Robust Graph Attention for Graph Adversarial Attacks: An Information Bottleneck Inspired Approach

Anonymous authors
Paper under double-blind review

Abstract

Graph Neural Networks (GNNs) have shown exceptional performance in learning node representations for node-level tasks such as node classification. However, traditional message-passing mechanisms solely based on graph structure in GNNs make them vulnerable to adversarial attacks. Attention-based GNNs have been employed to improve the robustness of GNNs due to their ability to selectively emphasize informative signals over noisy or less relevant ones. In this paper, we propose a novel graph attention method termed Robust Graph Attention inspired by Information Bottleneck, or RGA-IB, which explicitly minimizes the IB loss of a multi-layer GNN through a carefully designed graph attention mechanism. In contrast to existing IB-based graph learning methods that rely on unrealistic Gaussian distributional or local-dependence assumptions due to optimizing variational upper bounds of the IB loss, RGA-IB introduces a novel robust graph attention mechanism that directly reduces the original IB loss, thereby avoiding such restrictive assumptions. Extensive experimental results on semi-supervised node classification under various graph adversarial attacks show that GNNs equipped with RGA-IB exhibit lower IB loss, which indicates better adherence to the IB principle, and show significantly improved node classification accuracy under graph adversarial attacks compared to existing robust GNNs. The code of RGA-IB is available at https://anonymous.4open.science/status/RGA-IB.

1 Introduction

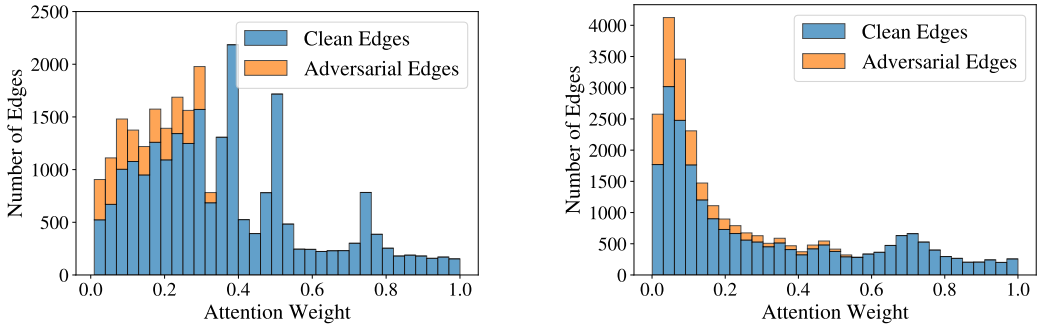
As generalizations of Deep Neural Networks (DNNs), Graph Neural Networks (GNNs) have emerged as popular tools for machine learning on graph-structured data (Kipf & Welling, 2017; Bruna et al., 2014; Hamilton et al., 2017a; Xu et al., 2019b). Most prevailing GNNs (Kipf & Welling, 2017; Hamilton et al., 2017a) follow the message-passing scheme and learn the representation of each node by iteratively transforming and propagating the information within its neighborhood. Benefiting from such merits, GNNs show dominant performance on various graph learning tasks, such as node classification (Ding et al., 2023), link prediction (Zhang & Chen, 2018), and graph classification (Zeng & Xie, 2020). Among different graph learning tasks, semi-supervised node classification, aiming at predicting the labels for a set of unlabeled nodes in a partially labeled attributed graph (Kipf & Welling, 2017), benefits the most from the message-passing scheme as it allows information from labeled nodes to propagate and influence the predictions for unlabeled nodes. However, the message-passing scheme also makes GNNs vulnerable to adversarial attacks (Zügner et al., 2018). Recent works (Zügner & Günnemann, 2019; Sun et al., 2020) have shown that by carefully perturbing only a small number of edges or nodes in the graph, adversarial attacks can catastrophically reduce the performance of GNNs in predicting the labels on either all unlabeled nodes (Zügner & Günnemann, 2019) or only a small targeted set of unlabeled nodes (Zügner et al., 2018) in the semi-supervised node classification task. Some efforts have been devoted to improving the robustness of GNNs by adversarial training (Li et al., 2022a), graph pre-processing (Wu et al., 2019; Entezari et al., 2020; Jin et al., 2020; Lei et al., 2022), and model robustification (Zhao et al., 2023; Song et al., 2022; Jia et al., 2023).

Adaptively assigning weights to the neighbors of a node in the message-passing scheme, graph attention modules (Veličković et al., 2018; Zhang & Zitnik, 2020; Yang et al., 2021b; Feng et al., 2021; Fountoulakis et al., 2023; Wu et al., 2023a) have recently drawn increasing attention in improving the robustness of

GNNs among model robustification methods. Early graph attention methods, such as GAT (Veličković et al., 2018), assign weights to edges in the graph by either estimating the uncertainties of edges (Feng et al., 2021; Yang et al., 2021b) or the similarities between neighboring nodes (Veličković et al., 2018; Zhang & Zitnik, 2020). Recent advancements of Transformer-based GNNs (Wu et al., 2023b; Fountoulakis et al., 2023; Wu et al., 2023a) such as Difformer (Wu et al., 2023a) have introduced mechanisms that extend beyond immediate neighbors to capture the dense correlations among all the nodes in the graph. By focusing on the most relevant and reliable information within the graph structure, graph attention networks dynamically modulate the influence of the neighbors of each node, improving the resilience of the GNNs against malicious graph structures and features. For example, RGCN (Zhu et al., 2019) and GAR (Fountoulakis et al., 2023) design novel graph attention modules to improve the robustness of GNNs against graph adversarial attacks. However, the attention modules in existing attention-based GNNs are often empirically designed and lack theoretical support.

Understanding Robust Graph Attention from the Information Bottleneck (IB) perspective. In this work, we understand the robust graph attention mechanism from the Information Bottleneck (IB) principle. The IB principle (Tishby et al., 2000) encourages maximizing the mutual information between the node representation and input features while minimizing the mutual information between the node representation and class labels. Let X be the random variable representing the input features and Z be the random variable representing the node representations to be learned by the GNNs. Let Y be the ground truth training labels for the node classification task. The IB principle is to increase the mutual information between Z and Y while decreasing the mutual information between Z and X, that is, the IB loss $\mathrm{IB}(Z, X, Y) = I(Z, X) - I(Z, Y)$. Lower IB loss indicates better adherence to the IB principle. As a result, learned GNNs adhering to the IB principle naturally avoid overfitting to the inputs and become more robust to adversarial attacks in the input graph data. Recent works find that minimizing the IB loss can improve the adversarial robustness of DNNs (Wang et al., 2021; Kuang et al., 2023) and GNNs (Wu et al., 2020). We provide new insights into the underlying connection between the IB principle (Tishby et al., 2000) and the robustness of GNNs with graph attention modules which has not been revealed in the graph learning literature. The graph attention operation selectively aggregates informative signals over noisy or less relevant ones. Such selective focus leads to more relevant and compact node representations which correlate less to the input graph data while aligning better with the training class labels, thereby adhering better to the IB principle with a lower IB loss. This targeted refinement of node representations through the attention mechanism results in an efficient compression of information, which enhances the generalization capabilities of the GNNs with graph attention.

Limitation of Existing Works about IB-based Robust GNNs. Inspired by the IB principle, GIB (Wu et al., 2020) proposes the first framework for learning minimal sufficient node representations for node classification. However, GIB relies on an unrealistic local-dependence assumption, which assumes that node features learned by a multi-layer GNN are independent of features from nodes beyond two-hop neighborhoods. In practical GNNs, multi-layer message passing recursively aggregates information from increasingly distant nodes, making such independence assumptions inconsistent with the message passing framework of GNNs. For example, features of nodes beyond two-hop neighborhoods aggregated with more than three layers are dependent. For example, in a three-layer GNN, the representation of a node at the third layer implicitly depends on the features of all nodes within its three-hop neighborhood. Following GIB, the IB principle has been widely applied to a variety of graph learning tasks to learn discriminative node representations (Sun et al., 2022; Fu et al., 2025; Dong & Sun, 2024; Wang et al., 2024; Dai et al., 2023b; Yuan et al., 2024; Wang et al., 2025). However, all these methods, including graph attention methods (Dong & Sun, 2024; Wang et al., 2025), graph structure learning methods (Sun et al., 2022; Fu et al., 2025), and graph adversarial defense methods (Wang et al., 2024; Dai et al., 2023b; Yuan et al., 2024), rely on a variational upper bound based on VIB (Alemi et al., 2017) as part of their training objectives. The derivation of such variational upper bound imposes an unrealistic Gaussian assumption on the conditional distribution $p(Z|X)$ or $p(Z|X, A)$, where Z denotes the learned node representation, X and A denote the input node attributes and adjacency matrix. Such Gaussian distributional assumption may not hold on real graph data and generic GNNs. Furthermore, the training of RGIB (Wang et al., 2024) relies on the availability of clean subgraphs. However, clean graphs are often unavailable for certain attack scenarios in practice, such as poisoning attacks (Zügner &



(a) Cora (b) Pubmed

Figure 1: Illustration of the distribution of attention weights assigned to clean and adversarial edges on graphs attacked by Metattack (Zügner & Günnemann, 2019) with a perturbation rate of 25% on Cora and Pubmed. The histograms illustrate that RGA-IB usually assigns higher attention weights to clean edges and lower attention weights to adversarial edges, thereby mitigating the negative impact of adversarial perturbations. For instance, 99.2% of the edges with attention weights greater than 0.8 are clean edges, and 82.6% of adversarial edges receive attention weights below 0.2 on Pubmed. Such an advantage stems from the robust graph attention mechanism in RGA-IB, which reduces the IB loss and thus better adheres to the IB principle. As a result, higher attention weights are assigned to more informative and clean edges that contribute more to the node classification task, leading to improved performance of node classification under such attacks.

Günnemann, 2019; Xu et al., 2019a), where adversaries corrupt the training data prior to the training and defenders only observe poisoned graphs.

In contrast to existing IB-based graph learning methods, our RGA-IB introduces a novel robust graph attention method which reduces the original IB loss without relying on any distributional assumptions or local-dependence assumptions. Our novel graph attention reduces the original IB loss instead of its upper bound so as to avoid all the aforementioned assumptions in existing works. Moreover, RGA-IB can be trained directly on attacked graphs, without access to clean subgraphs during training.

1.1 Contributions

Our contributions are presented as follows.

First, we introduce a novel graph attention method termed Robust Graph Attention inspired by Information Bottleneck, or RGA-IB. RGA-IB is motivated by the connection between the principle of Information Bottleneck (IB) and the robustness of graph attention methods against adversarial attacks. In contrast to existing IB-based graph learning methods that rely on unrealistic Gaussian distributional or local-dependence assumptions, RGA-IB introduces a novel robust graph attention mechanism that directly reduces the original IB objective without imposing such assumptions. Unlike prior approaches that optimize an upper bound for the IB loss, thereby necessitating restrictive assumptions, our method reduces the IB loss itself instead of its upper bound, effectively avoiding these limitations. Moreover, RGA-IB can be trained directly on attacked graphs, eliminating the need for access to clean subgraphs during training.

Second, to reduce the IB loss of a GNN with graph attention, we view the GNN with multiple RGA-IB layers as an iterative process for the reduction of the IB loss by gradient descent, and each RGA-IB layer simulates one-step gradient descent on the IB loss. Inspired by this understanding, the attention weight matrix at the current layer is generated from the attention weight matrix at the previous layers, and the input node features at the current layer, following the gradient formula in Equation (1) in Section 3.2. As a result, the RGA-IB network with RGA-IB layers enjoys reduced IB loss compared to existing graph attention methods, which is evidenced in Table 5 in Section 4.3. Benefiting from the better adherence to the IB principle by reducing the IB loss, RGA-IB assigns higher attention weights to clean edges and lower attention weights to adversarial edges as illustrated in Figure 1, which effectively mitigates the negative impact of the

adversarial attacks. Extensive evaluation results on public graph benchmarks for semi-supervised node classification under different categories of graph adversarial attacks in Table 1-3 in Section 4.2 demonstrate the effectiveness of explicitly reducing IB loss with RGA-IB for improving robustness. Experiments in Section C.2 and Section C.3 of the appendix further demonstrate the effectiveness of RGA-IB on large-scale graph datasets and heterophilic graph datasets. As illustrated in Table 4 in Section 4.3, RGA-IB effectively reduces the IB loss to a lower level at deeper layers.

It is noted that although a portion of clean edges receive low attention weights as illustrated in Figure 1, message-propagation-based feature aggregation with graph attention is primarily performed on edges with large attention weights, which are predominantly clean. As a result, adversarial edges with low attention weights have negligible influence on feature aggregation, leading to robust node representations under attack.

