29	88.78 ± 0.43	92.57 ± 0.23	92.26 ± 0.12	95.38 ± 0.07
GMI	82.43 ± 0.90	70.14 ± 1.00	83.57 ± 0.40	88.04 ± 0.59	OOM	OOM
MVGRL	**84.39 ± 0.77**	69.85 ± 1.54	89.02 ± 0.21	92.92 ± 0.25	92.22 ± 0.22	95.49 ± 0.17
GRACE	83.40 ± 1.08	69.47 ± 1.36	87.77 ± 0.34	92.62 ± 0.25	93.06 ± 0.08	95.64 ± 0.08
GCA-DE	82.57 ± 0.87	72.11 ± 0.98	88.10 ± 0.33	92.87 ± 0.27	93.08 ± 0.18	95.62 ± 0.13
GCA-PR	82.54 ± 0.87	72.16 ± 0.73	88.18 ± 0.39	92.85 ± 0.34	93.09 ± 0.15	95.58 ± 0.12
GCA-EV	81.80 ± 0.92	67.07 ± 0.79	87.95 ± 0.43	92.63 ± 0.33	93.06 ± 0.14	95.64 ± 0.08
ARIEL	84.28 ± 0.96	**72.74 ± 1.10**	**91.13 ± 0.34**	**94.01 ± 0.23**	**93.83 ± 0.14**	**95.98 ± 0.05**

Table 2: Node classification accuracy in percentage on six real-world datasets. We bold the results with the best mean accuracy. The methods above the line are the supervised ones, and the ones below the line are unsupervised. OOM stands for Out-of-Memory on our 32G GPUs.

Experiments

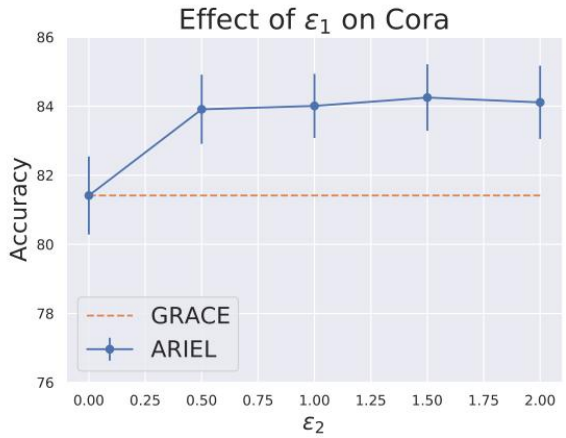
RQ2. To what extent does ArieL gain robustness over the attacked graph?

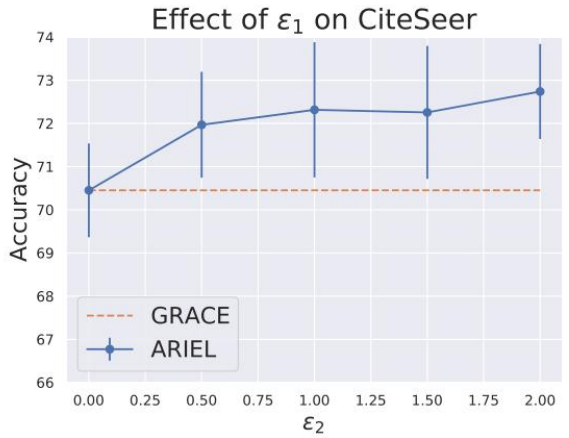
Method	Cora	CiteSeer	Amazon-Computers	Amazon-Photos	Coauthor-CS	Coauthor-Physics
GCN	80.03 ± 0.91	62.98 ± 1.20	84.10 ± 1.05	91.72 ± 0.94	80.32 ± 0.59	87.47 ± 0.38
GAT	79.49 ± 1.29	63.30 ± 1.11	81.60 ± 1.59	90.66 ± 1.62	77.75 ± 0.80	86.65 ± 0.41
DeepWalk+features	74.12 ± 1.02	63.20 ± 0.80	79.08 ± 0.67	88.06 ± 0.41	49.30 ± 1.23	79.26 ± 1.38
DGI	80.84 ± 0.82	64.25 ± 0.96	83.36 ± 0.55	91.27 ± 0.29	78.73 ± 0.50	85.88 ± 0.37
GMI	79.17 ± 0.76	65.37 ± 1.03	77.42 ± 0.59	89.44 ± 0.47	80.92 ± 0.64	87.72 ± 0.45
MVGRL	**80.90 ± 0.75**	64.81 ± 1.53	83.76 ± 0.69	91.76 ± 0.44	79.49 ± 0.75	86.98 ± 0.61
GRACE	78.55 ± 0.81	63.17 ± 1.81	84.74 ± 1.13	91.26 ± 0.37	80.61 ± 0.63	85.71 ± 0.38
GCA	76.79 ± 0.97	64.89 ± 1.33	85.05 ± 0.51	91.71 ± 0.34	82.72 ± 0.58	89.00 ± 0.31
ARIEL	80.33 ± 1.25	**69.13 ± 0.94**	**88.61 ± 0.46**	**92.99 ± 0.21**	**84.43 ± 0.59**	**89.09 ± 0.31**

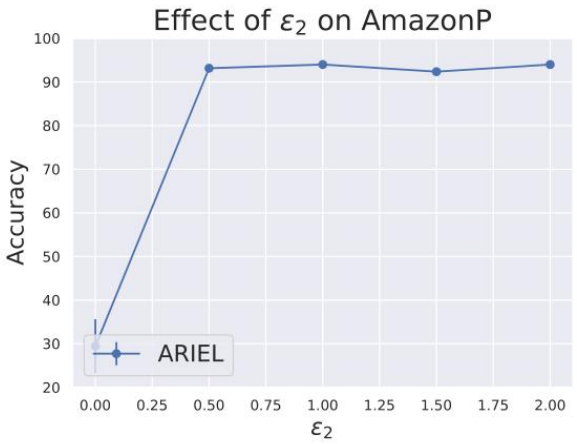
Table 3: Node classification accuracy in percentage on the graphs under Metattack, where subgraphs of Amazon-Computers, Amazon-Photo, Coauthor-CS and Coauthor-Physics are used for attack and their results are not directly comparable to those in Table 2. We bold the results with the best mean accuracy. GCA is evaluated on its best variant on each clean graph.

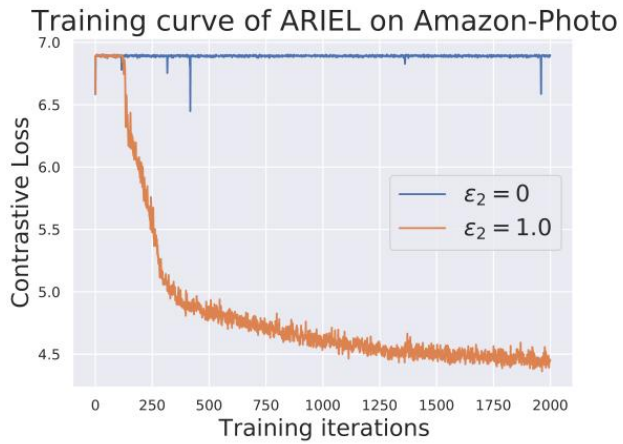
Experiments

RQ3. How does each part of ArieL contribute to its performance?









Thanks