2 Related Works

2.1 Graph Adversarial Attacks and Defense

Despite the success of Graph Neural Networks (GNNs) in various applications (Kipf & Welling, 2017; Zhang & Chen, 2018) on graph-structured data, recent works have shown that GNNs are vulnerable to adversarial attacks. Graph adversarial attacks (Dai et al., 2022) aim to degrade the performance of GNNs by injecting deliberate perturbations into the graph dataset. Based on whether the goal of the attacker is to reduce the performance of the GNN on a set of target instances or reduce the overall performance of the GNN model on the targeted datasets, threat models can also be categorized as: (1) targeted attack (Zügner et al., 2018), which aims to fool a GNN model into misclassifying a set of target nodes, and (2) untargeted attack (Zügner & Günnemann, 2019; Sun et al., 2020), which aims to reduce the overall performance of the GNN model on the target dataset. To address the vulnerability of GNNs to adversarial attacks, various robust learning methods have been proposed, which can be categorized into three classes: Adversarial Training, Graph Processing, and Model Robustification. Adversarial training methods (Li et al., 2022a) train robust models on a training set augmented with handcrafted adversarial samples. Graph processing methods (Wu et al., 2019; Entezari et al., 2020; Jin et al., 2020; Lei et al., 2022) aim to purify the graph data and remove adversarial perturbations. Model robustification methods (Xie et al., 2023; Chamberlain et al., 2021; Rusch et al., 2022; Song et al., 2022; Zhao et al., 2023; Jia et al., 2023; Abbahaddou et al., 2024) refine the GNN models to prepare against potential adversarial threats. Recent adversarial defense studies show that perturbations predominantly reside in high-rank residual components while clean signals concentrate in low-rank subspaces, motivating low-rank projection of representations or low-rank constraints on model weights as effective strategies for enhancing robustness (Awasthi et al., 2020; 2021; Xu et al., 2024; Savostianova et al., 2023). In the graph domain, prior low-rank learning methods predominantly focus on low-rank approximation of the adjacency matrix (Wu et al., 2022b; Entezari et al., 2020; Deng et al., 2022; Fan et al., 2025; Zhou et al., 2025; Zhang et al., 2025) for predictive tasks. Recent studies have explored graph purification (He et al., 2025; Luo et al., 2025; Zhao et al., 2025) for graph adversarial defense. However, existing diffusion-based graph purification methods rely on paired attacked and clean graphs to learn the denoising process. In practice, clean graphs are often unavailable for certain attacks, such as poisoning attacks, where adversaries corrupt the training data prior to learning. Consequently, defenders only observe poisoned graphs. Recently, GAR (Fountoulakis et al., 2023) proves that graph attention modules exhibit strictly better robustness against structural noise in the graph over both the graph convolution and linear classifier.

2.2 Information Bottleneck-Based Robust Graph Neural Networks

The Information Bottleneck (IB) (Tishby et al., 2000) principle aims to learn latent representations of data that retain information relevant to the target task while minimizing redundant information from the input. Deep VIB (Alemi et al., 2017) first introduces the IB principle as the objective for the training of deep neural networks. Inspired by the IB principle, (Lai et al., 2021) proposes a spatial attention module that reduces the IB loss on the attention-modulated representation. Following that, (Zhou et al., 2022) proves that self-attention can be interpreted as an iterative optimization of the IB objective. To justify the benefits of minimizing IB loss in deep learning, recent works (Amjad & Geiger, 2020; Kawaguchi et al., 2023)

theoretically prove that controlling IB loss is one way to control generalization errors in deep learning. In addition, recent works also find that (Wang et al., 2021; Kuang et al., 2023) minimizing the IB loss can improve the adversarial robustness of DNNs.

Information Bottleneck for GNNs. More recently, the IB principle has been successfully adapted to different graph learning tasks (Wu et al., 2020; Wang et al., 2024; Dai et al., 2023a; Dong & Sun, 2024; Wang et al., 2025; Sun et al., 2022; Fu et al., 2025; Xu et al., 2021; Yang et al., 2021a; Yu et al., 2022; Wang et al., 2023; Liu et al., 2023; Zhou et al., 2023; Fang et al., 2024; Yuan et al., 2024; Seo et al., 2024) to learn more discriminative node representations. Graph Information Bottleneck (GIB) (Wu et al., 2020) first extends the IB principle to graph neural networks to learn adversarially robust node representations under a local dependency assumption. The IB principle is also applied to contrastive and heterogeneous graph representation learning (Xu et al., 2021; Yang et al., 2021a), as well as graph-level learning tasks including graph classification (Yu et al., 2022; Wang et al., 2023), anomaly detection (Liu et al., 2023), graph reconstruction (Zhou et al., 2023), graph condensation (Fang et al., 2024), and temporal graph learning (Yuan et al., 2024; Seo et al., 2024). In the graph domain, attention mechanisms have been extensively explored, ranging from GAT (Veličković et al., 2018) and its robustness to adversarial attacks (Zügner & Günnemann, 2019; Sun et al., 2020; Goyal et al., 2023; Zhou et al., 2022), to uncertainty- and structure-aware models such as GNNGuard (Zhang & Zitnik, 2020), TWIRLS (Yang et al., 2021b), and UAG (Feng et al., 2021), as well as transformer-inspired architectures including NodeFormer (Wu et al., 2022a) and SGFormer (Wu et al., 2023b) built upon self-attention (Vaswani et al., 2017). To learn better attention weights against graph adversarial attacks, DGAIB (Dong & Sun, 2024) and IB-D^{2}GAT (Wang et al., 2025) propose to reduce the IB loss when training Graph Attention Networks (GATs) (Veličković et al., 2018). Similar to attention-based methods that assign weights to edges, VIB-GSL (Sun et al., 2022) and CurvGIB (Fu et al., 2025) propose to learn a better graph adjacency matrix by reducing the IB loss. Moreover, RM-GIB (Dai et al., 2023b), RGIB (Wang et al., 2024), and DGIB (Yuan et al., 2024) propose to reduce the IB loss to learn node representations that are robust against graph adversarial attacks. Notably, the training of RGIB (Wang et al., 2024) relies on the availability of clean subgraphs.

3 Methods

In this section, we propose a novel graph attention module inspired by the Information Bottleneck (IB) principle, termed Robust Graph Attention (RGA-IB). In Section 3.1, we detail the notations of the attributed graph and introduce the formulation of RGA-IB. Next, we present how the attention weight matrix of RGA-IB is generated to reduce IB loss in Section 3.2.

3.1 Robust Graph Attention inspired by Information Bottleneck (RGA-IB)

We begin by formally defining the notations used for an attributed graph. Subsequently, we present the detailed formulation of Robust Graph Attention inspired by Information Bottleneck, or RGA-IB.

Attributed Graph. An attributed graph with N nodes is formally denoted as $\mathcal{G} = (\mathcal{V}, X, A)$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edges. The labels of the nodes in $\mathcal{G}$ are $\{y_i\}_{i=1}^n$. $\mathcal{V}_\mathrm{L}$ and $\mathcal{V}_\mathrm{U}$ denote the sets of the indices of the labeled nodes and the unlabeled nodes. The node attributes are represented by $X \in \mathbb{R}^{N\times D}$, where each row $X_i \in \mathbb{R}^D$ corresponds to the attributes of node i and D is the attribute dimension. The adjacency matrix $A \in \{0,1\}^{N\times N}$ defines the connections in the graph $\mathcal{G}$, with $A_{ij} = 1$ if and only if there is an edge $(v_i, v_j) \in \mathcal{E}$. The adjacency matrix with self-loops included is given by $\widehat{A} = A + I$, and the corresponding diagonal degree matrix is $\widehat{D}$. $\tilde{A} = \widehat{D}^{-\frac{1}{2}}\widehat{A}\widehat{D}^{-\frac{1}{2}}$ is the normalized graph Laplacian.

Robust Graph Attention inspired by Information Bottleneck (RGA-IB). In this work, we aim to propose a novel graph attention operation, termed Robust Graph Attention inspired by Information Bottleneck (IB), or RGA-IB, which can be incorporated into multi-layer GNNs for semi-supervised node classification. We first introduce the setup of graph attention operation in GNNs. Let $X \in \mathbb{R}^{N\times D}$ be the input feature matrix to the graph attention operation. The output features of a GNN layer with graph attention operation are then calculated by $Z = B\sigma\left(\tilde{A}XW\right)$, where $W \in \mathbb{R}^{D\times D'}$ is the weight matrix for the

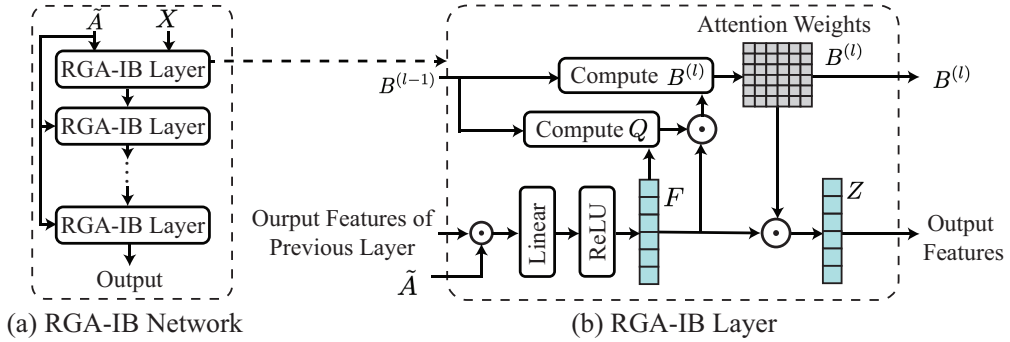


Figure 2: Overall framework of GNN with multiple Robust Graph Attention (RGA-IB) layers and the detailed structure of the RGA-IB layer. An RGA-IB network generates the node representations given the normalized adjacency matrix $\tilde{A}$ and node features X. Given the output features of the previous layer, the attention weight matrix $B^{(\ell-1)}$ of the previous layer, and the normalized adjacency matrix $\tilde{A}$, an RGA-IB layer first computes the latent node features F. Next, the attention weight matrix $B^{(l)}$ is generated by Equation (1). After obtaining the attention weight matrix $B^{(\ell)}$, the attention augmented node features are computed $Z = B^{(l)}F$. Detailed formulation of RGA-IB can be found in Section 3.1.

linear transformation of the input. D' is the hidden dimension of the linear transformation. $B \in \mathbb{R}^{N\times N}$ is the attention weight matrix, where B_{ij} denotes the feature correlation between node v_i and node v_j. $\sigma(\cdot)$ is a non-linear activation function such as ReLU. Let $F = \sigma(\tilde{A}XW) \in \mathbb{R}^{D'}$ be the latent node features before applying the graph attention operation. The output features of a GNN layer with graph attention operation can be denoted as $Z = BF$.

Graph attention operation has been widely studied for designing Graph Neural Networks (GNNs) that adaptively model the feature correlation between nodes in a graph (Lee et al., 2019; Veličković et al., 2018). The major differences between different graph attention operations lie in how the attention weight matrix B is computed. For instance, GAT (Veličković et al., 2018) concatenates the node features of different nodes and applies a linear transformation to compute their similarity. Difformer (Wu et al., 2023a) adopts the dot-product operation widely used in the self-attention operation in transformers (Vaswani et al., 2017) to capture pair-wise similarities among nodes in a graph. Although the graph attention operation has been widely studied, all existing graph attention methods do not realize that the graph attention operation can reduce the IB loss of GNNs by enhancing the correlation of learned features with class labels while reducing their correlation with the input. The enhanced robustness of GNNs with graph attention is attributed to the ability of graph attention modules to enhance informative signals while diminishing noise or less pertinent details. The selective attention mechanism produces node representations that are less correlated with the input training data, which might contain adversarial noise, and more aligned with the desired outputs, adhering more closely to the Information Bottleneck (IB) principle. Inspired by the observation that graph attention can reduce the IB loss of GNNs, we propose RGA-IB that explicitly reduces the IB loss via graph attention operations at consecutive layers. Similar to existing graph attention methods, a linear layer followed by a non-linear activation function is first applied to the input features at each RGA-IB layer to obtain the latent node features F. At the ℓ-th RGA-IB layer in a multi-layer RGA-IB network, the output features are computed by $Z = B^{(\ell)}\sigma\left(\tilde{A}XW\right)$, where $B^{(\ell)}$ is the graph attention weight matrix of the ℓ-th RGA-IB layer. Here $\ell \in [L]$ where L is the total number of RGA-IB layers. As illustrated in Figure 2, the graph attention weight matrix $B^{(\ell)}$ of the ℓ-th RGA-IB layer is generated by the attention weight matrix $B^{(\ell-1)}$ of the previous layer and the latent node features F, which is motivated by reducing the IB loss and detailed in Section 3.2. The attention weight matrix $B^{(1)}$ for the first layer of the RGA-IB network is generated from the input node features X with a GAT layer (Veličković et al., 2018).

3.2 Generating Attention Weight Matrix of RGA-IB by Reducing the IB Loss

In this section, we detail the process of generating the attention weight matrix for an RGA-IB layer inspired by reducing the Information Bottleneck (IB) loss. We first describe the configuration in which the IB

loss for a GNN is specified. Given the training data $\{X_i, y_i\}_{i=1}^N$, where $X_i \in \mathbb{R}^D$ is the i-th input node feature, and y_i is the corresponding class label, we first specify how to compute the IB loss, $\mathrm{IB}(Z, X, Y) = I(Z, X) - I(Z, Y)$, where X is a random variable representing the input features, which takes values in $\{X_i\}_{i=1}^N$. Y is a random variable representing the class label, which takes values in $\{y_i\}_{i=1}^N$. Z is a random variable representing the node representations, which takes values in $\{Z_i\}_{i=1}^N$ with $Z_i \in \mathbb{R}^D$ being the node representation of node v_i. $I(\cdot, \cdot)$ denotes the mutual information. To compute the mutual information, we first calculate the class centroids on $\{Z_i\}_{i=1}^N$ and $\{X_i\}_{i=1}^N$, resulting in class centroids $\{\mathcal{C}_a\}_{a=1}^C$ and $\left\{\mathcal{C}_b'\right\}_{b=1}^C$ for representation space and input feature space respectively, where C is the number of classes. Then we define the probability that node representation Z belongs to class a as $\Pr[Z \in a] = \frac{1}{N}\sum_{i=1}^{N} \phi(Z_i, a)$, where $\phi(Z_i, a) = \frac{\exp\left(-\|Z_i - C_a\|_2^2\right)}{\sum_{t=1}^{A} \exp\left(-\|Z_i - C_t\|_2^2\right)}$. Similarly, we define the probability that the input node feature X belongs to class b as $\Pr[X \in b] = \frac{1}{n}\sum_{i=1}^{n} \phi(X_i, b)$, with $\phi(X_i, b) = \frac{\exp\left(-\left\|X_i - C_b'\right\|_2^2\right)}{\sum_{t=1}^{B} \exp\left(-\left\|X_i - C_t'\right\|_2^2\right)}$. Moreover, the joint probabilities are calculated by $\Pr[Z \in a, X \in b] = \frac{1}{N}\sum_{i=1}^{N} \phi(Z_i, a)\phi(X_i, b)$ and $\Pr[Z \in a, Y = y] = \frac{1}{N}\sum_{i=1}^{N} \phi(Z_i, a)\mathbb{1}_{\{y_i = y\}}$, where $\mathbb{1}_{\{\}}$ is an indicator function. As a result, the mutual information $I(Z, X)$ and $I(Z, Y)$ can be computed by

$$
\begin{aligned}
I(Z, X) &= \sum_{a=1}^{A}\sum_{b=1}^{B} \Pr[Z \in a, X \in b] \ln \frac{\Pr[Z \in a, X \in b]}{\Pr[Z \in a]\Pr[X \in b]}, \\
I(Z, Y) &= \sum_{a=1}^{A}\sum_{y=1}^{C} \Pr[Z \in a, Y = y] \ln \frac{\Pr[Z \in a, Y = y]}{\Pr[Z \in a]\Pr[Y = y]},
\end{aligned}
$$

and the IB loss $\mathrm{IB}(Z, X, Y)$ is then computed by

$$
\begin{aligned}
\mathrm{IB}(Z, X, Y) = &\sum_{a=1}^{A}\sum_{b=1}^{B} \Pr[Z \in a, X \in b] \ln \frac{\Pr[Z \in a, X \in b]}{\Pr[Z \in a]\Pr[X \in b]} \\
&- \sum_{a=1}^{A}\sum_{y=1}^{C} \Pr[Z \in a, Y = y] \ln \frac{\Pr[Z \in a, Y = y]}{\Pr[Z \in a]\Pr[Y = y]}.
\end{aligned}
$$

In the RGA-IB network, the attention weight matrix B is updated by simulating one-step gradient descent on the IB loss and explicitly projected onto the adjacency matrix A after each update. In particular, we apply an element-wise masking operation on $\nabla_B \mathrm{IB}(B^{(\ell-1)})$, such that the gradient descent update is restricted to the edges of the graph, $B^{(\ell)} = B^{(\ell-1)} - \eta\left(\nabla_B \mathrm{IB}(B^{(\ell-1)})\right) \circ A$, where η denotes the learning rate and $\circ$ denotes the Hadamard product or element-wise product operation. As a result, the attention matrix B remains sparse throughout training and inference, sharing the same sparsity pattern as the original adjacency matrix A.

Theorem 3.1. Suppose $Z = BF$, with B being the attention weight matrix and F being the hidden node feature before applying the graph attention operation. For simplicity, we denote $\mathrm{IB}(Z, X, Y)$ or $\mathrm{IB}(BF, X, Y)$ by $\mathrm{IB}(B)$. At step ℓ of gradient descent on $\mathrm{IB}(B) = I(BF, X) - I(BF, Y)$, we have

$$
B^{(\ell)} = B^{(\ell-1)} - \eta\left(\nabla_B \mathrm{IB}(B^{(\ell-1)})\right) \circ A = B^{(\ell-1)} - \eta\left(Q^{(\ell-1)} \cdot F^\top\right) \circ A. \tag{1}
$$

where η is the learning rate. $Q^{(\ell-1)} = \nabla_Z I(Z^{(\ell-1)}, X) - \nabla_Z I(Z^{(\ell-1)}, Y)$, where $Z^{(\ell-1)} = B^{(\ell-1)}F$. Formulations of $\nabla_Z I(Z, X)$ and $\nabla_Z I(Z, Y)$ are deferred to Lemma A.1 and Lemma A.2 in Section A of the appendix.

The proof of Theorem 3.1 is deferred to Section A of the appendix. Inspired by Theorem 3.1, we can understand a GNN with graph attention operations at multiple layers as an iterative process that reduces

$\mathrm{IB}(B)$ by gradient descent. The ℓ-th graph attention layer simulates one step of gradient descent on $\mathrm{IB}(B)$ according to Equation (1). Based on the gradient descent formulation, we design the formulation of the RGA-IB network, whose attention weight matrix $B^{(\ell)}$ at the ℓ-th RGA-IB layer is generated from $B^{(\ell-1)}$ by Equation (1), the attention weight matrix of the previous RGA-IB layer. Figure 2 illustrates the overall framework of an RGA-IB network and the structure of the RGA-IB layer. The training of RGA-IB for semi-supervised node classification is supervised by the cross-entropy loss on the labeled nodes as $\mathcal{L}_{\text{train}} = \sum_{i \in \mathcal{V}_{\text{L}}} \mathrm{CE}(Z_i, y_i)$, where $\mathcal{V}_{\text{L}}$ denotes the set of indices of labeled nodes. $\mathrm{CE}(\cdot, \cdot)$ is the cross-entropy function. We present the training algorithm of the RGA-IB network in Algorithm 1.

Complexity Analysis. We now analyze the training complexity for a particular graph attention layer with the attention matrix $B^{(\ell)}$ at the ℓ-th layer of the GNN. Suppose $F \in \mathbb{R}^{N \times D'}$ is the hidden features with hidden dimension D' as the input to the ℓ-th layer, C is the number of classes, E is the number of edges, and t_{train} denotes the number of training epochs. Let T_0 be the computational complexity of the forward pass for computing F. RGA-IB first computes soft class assignments $\phi(Z_i, a)$ for all nodes and classes, which requires $\Theta(NCD')$ complexity. The computation of the gradients $\nabla_Z I(Z, X)$ and $\nabla_Z I(Z, Y)$ involves double summations over representation and label indices, leading to a complexity of $\Theta(NC^2D')$ for computing $Q^{(\ell-1)}$. The attention weight matrix $B^{(\ell)}$ is updated with a complexity of $\Theta(ED')$ according to Equation (1) since the update is restricted to the edges in the graph. As a result, the total training complexity for $B^{(\ell)}$ is $\Theta\left(t_{\text{train}} \cdot \left(T_0 + NC^2D' + ED'\right)\right)$. In practice, C is typically small in semi-supervised node classification benchmarks with $C \ll N$, so the overall training complexity is linear in the number of nodes N and the number of edges E. As shown in Table 8 in Section C.1 of the appendix, the training time of RGA-IB is comparable to a standard GAT (Veličković et al., 2018) with the same number of graph attention layers, while RGA-IB significantly outperforms the latter.

Algorithm 1: Training Algorithm of the RGA-IB network

1: **Input:** The number of training epochs t_{train}, the number of warm-up epochs t_{warm},
2: **Output:** The weights $\mathcal{W}$ of the RGA-IB network.
3: Randomly initialize the weights of the RGA-IB network $\mathcal{W}$.
4: Initialize the attention weight matrices $\left\{B^{(\ell)}\right\}_{\ell=1}^{L}$ of all RGA-IB layers to identity matrices.
    for $t \leftarrow 1$ *to* t_{train} **do**
        if $t < t_{warm}$ **then**
            Perform gradient descent by a standard step of SGD on the cross-entropy loss $\mathcal{L}_{\text{train}}$ to update the weights in the RGA-IB network with fixed attention weight matrices.
        else
            Update $\phi(Z_i, a)$ for all the clusters $a \in [C_a]$ and $i \in [N]$.
            Forward step: compute the attention weight matrices $\left\{B^{(\ell)}\right\}_{\ell=1}^{L}$ for all the RGA-IB layers by Equation (1) using the updated $\left\{\mathcal{C}_a^{(t)}\right\}_{a=1}^{C}$, and compute $\{Z_i\}_{i=1}^{N}$.
            Backward step: perform gradient descent by a standard step of SGD on the cross-entropy loss $\mathcal{L}_{\text{train}}$ to update the weights in the RGA-IB network.
            Compute the class centroids $\{\mathcal{C}_a^{(t)}\}_{a=1}^{C}$ with the updated node representations $\{Z_i\}_{i=1}^{N}$.
5: **return** The trained weights $\mathcal{W}$ of the RGA-IB network.

4 Experiments

In this section, we perform empirical evaluations of RGA-IB on public graph benchmarks for semi-supervised node classification under graph adversarial attacks. Implementation details of our experiments are presented in Section 4.1. Experiment results for semi-supervised node classification under adversarial attacks are presented in Section 4.2. Comprehensive ablation studies on IB loss at different layers of RGA-IB and the effects of RGA-IB in reducing IB loss are presented in Section 4.3. Additional experimental results are presented in Section C of the appendix. In Section C.1, we evaluate the training time of the proposed RGA-IB in comparison to the widely used attention-based GNN, GAT (Veličković et al., 2018). In Section C.2, we perform experiments for adversarial defense on large-scale graph datasets, Reddit (Hamilton et al., 2017b)

and ogbn-arxiv (Hu et al., 2020). In Section C.3, we perform experiments on heterophilic graph datasets, Chameleon (Pei et al., 2020) and Actor (Tang et al., 2009). Performance comparison between RGA-IB with additional baseline methods is presented in Section C.4 of the appendix. Both models are trained for 500 epochs on all the evaluated datasets. In addition, implementation details on datasets, training settings, and attack settings are presented in Section B of the appendix.

4.1 Implementation Details

The experiments are conducted on Cora, Citeseer, Pubmed (Sen et al., 2008), and Polblogs (Adamic & Glance, 2005), Reddit (Hamilton et al., 2017b), ogbn-arxiv (Hu et al., 2020), Chameleon (Pei et al., 2020), and Actor (Tang et al., 2009), where Reddit and ogbn-arxiv are large-scale graph datasets. Chameleon and Actor are heterophilic graph datasets, where connected nodes are more likely to belong to different classes rather than the same class. Following the settings in existing works on graph adversarial attacks (Jin et al., 2020; Zhao et al., 2023), we evaluate our method and competing baselines on the largest connected component (LCC) of the graph datasets. Details on the statistics of the datasets are deferred to Table 6 in Section B.1 of the supplementary. In our experiments, we randomly choose 10% of nodes for training, 10% of nodes for validation, and the remaining 80% of nodes for testing following (Jin et al., 2020). For the training of the RGA-IB network, we first warm up the training of the network parameters for 100 epochs by only optimizing the weights of the linear layers and fixing the attention weight matrices as identity matrices. Following that, we train all the network weights in the RGA-IB network for 500 epochs. Adam is used as the optimizer for the training. Additional training settings and implementation details are deferred to Section B of the appendix.

Table 1: Node classification performance (Accuracy±Std) under non-targeted attack Metattack (Zügner & Günnemann, 2019). The best result is highlighted in bold, and the second-best result is underlined. This convention is followed by all the tables in this paper. The results of RGA-IB are followed by the improvements over the best baselines. Results for comparison to GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a) are deferred to Table 11 in Section C.4 of the appendix.

Dataset	Ptb Rate	DGIB	CurvGIB	HANG	RGIB	Differmer	GAR	GCORNs	Pro-GNN	RGA-IB (Ours)
Cora	0	82.8±0.6	82.8±0.6	80.0±0.3	82.1±0.6	84.9±0.6	83.2±0.6	82.5±0.4	82.9±0.2	**85.0** (↑0.1) ±1.3
	5	81.4±1.4	82.6±1.4	78.1±1.2	79.9±1.2	76.6±1.3	75.2±1.1	75.9±1.8	82.2±1.9	**83.9** (↑1.3) ±1.9
	10	78.6±1.6	79.4±1.6	75.9±1.5	72.6±1.8	73.6±1.6	73.3±1.3	74.6±2.0	79.0±1.9	**81.2** (↑1.8) ±1.2
	15	73.2±1.6	75.3±1.6	72.8±1.6	72.0±1.5	72.4±1.3	71.4±1.7	72.5±1.2	76.4±1.3	**77.9** (↑1.5) ±1.7
	20	71.5±1.5	73.5±1.6	70.3±1.3	69.2±1.8	70.2±2.2	71.5±2.4	71.9±1.5	73.3±1.8	**74.5** (↑1.0) ±1.1
	25	68.5±1.8	68.5±1.9	68.7±1.3	66.8±2.2	65.9±2.1	68.3±2.8	69.4±2.9	69.7±1.5	**71.0** (↑1.3) ±1.1
Citeseer	0	71.9±0.3	72.5±1.5	73.2±0.3	72.5±0.1	73.3±0.7	73.1±0.3	72.6±0.4	73.2±0.6	**74.4** (↑1.1) ±1.0
	5	70.8±1.5	72.2±1.8	73.0±2.2	67.3±1.3	69.4±1.5	68.2±0.8	68.1±1.6	72.9±1.1	**74.0** (↑1.0) ±2.0
	10	70.4±1.5	72.0±1.9	72.4±2.2	65.2±1.4	66.2±2.0	65.2±1.4	66.2±1.5	72.5±1.1	**73.7** (↑1.2) ±1.7
	15	70.8±1.4	72.1±1.4	70.8±2.0	64.5±1.2	65.0±1.3	64.5±1.8	65.4±2.0	72.0±1.9	**73.2** (↑1.1) ±1.7
	20	70.1±1.6	69.8±1.3	66.1±2.0	64.1±1.4	63.6±1.4	64.4±2.5	65.5±1.5	70.0±2.3	**71.4** (↑1.3) ±1.9
	25	67.7±2.3	68.5±2.5	66.4±2.8	63.1±2.2	62.6±2.3	64.4±2.3	65.2±2.2	68.9±1.7	**69.9** (↑1.0) ±1.9
Polblogs	0	95.0±0.8	95.5±0.6	94.7±1.0	95.2±0.8	95.7±0.2	95.5±0.1	95.3±0.8	93.2±0.6	**96.5** (↑0.8) ±1.1
	5	89.7±0.9	89.7±1.9	80.1±1.3	90.1±0.2	89.6±2.2	90.1±0.2	89.3±0.3	93.2±1.4	**94.0** (↑0.8) ±1.2
	10	88.6±1.2	89.5±1.4	74.9±1.3	85.4±1.3	85.2±1.7	85.9±1.4	87.3±1.7	89.4±1.4	**91.1** (↑1.6) ±1.4
	15	80.6±1.6	84.0±1.7	71.6±1.6	78.8±1.9	79.5±1.8	83.0±1.7	82.0±1.4	86.0±1.9	**87.1** (↑1.1) ±1.3
	20	78.8±1.6	79.8±1.5	66.2±2.1	68.8±1.3	70.9±1.8	69.4±1.2	69.9±1.2	79.5±1.7	**80.8** (↑1.0) ±1.3
	25	68.6±1.8	68.6±1.8	65.8±2.3	64.8±1.4	68.0±2.0	67.1±1.9	66.5±2.7	63.1±2.0	**69.9** (↑1.3) ±1.6
Pubmed	0	85.0±0.3	85.0±0.3	85.0±0.2	84.6±0.1	87.3±0.1	87.2±0.2	86.4±0.7	87.3±0.1	**88.3** (↑1.0) ±1.2
	5	85.9±1.3	85.2±1.3	85.0±2.0	80.8±0.2	79.6±0.3	81.2±0.6	82.9±0.8	87.2±1.7	**88.1** (↑0.9) ±1.9
	10	81.6±1.4	84.8±1.5	85.1±1.8	84.7±1.3	84.0±1.7	84.0±1.6	80.7±1.6	87.2±1.6	**88.5** (↑1.3) ±1.7
	15	82.8±1.4	84.0±1.4	85.0±1.4	84.6±1.2	83.6±1.8	82.3±1.2	76.8±1.8	87.2±2.3	**88.2** (↑1.0) ±1.2
	20	82.5±2.0	84.6±2.0	85.2±2.8	81.8±1.5	80.9±2.6	80.4±1.8	71.2±1.5	87.1±2.1	**88.0** (↑0.9) ±1.0
	25	81.6±2.1	84.1±2.1	85.0±2.1	81.1±2.3	79.6±2.5	79.2±2.5	70.7±2.8	86.7±2.7	**87.8** (↑1.1) ±1.3

4.2 Semi-supervised Node Classification Under Adversarial Attacks

In this section, we summarize the experimental results of the proposed RGA-IB on semi-supervised node classification under graph adversarial attacks. We compare the node classification accuracy of RGA-IB

with the most competitive baseline methods, including DGIB (Yuan et al., 2024), IB-D^{2}GAT (Wang et al., 2025), CurvGIB (Fu et al., 2025), HANG (Zhao et al., 2023), RGIB (Wang et al., 2024), RM-GIB (Dai et al., 2023a), Differmer (Wu et al., 2023a), GAR (Fountoulakis et al., 2023), GCORNs (Abbahaddou et al., 2024), and Pro-GNN (Jin et al., 2020), across different types of graph adversarial attacks and varying attack strengths. Details on the attack settings are deferred to Section B.3 of the appendix. The results for Metattack (Zügner & Günnemann, 2019), Nettack (Zügner et al., 2018), and Topology Attack (Xu et al., 2019a) are shown in Table 1, Table 2, and Table 3, respectively. Comparisons with additional baseline methods including, GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a) for Metattack, Nettack, and Topology Attack are deferred to Table 11, Table 12, and Table 13 in Section C.4 of the appendix. We run all experiments ten times and report the mean and standard deviation of the node classification accuracy. It is observed from the results that RGA-IB significantly outperforms existing robust graph learning methods under different adversarial attacks. For instance, the average improvement of RGA-IB over the second-best methods across different attack budgets on Polblogs for Topology Attack is 1.23%, demonstrating the superiority of RGA-IB in mitigating the negative impacts of adversarial attacks by reducing the IB loss with the robust graph attention design.

Table 2: Node classification performance (Accuracy±Std) under targeted attack Nettack (Zügner et al., 2018). The results of RGA-IB are followed by the improvements over the best baselines. Results for comparison to GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a) are deferred to Table 12 in Section C.4 of the appendix.

Dataset	Budget	DGIB	CurvGIB	HANG	RGIB	Differmer	GAR	GCORNs	Pro-GNN	RGA-IB (Ours)
Cora	0	80.6±1.6	85.0±1.5	80.8±1.3	81.3±1.8	84.9±1.8	83.3±0.7	82.6±0.4	84.8±0.6	**86.1** (↑ 1.1) ±1.1
	1	82.0±2.4	83.5±2.2	77.0±3.2	80.0±1.5	80.3±1.6	81.3±1.2	81.0±1.1	83.8±0.5	**85.5** (↑ 1.7) ±1.4
	2	75.8±2.3	77.2±2.1	76.5±2.6	75.9±2.2	78.5±1.7	75.9±1.3	78.4±1.3	78.7±0.7	**80.7** (↑ 2.0) ±1.2
	3	73.1±2.0	73.5±1.9	73.1±2.9	71.9±1.4	73.1±1.5	73.5±1.1	73.1±0.7	72.4±0.5	**74.7** (↑ 1.2) ±1.4
	4	72.6±2.0	72.6±1.0	72.5±2.1	70.5±1.2	70.5±1.2	70.9±1.2	71.2±0.6	70.1±0.7	**73.6** (↑ 1.0) ±1.2
	5	66.8±2.6	68.4±2.4	68.8±2.6	64.4±2.4	67.4±1.5	67.2±1.2	66.8±1.0	66.9±0.7	**69.9** (↑ 1.1) ±1.8
Citeseer	0	80.5±1.2	81.6±1.1	81.1±1.1	80.6±1.0	82.3±1.2	81.4±1.3	81.9±1.1	82.1±0.8	**83.4** (↑ 1.1) ±1.7
	1	78.6±2.1	80.4±1.9	79.1±1.4	79.0±2.8	80.8±2.5	80.4±2.8	80.7±0.8	81.8±0.8	**82.9** (↑ 1.1) ±1.9
	2	77.8±1.7	80.0±1.6	77.9±2.3	78.2±1.2	79.4±0.2	79.9±0.9	79.6±2.8	81.3±1.0	**82.5** (↑ 1.2) ±1.4
	3	77.0±2.9	78.4±1.8	77.1±2.5	78.1±1.3	77.1±2.2	77.9±2.6	78.1±2.0	79.7±2.0	**81.0** (↑ 1.3) ±1.6
	4	77.2±1.1	78.5±2.0	78.4±1.6	77.4±2.5	69.2±3.4	69.4±3.9	77.7±1.6	77.8±2.8	**79.5** (↑ 1.0) ±1.6
	5	71.8±1.9	72.6±2.7	73.5±3.5	70.5±2.5	69.2±2.3	69.7±2.6	71.6±2.7	71.3±5.0	**74.6** (↑ 1.1) ±1.9
Polblogs	0	97.2±1.5	97.3±0.4	97.4±0.5	97.3±0.3	97.2±0.4	97.4±0.8	97.5±0.2	97.1±0.2	**98.2** (↑ 0.7) ±1.6
	1	97.0±0.5	97.2±0.4	97.4±0.4	97.1±0.3	97.4±0.2	97.5±0.4	97.6±0.5	96.8±0.1	**98.1** (↑ 0.5) ±1.9
	2	96.0±0.5	96.6±0.4	96.9±0.2	95.3±0.5	97.1±0.2	95.9±0.3	96.0±0.4	97.2±0.1	**97.8** (↑ 0.6) ±1.5
	3	96.0±0.4	96.6±0.3	96.7±0.2	95.4±0.1	96.9±0.9	95.3±0.7	95.6±1.0	96.9±0.1	**97.5** (↑ 0.6) ±1.1
	4	95.4±0.6	96.0±0.5	96.3±0.5	94.9±0.6	96.8±0.6	94.1±0.2	94.4±0.5	96.9±0.2	**97.4** (↑ 0.5) ±1.3
	5	94.8±0.5	95.4±0.4	95.9±0.3	93.2±0.4	95.1±0.8	93.6±0.7	93.4±0.8	96.1±0.3	**97.2** (↑ 1.1) ±1.5
Pubmed	0	85.0±1.2	88.6±1.1	85.4±1.2	85.4±1.1	87.4±1.8	84.6±1.5	84.5±1.3	88.5±1.2	**89.6** (↑ 1.0) ±1.9
	1	84.1±1.9	87.3±1.7	84.4±2.0	84.5±1.6	86.0±2.3	83.8±1.6	83.8±1.9	87.6±2.0	**88.8** (↑ 1.2) ±1.4
	2	83.9±1.9	85.9±2.6	84.4±3.1	84.1±3.5	86.0±2.2	84.3±1.5	84.4±1.7	85.9±3.1	**87.0** (↑ 1.0) ±1.3
	3	83.8±2.5	84.6±2.2	84.0±3.1	84.3±3.3	81.8±3.5	82.3±1.1	82.2±1.4	84.4±3.1	**85.6** (↑ 1.0) ±1.9
	4	78.6±2.3	80.1±2.9	79.7±1.3	78.0±3.2	77.4±2.4	80.0±2.1	81.0±2.7	80.0±3.3	**81.6** (↑ 1.5) ±1.2
	5	73.5±2.1	75.3±2.0	70.6±2.3	71.2±1.6	71.3±2.2	75.1±3.4	75.2±1.1	72.2±4.3	**76.8** (↑ 1.5) ±1.3

4.3 Ablation Study

Study on the IB Loss at Different Layers of RGA-IB. To study how the IB loss IB(B) decreases with respect to layer index ℓ of an RGA-IB network, we calculate IB(B) across different RGA-IB layers in the RGA-IB network. The study is performed on Cora and Citeseer under Topology Attack with a perturbation rate of 25% with both a 2-layer RGA-IB network and a 4-layer RGA-IB network. We also calculate the IB loss at different layers of 2-layer Differmer, 4-layer Differmer, 2-layer GAR, and 4-layer GAR. It is observed from the results in Table 4 that the IB loss decreases in deeper layers with a larger layer index for both the 2-layer RGA-IB network and the 4-layer RGA-IB network. In Table 4, all the layers of RGA-IB network are graph attention layers with their attention weights updated by Equation (1). This observation suggests that node features in deeper layers of RGA-IB networks correlate more closely with the class labels and less with

Table 3: Node classification performance (Accuracy±Std) under Topology Attack (Xu et al., 2019a). The results of RGA-IB are followed by the improvements over the best baselines. Results for comparison to GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a) are deferred to Table 13 in Section C.4 of the appendix.

Dataset	Ptb Rate	DGIB	CurvGIB	HANG	RGIB	Difformer	GAR	GCORNs	Pro-GNN	RGA-IB (Ours)
Cora	0	82.8±0.6	82.8±0.6	80.0±0.3	82.1±0.6	84.9±0.6	83.2±0.6	82.5±0.4	82.9±0.2	**85.0** (↑ 0.1) ±1.3
	5	76.0±1.3	77.8±1.0	76.9±1.2	76.8±1.2	77.0±1.3	75.6±1.1	75.9±1.8	77.6±1.9	**79.6** (↑ 1.8) ±1.9
	10	75.1±1.6	77.5±1.1	76.8±1.5	75.9±1.8	74.0±1.6	73.7±1.3	74.6±2.0	77.3±1.9	**79.3** (↑ 1.8) ±1.2
	15	73.8±1.5	75.0±1.8	75.6±1.6	74.3±1.5	72.8±1.3	71.8±1.7	72.5±1.2	75.4±1.3	**76.9** (↑ 1.3) ±1.7
	20	71.7±1.5	72.4±1.7	72.2±1.3	72.0±1.8	70.6±2.2	71.9±2.4	71.9±1.5	71.3±1.8	**74.5** (↑ 2.1) ±1.1
	25	68.7±1.7	69.5±1.7	69.1±1.3	68.9±2.2	66.3±2.1	68.7±2.8	69.4±2.9	68.7±1.5	**71.0** (↑ 1.5) ±1.1
Citeseer	0	71.9±0.3	72.5±1.5	73.2±0.3	72.5±0.1	73.3±0.7	73.1±0.3	72.6±0.4	73.2±0.6	**74.4** (↑ 1.1) ±1.0
	5	69.7±1.6	71.0±1.3	69.4±2.2	69.5±1.3	69.8±1.5	68.6±0.8	68.1±1.6	71.2±1.1	**72.7** (↑ 1.5) ±2.0
	10	66.3±1.6	68.6±1.3	67.5±2.2	68.0±1.4	66.6±2.0	65.6±1.4	66.2±1.5	68.2±1.1	**69.7** (↑ 1.1) ±1.7
	15	65.4±1.4	66.5±1.4	66.9±2.0	66.9±1.2	65.4±1.3	64.9±1.8	65.4±2.0	67.0±1.9	**68.5** (↑ 1.5) ±1.7
	20	65.1±1.6	66.4±1.1	66.4±2.0	65.3±1.4	64.0±1.4	64.8±2.5	65.5±1.5	65.9±2.3	**67.4** (↑ 1.0) ±1.9
	25	64.0±2.4	65.0±2.0	65.1±2.8	63.5±2.2	63.0±2.3	64.8±2.3	65.2±2.2	64.9±1.7	**66.7** (↑ 1.5) ±1.9
Polblogs	0	95.0±0.8	95.5±0.6	94.7±1.0	95.2±0.8	95.7±0.2	95.5±0.1	95.3±0.8	93.2±0.6	**96.5** (↑ 0.8) ±1.1
	5	90.2±0.9	90.6±0.2	90.3±1.3	90.0±0.2	90.0±2.2	90.5±0.2	89.3±0.3	90.9±1.4	**92.0** (↑ 1.1) ±1.2
	10	86.1±1.3	87.5±1.1	86.7±1.3	86.6±1.3	85.6±1.7	86.3±1.4	87.3±1.7	87.1±1.4	**89.1** (↑ 1.6) ±1.4
	15	81.2±1.6	82.9±1.2	82.3±1.6	82.1±1.9	79.9±1.8	83.4±1.7	82.0±1.4	83.3±1.9	**85.0** (↑ 1.6) ±1.3
	20	72.5±1.6	71.9±1.1	72.6±2.1	69.4±1.3	71.3±1.8	69.8±1.2	69.9±1.2	72.5±1.7	**73.8** (↑ 1.2) ±1.3
	25	69.8±1.8	70.4±1.8	70.8±2.3	68.2±1.4	68.4±2.0	67.5±1.9	66.5±2.7	69.4±2.0	**71.9** (↑ 1.1) ±1.6
Pubmed	0	85.0±0.3	85.0±0.3	85.0±0.2	84.6±0.1	87.3±0.1	87.2±0.2	86.4±0.7	87.3±0.1	**88.3** (↑ 1.0) ±1.2
	5	81.4±1.2	82.6±1.4	82.2±2.0	82.4±0.2	80.0±0.3	81.6±0.6	82.9±0.8	83.1±1.7	**84.2** (↑ 1.1) ±1.9
	10	80.2±1.4	81.1±1.2	80.6±1.8	80.8±1.3	78.4±1.7	80.4±1.6	80.7±1.6	81.3±1.6	**82.5** (↑ 1.2) ±1.7
	15	76.3±1.4	77.3±1.3	77.3±1.4	76.9±1.2	74.0±1.8	75.7±1.2	76.8±1.8	77.1±2.3	**78.7** (↑ 1.4) ±1.2
	20	72.9±2.0	73.6±1.9	73.2±2.8	72.2±1.5	71.3±2.6	72.8±1.8	71.2±1.5	72.9±2.1	**74.6** (↑ 1.0) ±1.0
	25	72.0±2.2	72.5±2.0	72.7±2.1	72.8±2.3	70.0±2.5	71.6±2.5	70.7±2.8	71.7±2.7	**73.9** (↑ 1.2) ±1.3

the input node attributes, adhering to the IB principle. Moreover, the RGA-IB networks reduce the IB loss to lower levels in deeper layers compared to Difformer and GAR, demonstrating the superiority of RGA-IB over the existing state-of-the-art graph attention method and robust graph attention method.

Table 4: IB loss at different layers of RGA-IB. The study is performed on Cora and Citeseer under Topology Attack with a perturbation rate of 25%.

Methods	Layer Number	Cora					Citeseer				
		IB Loss				ACC	IB Loss				ACC
		Layer 1	Layer 2	Layer 3	Layer 4		Layer 1	Layer 2	Layer 3	Layer 4	
Difformer	2	-0.096	-0.115	-	-	65.8	-0.088	-0.118	-	-	62.8
Difformer	4	-0.077	-0.095	-0.099	-0.122	66.3	-0.070	-0.082	-0.103	-0.117	63.0
GAR	2	-0.089	-0.132	-	-	68.2	-0.074	-0.125	-	-	64.2
GAR	4	-0.065	-0.092	-0.116	-0.138	68.7	-0.070	-0.112	-0.130	-0.125	64.8
RGA-IB	2	-0.134	-0.208	-	-	70.8	-0.096	-0.185	-	-	66.4
RGA-IB	4	-0.125	-0.179	-0.195	-0.214	**71.0**	-0.082	-0.146	-0.175	-0.189	**66.7**

Study on the Effects of RGA-IB in Reducing the IB Loss. In this section, we evaluate the effectiveness of RGA-IB in minimizing the IB loss. We calculate the IB loss for both RGA-IB and baseline graph attention methods, including GAT (Veličković et al., 2018), UAG (Feng et al., 2021), RGCN (Zhu et al., 2019), Difformer (Wu et al., 2023a), and GAR (Fountoulakis et al., 2023) on Cora and Citeseer under Topology Attack with different perturbation rates. The results are shown in Table 5. It is observed from the results that the IB loss of graph attention methods correlates closely with the node classification accuracy under adversarial attacks, and the two methods with the lowest IB losses always achieve the two best robust accuracies. Graph attention methods with lower IB loss better adhere to the IB principle. Moreover, the IB loss can be further decreased to a considerable extent by optimizing the IB loss explicitly in RGA-IB.

Table 5: Ablation study on the effects of RGA-IB in reducing the IB loss compared to the existing graph attention methods. Node classification accuracy for all methods on all the datasets is attached in parentheses after the IB loss.

Dataset	Attack Budget	GAT	RGCN	UAG	Difformer	GAR	RGA-IB (Ours)
Cora	0	-0.105 (84.0)	-0.084 (83.1)	-0.103 (82.1)	-0.114 (84.9)	-0.096 (83.2)	**-0.127** (**85.0**)
	5	-0.110 (77.0)	-0.094 (75.0)	-0.105 (76.2)	-0.109 (77.0)	-0.117 (75.6)	**-0.128** (**79.2**)
	10	-0.102 (74.1)	-0.101 (73.1)	-0.110 (76.0)	-0.118 (74.0)	-0.122 (73.7)	**-0.144** (**79.1**)
	15	-0.105 (70.6)	-0.102 (71.6)	-0.105 (74.9)	-0.117 (72.8)	-0.125 (71.8)	**-0.179** (**76.9**)
	20	-0.109 (68.8)	-0.105 (67.9)	-0.110 (71.9)	-0.115 (70.6)	-0.133 (71.9)	**-0.186** (**74.2**)
	25	-0.114 (65.5)	-0.100 (66.0)	-0.119 (68.0)	-0.122 (66.3)	-0.138 (68.7)	**-0.214** (**71.0**)
Citeseer	0	-0.110 (73.3)	-0.080 (71.2)	-0.093 (72.1)	-0.124 (73.3)	-0.115 (73.1)	**-0.133** (**74.4**)
	5	-0.102 (68.2)	-0.082 (68.2)	-0.090 (71.3)	-0.114 (69.8)	-0.119 (68.6)	**-0.141** (**72.7**)
	10	-0.095 (66.7)	-0.084 (65.9)	-0.092 (67.9)	-0.088 (66.6)	-0.130 (65.6)	**-0.153** (**69.7**)
	15	-0.115 (63.6)	-0.110 (64.6)	-0.100 (66.4)	-0.091 (65.4)	-0.119 (64.9)	**-0.172** (**68.5**)
	20	-0.104 (61.9)	-0.108 (62.9)	-0.098 (64.0)	-0.112 (64.0)	-0.145 (64.8)	**-0.176** (**67.4**)
	25	-0.102 (59.1)	-0.092 (60.9)	-0.110 (62.9)	-0.117 (63.0)	-0.130 (64.8)	**-0.189** (**66.7**)

5 Conclusions

In this work, we propose a novel graph attention method termed Robust Graph Attention inspired by Information Bottleneck, or RGA-IB, which reduces the IB loss of a multi-layer GNN through a carefully designed graph attention mechanism. By directly optimizing the IB loss rather than relying on variational upper bounds, RGA-IB avoids restrictive Gaussian distributional and local-dependence assumptions adopted in prior IB-based graph learning methods. Extensive experiments on semi-supervised node classification under a wide range of graph adversarial attacks demonstrate that GNNs equipped with RGA-IB consistently achieve lower IB loss, indicating stronger adherence to the IB principle, and significantly outperform existing robust GNN approaches in terms of node classification accuracy.

References

Yassine Abbahaddou, Sofiane Ennadir, Johannes F. Lutzeyer, Michalis Vazirgiannis, and Henrik Boström. Bounding the expected robustness of graph neural networks subject to node feature attacks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, 2005.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Rana Ali Amjad and Bernhard C. Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(9):2225–2239, 2020.

Pranjal Awasthi, Himanshu Jain, Ankit Singh Rawat, and Aravindan Vijayaraghavan. Adversarial robustness via robust low rank representations. *Advances in Neural Information Processing Systems*, 33:11391–11403, 2020.

Pranjal Awasthi, Vaggos Chatziafratis, Xue Chen, and Aravindan Vijayaraghavan. Adversarially robust low dimensional representations. In *Conference on Learning Theory*, pp. 237–325. PMLR, 2021.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *ICLR*, 2014.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International Conference on Machine Learning*, pp. 1407–1418. PMLR, 2021.

Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *arXiv preprint arXiv:2204.08570*, 2022.

Enyan Dai, Limeng Cui, Zhengyang Wang, Xianfeng Tang, Yinghan Wang, Monica Xiao Cheng, Bing Yin, and Suhang Wang. A unified framework of graph information bottleneck for robustness and membership privacy. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 368–379, 2023a.

Enyan Dai, Limeng Cui, Zhengyang Wang, Xianfeng Tang, Yinghan Wang, Monica Xiao Cheng, Bing Yin, and Suhang Wang. A unified framework of graph information bottleneck for robustness and membership privacy. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 368–379, 2023b.

Chenhui Deng, Xiuyu Li, Zhuo Feng, and Zhiru Zhang. Garnet: Reduced-rank topology learning for robust and scalable graph neural networks. In *Learning on Graphs Conference*, pp. 3–1. PMLR, 2022.

Kaize Ding, Yancheng Wang, Yingzhen Yang, and Huan Liu. Eliciting structural and semantic global knowledge in unsupervised graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7378–7386, 2023.

Changxu Dong and Dengdi Sun. Brain network classification based on dynamic graph attention information bottleneck. *Comput. Methods Programs Biomed.*, 243:107913, 2024.

Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 169–177, 2020.

Xinyuan Fan, Feiyan Ma, Chenlei Leng, and Weichi Wu. Low-rank graphon learning for networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Junfeng Fang, Xinglin Li, Yongduo Sui, Yuan Gao, Guibin Zhang, Kun Wang, Xiang Wang, and Xiangnan He. Exgc: Bridging efficiency and explainability in graph condensation. In *Proceedings of the ACM on Web Conference 2024*, pp. 721–732, 2024.

Boyuan Feng, Yuke Wang, and Yufei Ding. Uag: Uncertainty-aware attention graph neural network for defending adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7404–7412, 2021.

Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. *Journal of Machine Learning Research*, 24(246):1–52, 2023.

Xingcheng Fu, Jian Wang, Yisen Gao, Qingyun Sun, Haonan Yuan, Jianxin Li, and Xianxian Li. Discrete curvature graph information bottleneck. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 16666–16673, 2025.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M Khapra, and Balaraman Ravindran. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017a.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30, NeurIPS 2017, December 4-9, 2017*, pp. 1024–1034, 2017b.

Xin He, Wenqi Fan, Yili Wang, Chengyi Liu, Rui Miao, Xin Juan, and Xin Wang. Graph defense diffusion model. *CoRR*, abs/2501.11568, 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Yaning Jia, Dongmian Zou, Hongfei Wang, and Hai Jin. Enhancing node-level adversarial defenses by lipschitz regularization of graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 951–963, 2023.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16049–16096, 2023.

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

Huafeng Kuang, Hong Liu, Yongjian Wu, Shin'ichi Satoh, and Rongrong Ji. Improving adversarial robustness via information bottleneck distillation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Qiuxia Lai, Yu Li, Ailing Zeng, Minhao Liu, Hanqiu Sun, and Qiang Xu. Information bottleneck approach to spatial attention learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 779–785, 2021.

John Boaz Lee, Ryan A. Rossi, Sungchul Kim, Nesreen K. Ahmed, and Eunyee Koh. Attention models in graphs: A survey. *ACM Trans. Knowl. Discov. Data*, 13(6):62:1–62:25, 2019.

Runlin Lei, Zhen Wang, Yaliang Li, Bolin Ding, and Zhewei Wei. Evennet: Ignoring odd-hop neighbors improves robustness of graph neural networks. *Advances in Neural Information Processing Systems*, 35: 4694–4706, 2022.

Jintang Li, Jiaying Peng, Liang Chen, Zibin Zheng, Tingting Liang, and Qing Ling. Spectral adversarial training for robust graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022a.

Kuan Li, Yang Liu, Xiang Ao, and Qing He. Revisiting graph adversarial attack and defense from a data distribution perspective. In *The Eleventh International Conference on Learning Representations*, 2022b.

Yixin Liu, Kaize Ding, Qinghua Lu, Fuyi Li, Leo Yu Zhang, and Shirui Pan. Towards self-interpretable graph-level anomaly detection. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Jiayi Luo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, and Jianxin Li. Robust graph learning against adversarial evasion attacks via prior-free diffusion-based structure purification. In *Proceedings of the ACM on Web Conference 2025, WWW 2025*, pp. 2098–2110, 2025.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Chenyang Qiu, Guoshun Nan, Tianyu Xiong, Wendi Deng, Di Wang, Zhiyang Teng, Lijuan Sun, Qimei Cui, and Xiaofeng Tao. Refining latent homophilic structures over heterophilic graphs for robust graph convolution networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8930–8938, 2024.

T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, pp. 18888–18909. PMLR, 2022.

Dayana Savostianova, Emanuele Zangrando, Gianluca Ceruti, and Francesco Tudisco. Robust low-rank training via approximate orthonormal constraints. *Advances in Neural Information Processing Systems*, 36:66064–66083, 2023.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.

Sangwoo Seo, Sungwon Kim, Jihyeong Jung, Yoonho Lee, and Chanyoung Park. Self-explainable temporal graph networks based on graph information bottleneck. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2572–2583, 2024.

Yang Song, Qiyu Kang, Sijie Wang, Kai Zhao, and Wee Peng Tay. On the robustness of graph neural diffusion to topology perturbations. *Advances in Neural Information Processing Systems*, 35:6384–6396, 2022.

Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and Philip S. Yu. Graph structure learning with variational information bottleneck. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 4165–4174, 2022.

Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *Proceedings of the Web Conference 2020*, pp. 673–683, 2020.

Jiliang Tang, Jimeng Sun, Chi Wang, Zhi Yang, and Huan Liu. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 807–816. ACM, 2009.

Naftali Tishby, Fernando C. N. Pereira, and William Bialek. The information bottleneck method. *CoRR*, physics/0004057, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Jihong Wang, Minnan Luo, Jundong Li, Yun Lin, Yushun Dong, Jin Song Dong, and Qinghua Zheng. Empower post-hoc graph explanations with information bottleneck: A pre-training and fine-tuning perspective. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2349–2360, 2023.

Jihong Wang, Minnan Luo, Jundong Li, Ziqi Liu, Jun Zhou, and Qinghua Zheng. Toward enhanced robustness in unsupervised graph representation learning: A graph information bottleneck perspective. *IEEE Trans. Knowl. Data Eng.*, 36(8):4290–4303, 2024.

Xin Wang, Haoyang Li, Zeyang Zhang, Haibo Chen, Tong Xiao, Kehan Li, and Wenwu Zhu. Uncertainty-aware disentangled dynamic graph attention network for out-of-distribution generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025.

Zifeng Wang, Tong Jian, Aria Masoomi, Stratis Ioannidis, and Jennifer Dy. Revisiting hilbert-schmidt information bottleneck for adversarial robustness. *Advances in Neural Information Processing Systems*, 34:586–597, 2021.

Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610*, 2019.

Qitian Wu, Wentao Zhao, Zenan Li, David P. Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022a.

Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. Difformer: Scalable (graph) transformers induced by energy constrained diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023a.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Simplifying and empowering transformers for large-graph representations. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Zhebin Wu, Lin Shu, Ziyue Xu, Yaomin Chang, Chuan Chen, and Zibin Zheng. Robust tensor graph convolutional networks via t-svd based graph augmentation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2090–2099, 2022b.

Beini Xie, Heng Chang, Ziwei Zhang, Xin Wang, Daixin Wang, Zhiqiang Zhang, Rex Ying, and Wenwu Zhu. Adversarially robust neural architecture search for graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8143–8152, 2023.

Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 30414–30425, 2021.

Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019a.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019b.

Nanqing Xu, Weiwei Feng, Tianzhu Zhang, and Yongdong Zhang. Fd-gan: Generalizable and robust forgery detection via generative adversarial networks. *International Journal of Computer Vision*, 132(12):5801–5819, 2024.

Liang Yang, Fan Wu, Zichen Zheng, Bingxin Niu, Junhua Gu, Chuan Wang, Xiaochun Cao, and Yuanfang Guo. Heterogeneous graph information bottleneck. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 1638–1645, 2021a.

Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou, Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, and David Wipf. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning*, pp. 11773–11783. PMLR, 2021b.

Junchi Yu, Jie Cao, and Ran He. Improving subgraph recognition with variational graph information bottleneck. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 19374–19383, 2022.

Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. Dynamic graph information bottleneck. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pp. 469–480, 2024.

Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. *arXiv preprint arXiv:2009.05923*, 2020.

Jianfu Zhang, Yan Hong, Dawei Cheng, Liqing Zhang, and Qibin Zhao. Defending adversarial attacks in graph neural networks via tensor enhancement. *Pattern Recognition*, 158:110954, 2025.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, pp. 5171–5181, 2018.

Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

Chu Zhao, Enneng Yang, Yuliang Liang, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. Distributionally robust graph out-of-distribution recommendation via diffusion model. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 2018–2031, 2025.

Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Adversarial robustness in graph neural networks: A hamiltonian approach. *Advances in Neural Information Processing Systems*, 2023.

Chenyu Zhou, Wei Huang, Xinyuan Miao, Yabin Peng, Xianglong Kong, Yi Cao, and Xi Chen. Fortifying graph neural networks against adversarial attacks via ensemble learning. *Knowledge-Based Systems*, 309: 112867, 2025.

Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pp. 27378–27394. PMLR, 2022.

Zhanke Zhou, Chenyu Zhou, Xuan Li, Jiangchao Yao, Quanming Yao, and Bo Han. On strengthening and defending graph reconstruction attack with markov chain approximation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42843–42877, 2023.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *KDD*, 2019.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2847–2856, 2018.

A Proof of Theorem 3.1

We need the following two lemmas before the proof of Theorem 3.1. It is noted $Z = BF$, with B being the attention weight matrix and F being the hidden node feature before applying the graph attention operation. We abbreviate $\mathrm{IB}(Z, X, Y)$ as $\mathrm{IB}(B)$.

Lemma A.1.

$$\nabla_B I(Z, X) = \nabla_Z I(Z, X) \cdot F^\top. \tag{2}$$

Lemma A.2.

$$\nabla_B I(Z, Y) = \nabla_Z I(Z, Y) \cdot F^\top. \tag{3}$$

Proof of Theorem 3.1. We note that $IB(B) = I(Z,X) - I(Z,Y)$. Then $\nabla_B \mathrm{IB}(B) = \nabla_B I(Z,X) - \nabla_B I(Z,Y)$. With Lemma A.1 and Lemma A.2, we have

$$\begin{aligned}\nabla_B \mathrm{IB}(B) &= \nabla_B I(BF,X) - \nabla_B I(BF,Y)\\ &= \nabla_Z I(Z,X)\cdot F^\top - \nabla_Z I(Z,Y)\cdot F^\top\\ &= (\nabla_Z I(Z,X) - \nabla_Z I(Z,Y))\cdot F^\top.\end{aligned}$$

Let $Q^{(\ell-1)} = \nabla_Z I(Z^{(\ell-1)},X) - \nabla_Z I(Z^{(\ell-1)},Y)$, we get $\nabla_B \mathrm{IB}(B^{(\ell-1)}) = Q^{(\ell-1)}\cdot F^\top$. Therefore, at step ℓ of gradient descent on $\mathrm{IB}(B)$, we have

$$B^{(\ell)} = B^{(\ell-1)} - \eta Q^{(\ell-1)}\cdot F^\top.$$

□

Proof of Lemma A.1.

$$\begin{aligned}I(Z,X) = &\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}\phi(Z_i,a)\phi(X_i,b)\right)\left(\ln n + \ln\sum_{j=1}^{n}\phi(Z_j,a)\phi(X_j,b)\right)\\ &-\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}\phi(Z_i,a)\phi(X_i,b)\right)\left(\ln\sum_{k=1}^{n}\phi(Z_k,a) + \ln\sum_{m=1}^{n}\phi(X_m,b)\right)\end{aligned} \tag{4}$$

Let $G_{ia} = \phi(Z_i,a)$, and $Z_{ib} = \phi(X_i,b)$, then

$$I(Z,X) = \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}G_{ia}Z_{ib}\right)\left(\ln n + \ln\sum_{j=1}^{n}G_{ja}Z_{jb} - \ln\sum_{k=1}^{n}G_{ka} - \ln\sum_{m=1}^{n}Z_{mb}\right).$$

Next, we have

$$\begin{aligned}\nabla_{Z_i} I(Z,X) = &\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}Z_{ib}U_{ia}\left(\ln n + \ln\sum_{j=1}^{n}G_{ja}Z_{jb} - \ln\sum_{k=1}^{n}G_{ka} - \ln\sum_{m=1}^{n}Z_{mb}\right)\\ &+\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}G_{ia}Z_{ib}\right)\left(\frac{Z_{ib}U_{ia}}{\sum_{j=1}^{n}G_{ja}Z_{jb}} - \frac{U_{ia}}{\sum_{k=1}^{n}G_{ka}}\right)\\ &= \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}Z_{ib}U_{ia}\left(\ln n + \ln\sum_{j=1}^{n}G_{ja}Z_{jb} - \ln\sum_{k=1}^{n}G_{ka} - \ln\sum_{m=1}^{n}Z_{mb}\right)\\ &+\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}U_{ia}\left(Z_{ib} - \frac{\sum_{i=1}^{n}G_{ia}Z_{ib}}{\sum_{k=1}^{n}G_{ka}}\right),\end{aligned} \tag{5}$$

where $\nabla_{Z_i} I(Z,X)$ is the i-th row of $\nabla_Z I(Z,X)$. Define G_{ia} as $\phi(Z_i,a)$, Z_{ib} as $\phi(X_i,b)$, and U_{ia} as $\nabla_{Z_i} G_{ia}$. Let $S_{ia} = \exp\left(-\|Z_i - C_a\|^2\right)$, U_{ia} can be computed by

$$
\begin{aligned}
U_{ia} &= \frac{-2(Z_i - C_a)S_{ia}\sum_{b=1}^{A} S_{ib} - S_{ia}\sum_{b=1}^{A} -2(Z_i - C_b)S_{ib}}{\left(\sum_{b=1}^{A} S_{ib}\right)^2} \\
&= -2\left(\frac{S_{ia}(Z_i - C_a)}{\sum_{b=1}^{A} S_{ib}} - \frac{S_{ia}}{\sum_{b=1}^{A} S_{ib}}\sum_{b=1}^{A}\frac{S_{ib}(Z_i - C_b)}{\sum_{c=1}^{A} S_{ic}}\right) \\
&= -2\left(G_{ia}(Z_i - C_a) - G_{ia}\sum_{b=1}^{A} G_{ib}(Z_i - C_b)\right) \\
&= -2G_{ia}\left(Z_i - C_a - \sum_{b=1}^{A} G_{ib}(Z_i - C_b)\right).
\end{aligned} \tag{6}
$$

Taking the value of U_{ia} back to Equation (5), we can get $\nabla_{Z_i} I(Z,X)$. Note that $\nabla_{Z_i} I(Z,X) \in \mathbb{R}^{1\times d}$. $\nabla_Z I(Z,X) \in \mathbb{R}^{n\times d}$, where the i-th row of $\nabla_Z I(Z,X)$ is $\nabla_{Z_i} I(Z,X)$. Given that $Z = BF$, we have

$$
\nabla_B I(Z,X) = \nabla_Z I(Z,X) \cdot F^\top. \tag{7}
$$

□

Proof of Lemma A.2.

$$
\begin{aligned}
I(Z,Y) = &\frac{1}{n}\sum_{a=1}^{A}\sum_{c=1}^{C}\left(\sum_{i=1}^{n}\phi(Z_i,a)\mathbb{1}_{\{y_i=b\}}\right)\left(\ln n + \ln\sum_{j=1}^{n}\phi(Z_j,a)\mathbb{1}_{\{y_j=b\}}\right) \\
&- \frac{1}{n}\sum_{a=1}^{A}\sum_{c=1}^{C}\left(\sum_{i=1}^{n}\phi(Z_i,a)\mathbb{1}_{\{y_i=b\}}\right)\left(\ln\sum_{k=1}^{n}\phi(Z_k,a) + \ln\sum_{m=1}^{n}\mathbb{1}_{\{y_m=c\}}\right)
\end{aligned} \tag{8}
$$

By replacing the value of Z_{ib} with $Z_{ib} = \mathbb{1}_{\{y_i=b\}}$, we can get the value of $\nabla_B I(Z,Y)$ following the formulation of $\nabla_B I(Z,X)$.

Let $G_{ia} \coloneqq \phi(Z_i,a)$, then

$$
\begin{aligned}
I(Z,Y) = &\frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}G_{ia}\mathbb{1}_{\{y_i=b\}}\right)\left(\ln n + \ln\sum_{j=1}^{n}G_{ja}\mathbb{1}_{\{y_j=b\}}\right) \\
&- \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\left(\sum_{i=1}^{n}G_{ia}\mathbb{1}_{\{y_i=b\}}\right)\left(\ln\sum_{k=1}^{n}G_{ka} + \ln\sum_{m=1}^{n}\mathbb{1}_{\{y_m=b\}}\right)
\end{aligned} \tag{9}
$$

Next, we have

$$
\begin{aligned}
&\nabla_{Z_i} I(Z,Y) \\
&= \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B} \mathbb{1}_{\{y_i=b\}} U_{ia} \left(\ln n + \ln \sum_{j=1}^{n} G_{ja} \mathbb{1}_{\{y_j=b\}} - \ln \sum_{k=1}^{n} G_{ka} - \ln \sum_{m=1}^{n} \mathbb{1}_{\{y_m=b\}} \right) \\
&+ \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B} \left(\sum_{i=1}^{n} G_{ia} Z_{ib} \right) \left(\frac{\mathbb{1}_{\{y_i=b\}} U_{ia}}{\sum_{j=1}^{n} G_{ja} Z_{jb}} - \frac{U_{ia}}{\sum_{k=1}^{n} G_{ka}} \right) \\
&= \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B} \mathbb{1}_{\{y_i=b\}} U_{ia} \left(\ln n + \ln \sum_{j=1}^{n} G_{ja} \mathbb{1}_{\{y_j=b\}} - \ln \sum_{k=1}^{n} G_{ka} - \ln \sum_{m=1}^{n} \mathbb{1}_{\{y_m=b\}} \right) \\
&+ \frac{1}{n}\sum_{a=1}^{A}\sum_{b=1}^{B} \left(U_{ia} \mathbb{1}_{\{y_i=b\}} - \frac{\sum_{i=1}^{n} G_{ia} \mathbb{1}_{\{y_i=b\}}}{\sum_{k=1}^{n} G_{ka}} \right), \qquad (10)
\end{aligned}
$$

where $U_{ia} = \nabla_{Z_i} G_{ia} = \nabla_{Z_i} \phi(Z_i, a) = -2G_{ia}\left(Z_i - C_a - \sum_{b=1}^{A} G_{ib}(Z_i - C_b)\right)$. Taking the value of U_{ia} back to Equation (10), we can get $\nabla_{Z_i} I(Z,Y)$. Note that $\nabla_{Z_i} I(Z,Y) \in \mathbb{R}^{1\times d}$. $\nabla_Z I(Z,Y) \in \mathbb{R}^{n\times d}$, where the i-th row of $\nabla_Z I(Z,Y)$ is $\nabla_{Z_i} I(Z,Y)$. Given that $Z = BF$, we have

$$
\nabla_B I(Z,Y) = \nabla_Z I(Z,Y) \cdot F^\top. \qquad (11)
$$

□

B More Experiment Settings

B.1 Datasets

Following previous works on adversarial attacks and defense of GNNs (Jin et al., 2020; Zügner & Günnemann, 2019; Entezari et al., 2020), we evaluate RGA-IB on four public benchmark datasets for node classification, including three citation graphs, which are Cora, Citeseer, and Pubmed, and one blog graph, that is, Polblogs. Following previous works on graph adversarial attacks, we evaluate our method and baselines on the largest connected component (LCC) of the graphs. We show the statistics of the datasets in Table 6.

Table 6: Statistics of Cora, Citeseer, Polblogs, and Pubmed.

Datasets	# Node	# Edge	Classes	Features
Cora	2,485	5,069	7	1,433
Citeseer	2,110	3,668	6	3,703
Polblogs	1,222	16,714	2	1,222
Pubmed	19,717	44,338	3	500
Chameleon	2,277	36,101	5	2,325
Actor	7,600	30,019	5	931
ogbn-arxiv	169,343	1,166,243	40	128
Reddit	232,965	11,606,919	41	602

B.2 Additional Implementation Details

In our experiments for semi-supervised node classification, we search for the optimal values of different hyper-parameters, including learning rate, weight decay, hidden dimension, and dropout rate, by 5-fold cross-validation on the training data of each dataset. We search for the learning rate from $\{1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}, 3\times10^{-2}, 6\times10^{-2}, 1\times10^{-1}, 5\times10^{-1}\}$. We search for weight decay from $\{1\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}\}$. We search for the hidden dimension from $\{32, 64, 128, 256, 512\}$. The dropout rate is selected from $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. Values leading to the lowest

validation loss are selected for each dataset. Selected values of learning rate, weight decay, hidden dimension, and dropout rate on different datasets are shown in Table 7. All experiments in this paper are performed on a single NVIDIA Tesla V100 GPU.

Table 7: Selected values of learning rate, weight decay, hidden dimension, and dropout rate.

Hyper-parameters	Cora	Citeseer	PubMed	Polblogs	Chameleon	Actor	ogbn-arxiv	Reddit
Learning Rate	3×10^{-2}	3×10^{-2}	1×10^{-3}	6×10^{-2}	3×10^{-2}	1×10^{-2}	1×10^{-2}	5×10^{-3}
Weight Decay	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-5}	5×10^{-4}	5×10^{-4}	5×10^{-5}	5×10^{-5}
Hidden Dimension	96	128	64	128	128	128	256	256
Dropout Rate	0.5	0.5	0.7	0.4	0.6	0.6	0.3	0.3

B.3 Attack Settings

Non-targeted Adversarial Attacks (Metattack) (Zügner & Günnemann, 2019). We first evaluate the robustness of our method against the non-targeted adversarial attack method Metattack. Metattack treats the graph as a hyperparameter to optimize and uses the meta-gradients to solve the bi-level optimization problem, which minimizes the node classification accuracy. We follow the implementation in (Zügner & Günnemann, 2019). As Metattack has several variants, we follow (Jin et al., 2020) and adopt the most destructive attack version, Meta-Self, on Cora, Citeseer, and Polblogs datasets. On Pubmed, we adopt the approximate version of Meta-Self, A-Meta-Self, following the settings in (Jin et al., 2020). We measure the strength of the attack by the perturbation rate, which is the ratio of perturbed edges among all the edges in the graph. We evaluate the RGA-IB and all the baselines with the perturbation rates ranging from 0 to 25% with a step of 5%.

Targeted Adversarial Attack (Nettack) (Zügner et al., 2018). We adopt Nettack as the targeted attack method in evaluating the robustness of our method. Nettack manipulates the edges and node features to degrade the classification accuracy on the target nodes while minimizing the change in the graph's degree distribution and the feature co-occurrences. We use the default attack settings in the original implementation in (Zügner et al., 2018). The nodes in the test set whose degree is larger than 10 are set as the target nodes. The number of perturbations made on every targeted node is defined as the attack budget. Following (Jin et al., 2020), we evaluate the RGA-IB and all the baselines with attack budgets ranging from 1 to 5 with a step size of 1. Following the settings in (Jin et al., 2020), we only sample 10% of the target nodes for the evaluation on the Nettack.

Gradient-based Attack Method (Topology Attack) (Xu et al., 2019a). We evaluate the robustness of the RGA-IB against a gradient-based attack method named Topology Attack (Xu et al., 2019a). The Topology Attack proposes a projected gradient descent (PGD) adversarial attack method that perturbs the edges in a graph to degrade the overall node classification accuracy on the perturbed graph. We adopt the same settings in (Xu et al., 2019a) to attack the graphs in our experiments. Similar to the settings for Metattack, we measure the strength of the attack by the perturbation rate, which is the ratio of the perturbed edges among all the edges in the graph. We evaluate the RGA-IB and all the baselines with perturbation rates ranging from 0 to 25% with a step of 5%.

C Additional Experiment Results

C.1 Training Time Evaluation

In this section, we evaluate the training time of the proposed RGA-IB in comparison to the widely used attention-based GNN, GAT (Veličković et al., 2018), under a controlled setting with the same number of attention layers and identical hidden dimensions. The training time is evaluated on a single NVIDIA V100 GPU. Both models are trained for 500 epochs on all the evaluated datasets. The comparisons for Cora, Citeseer, Polblogs, Pubmed are conducted on graphs under Topology Attack (Xu et al., 2019a) with a perturbation rate of 25%. The comparisons for Chameleon and Actor are conducted on graphs under Metattack (Zügner & Günnemann, 2019) with a perturbation rate of 25%. The comparisons for ogbn-arxiv

and Reddit are conducted on graphs attacked by DICE (Li et al., 2022b) with a perturbation rate of 20%. As shown in Table 8, the training time of RGA-IB is comparable to that of GAT, while RGA-IB significantly outperforms GAT on all the evaluated datasets. For instance, RGA-IB requires only a marginal additional training overhead of 5.8% compared to GAT on the large-scale graph dataset, ogbn-arxiv, while significantly outperforming GAT by 2.7% in node classification accuracy under adversarial attacks, demonstrating the efficiency and effectiveness of RGA-IB in learning robust node representations through the novel IB-based graph attention method.

Table 8: Training time comparison between RGA-IB and standard GAT with the same number of graph attention layers and identical hidden dimensions.

Model	Metric	Cora	Citeseer	Polblogs	Pubmed	ogbn-arxiv	Reddit	Chameleon	Actor
GAT	Training Time (s/epoch)	0.244	0.288	0.268	0.472	67.2	58.2	0.532	0.588
	Accuracy (%)	65.5	59.1	64.2	68.2	56.1	76.3	66.5	37.2
RGA-IB	Training Time (s/epoch)	0.264	0.308	0.288	0.508	70.4	61.6	0.568	0.612
	Accuracy (%)	**71.0**	**66.7**	**71.9**	**73.9**	**58.9**	**78.0**	**70.8**	**39.4**

C.2 Adversarial Defense on Large-Scale Graphs

To evaluate the scalability of the proposed RGA-IB, we conduct additional experiments for graph adversarial defense on large-scale graph datasets, including the large-scale social network, Reddit (Hamilton et al., 2017b), and the large-scale knowledge graph, ogbn-arxiv (Hu et al., 2020). The statistics of the Reddit and ogbn-arxiv datasets are detailed in Table 6. Since Metattack (Zügner & Günnemann, 2019), Nettack (Zügner et al., 2018), and Topology Attack (Xu et al., 2019a) do not scale to large-scale social graphs, we instead employ a scalable graph adversarial attack method, DICE (Li et al., 2022b), for large-scale experiments. Following the experimental protocol in (Li et al., 2022b), we evaluate all methods under perturbation rates of 0%, 10%, and 20%. It is observed in Table 9 that RGA-IB consistently outperforms all competing baselines across all perturbation levels on both datasets. For example, RGA-IB outperforms the best baseline method by 1.7% on ogbn-arxiv with a perturbation rate of 20%, demonstrating its effectiveness in defending against adversarial attacks even on large-scale graphs.

Table 9: Node classification performance (Accuracy±Std) under the adversarial attack by DICE (Li et al., 2022b) on large-scale graphs. The best results are shown in bold, and the second-best results are underlined. Improvements of RGA-IB over the strongest baseline are reported in parentheses.

Dataset	Ptb Rate	IB-D^{2}GAT	DGIB	CurvGIB	HANG	RGIB	RM-GIB	Differmer	GAR	GCORNs	Pro-GNN	RGA-IB (Ours)
ogbn-arxiv	0	73.6±1.3	74.8±1.1	73.1±1.4	73.8±1.2	72.9±1.3	74.1±1.2	74.3±1.3	73.9±1.4	73.5±1.2	74.2±1.3	**76.4** (↑ 1.6)±1.2
	10	63.4±1.4	63.0±1.5	63.9±1.3	62.8±1.4	64.1±1.3	64.3±1.4	63.5±1.5	63.2±1.3	64.6±1.2	64.0±1.4	**66.1** (↑ 1.5)±1.3
	20	56.1±1.5	55.7±1.6	56.5±1.4	55.3±1.5	56.8±1.4	57.0±1.5	57.2±1.3	56.0±1.6	55.8±1.4	56.7±1.5	**58.9** (↑ 1.7)±1.4
Reddit	0	94.1±1.1	93.8±1.2	94.3±1.0	93.5±1.1	94.6±1.0	94.8±1.1	95.1±1.0	94.2±1.2	94.0±1.1	94.7±1.1	**96.7** (↑ 1.6)±1.0
	10	83.4±1.3	83.0±1.4	83.7±1.2	82.8±1.3	84.1±1.2	84.3±1.3	84.6±1.2	83.5±1.4	83.2±1.3	84.0±1.3	**85.9** (↑ 1.3)±1.2
	20	75.2±1.5	74.8±1.6	75.5±1.4	74.6±1.5	75.9±1.4	76.1±1.5	75.0±1.4	75.3±1.6	76.4±1.3	75.8±1.5	**78.0** (↑ 1.6)±1.4

C.3 Adversarial Defense on Heterophilic Graphs

In this section, we evaluate the robustness of the proposed RGA-IB on heterophilic graphs under adversarial attacks. Following (Qiu et al., 2024), we conduct experiments on the heterophilic graph datasets Chameleon (Pei et al., 2020) and Actor (Tang et al., 2009) under the non-targeted poisoning attack Metattack (Zügner & Günnemann, 2019) with perturbation rates of 15% and 25%. It is observed in Table 10 that RGA-IB consistently outperforms existing robust GNN baselines across different perturbation rates on both datasets. For instance, RGA-IB outperforms the strongest baseline by 2.6% on Actor under a perturbation rate of 25%, demonstrating the effectiveness of the proposed RGA-IB in defending against adversarial perturbations on heterophilic graphs.

Table 10: Node classification performance (Accuracy±Std) under the non-targeted poisoning attack (Metattack) (Zügner & Günnemann, 2019) on heterophilic graphs. The best results are shown in bold, and the second-best results are underlined. Improvements of RGA-IB over the strongest baseline are reported in parentheses.

Dataset	Ptb Rate	IB-D^{2}GAT	DGIB	CurvGIB	HANG	RGIB	RM-GIB	Differmer	GAR	GCORNs	Pro-GNN	RGA-IB (Ours)
Chameleon	0	71.2±1.4	70.9±1.6	72.0±1.5	69.8±1.3	70.6±1.7	71.0±1.6	70.1±1.4	72.4±1.2	71.8±1.3	72.2±1.1	**73.9** (↑ 1.5) ±1.4
	15	69.4±1.6	69.1±1.8	70.1±1.7	68.9±1.4	69.2±1.9	69.8±1.7	68.7±1.6	69.9±1.5	70.5±1.6	70.3±1.4	**72.1** (↑ 1.6) ±1.6
	25	67.0±1.9	66.8±2.0	68.1±1.8	66.5±1.7	66.9±2.1	67.6±1.9	65.9±2.0	68.5±1.8	68.0±1.9	68.2±1.7	**70.8** (↑ 2.3) ±1.8
Actor	0	39.6±1.5	39.1±1.6	40.4±1.4	38.7±1.3	39.4±1.7	39.9±1.6	38.9±1.5	40.7±1.4	40.3±1.5	40.5±1.3	**42.9** (↑ 2.2) ±1.4
	15	37.8±1.4	37.5±1.5	38.4±1.6	36.9±1.2	37.6±1.8	38.1±1.6	36.8±1.4	38.3±1.5	38.9±1.6	38.6±1.4	**41.1** (↑ 2.2) ±1.5
	25	35.6±1.8	35.2±1.9	36.4±1.7	34.8±1.6	35.4±2.0	36.0±1.8	34.5±1.9	36.8±1.7	36.5±1.8	36.6±1.6	**39.4** (↑ 2.6) ±1.7

C.4 Additional Experiment Results for Semi-Supervised Node Classification Under Adversarial Attacks

We compare RGA-IB with additional baseline methods including, GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), and GIB (Wu et al., 2020) for Metattack (Zügner & Günnemann, 2019), Nettack (Zügner et al., 2018), and Topology Attack (Xu et al., 2019a) in Table 11, Table 12, and Table 13. It is observed that RGA-IB consistently outperforms all baseline methods across all datasets and perturbation levels under the three adversarial attack settings. In particular, as the perturbation rate increases, the performance gap between RGA-IB and competing methods becomes more pronounced, indicating stronger robustness under severe structural corruption.

Table 11: Node classification performance (Accuracy±Std) under non-targeted attack Metattack (Zügner & Günnemann, 2019). Comparison between RGA-IB and additional baseline methods, including GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a).

Dataset	Ptb Rate	GCN	GAT	RGCN	UAG	GIB	IB-D^{2}GAT	RM-GIB	RGA-IB (Ours)
Cora	0	83.5±0.4	84.0±0.7	83.1±0.4	82.1±0.5	82.2±0.7	83.1±0.5	82.1±0.6	**85.0** ±1.3
	5	76.5±0.4	80.4±0.7	77.4±1.3	79.1±1.2	78.7±1.2	82.7±1.2	78.3±1.2	**83.9** ±1.9
	10	70.3±1.3	75.6±1.8	72.2±1.3	75.1±1.8	74.9±1.8	77.9±1.7	71.4±1.8	**81.2** ±1.2
	15	65.1±1.7	69.7±2.7	66.8±1.3	71.0±1.3	71.0±1.5	76.0±1.6	66.6±1.5	**77.9** ±1.7
	20	59.5±1.0	59.9±2.3	59.2±2.2	65.7±2.2	66.7±1.8	68.7±1.6	58.9±1.8	**74.5** ±1.1
	25	47.5±1.1	54.7±2.5	50.5±2.4	60.8±2.4	66.5±2.2	56.5±2.2	52.0±2.2	**71.0** ±1.1
Citeseer	0	72.0±0.6	73.3±0.8	71.2±0.8	72.1±0.6	71.5±0.2	72.5±0.2	71.8±0.1	**74.5** ±1.0
	5	70.8±0.6	72.8±0.9	70.5±0.8	70.5±0.8	68.3±1.3	71.3±1.7	68.8±1.3	**74.0** ±2.0
	10	67.5±1.3	70.6±1.4	67.7±1.9	69.5±1.9	66.6±1.4	71.6±1.6	68.8±1.4	**73.7** ±1.7
	15	64.5±1.4	69.0±1.5	65.6±1.7	65.9±1.7	65.8±1.2	69.8±1.4	63.2±1.2	**73.2** ±1.7
	20	62.0±1.4	61.0±1.4	62.4±2.0	59.3±2.0	64.2±1.4	64.2±1.7	58.5±1.4	**71.4** ±1.9
	25	56.9±1.4	61.8±2.2	55.3±2.4	59.8±2.4	63.2±2.2	66.2±2.5	57.1±2.2	**69.9** ±1.9
Polblogs	0	95.7±0.4	95.4±0.2	95.2±0.1	90.1±2.2	95.4±0.8	95.1±0.9	95.5±0.8	**96.5** ±1.1
	5	73.0±0.4	83.6±0.2	74.3±0.1	84.8±0.1	92.8±0.2	92.8±0.8	92.0±0.2	**94.0** ±1.2
	10	70.7±1.7	76.3±1.4	71.0±1.9	75.5±1.9	88.5±1.3	88.5±1.3	88.2±1.3	**91.1** ±1.4
	15	64.9±1.7	68.8±1.1	67.2±1.8	66.4±1.8	84.4±1.9	80.4±1.8	78.1±1.9	**87.1** ±1.3
	20	51.2±1.0	51.5±1.2	59.8±1.2	58.1±1.2	74.1±1.3	78.1±1.7	77.3±1.3	**80.8** ±1.3
	25	49.2±2.4	51.1±2.1	56.0±2.1	55.3±2.1	67.3±1.4	66.3±1.9	66.6±1.4	**69.9** ±1.6
Pubmed	0	87.2±0.1	83.7±0.4	86.2±0.2	87.1±0.1	85.7±0.1	85.2±0.2	85.3±0.1	**88.4** ±1.3
	5	83.0±0.1	78.0±0.4	81.0±0.2	86.3±0.2	84.9±0.2	83.9±1.1	83.4±0.2	**88.1** ±1.9
	10	81.2±1.4	74.9±1.6	77.5±1.4	85.7±1.4	83.6±1.3	84.6±1.3	83.2±1.3	**88.5** ±1.7
	15	78.6±1.6	71.1±1.6	73.9±1.6	84.7±1.6	82.8±1.2	84.8±1.3	83.1±1.2	**88.2** ±1.2
	20	77.3±0.7	68.2±1.7	71.1±1.5	83.8±1.5	81.8±1.5	84.8±2.1	83.0±1.5	**88.0** ±1.0
	25	75.5±2.0	65.4±1.0	67.9±2.2	83.6±2.2	80.0±2.3	84.0±2.2	82.7±2.3	**87.8** ±1.3

Table 12: Node classification performance (Accuracy±Std) under targeted attack Nettack (Zügner et al., 2018). Comparison between RGA-IB and additional baseline methods, including GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), and GIB (Wu et al., 2020).

Dataset	Budget	GCN	GAT	RGCN	UAG	GIB	IB-D^{2}GAT	RM-GIB	RGA-IB (Ours)
Cora	0	81.4±1.0	82.1±1.1	81.3±1.6	82.3±1.1	81.9±2.8	81.0±1.5	83.1±2.2	**86.1** ±1.1
	1	75.1±1.0	76.0±2.1	76.8±1.7	81.8±1.2	80.2±2.8	78.5±2.3	80.6±2.3	**85.5** ±1.4
	2	70.6±1.1	70.2±1.4	71.0±1.1	78.0±1.8	77.2±2.2	76.2±2.4	77.9±1.3	**80.7** ±1.2
	3	68.0±1.7	65.5±1.3	66.5±1.6	72.5±1.1	72.9±3.4	72.5±2.1	72.9±1.1	**74.7** ±1.4
	4	61.6±1.5	61.7±0.9	59.3±2.7	70.3±2.5	69.9±2.1	71.7±2.1	70.9±2.1	**73.6** ±1.2
	5	55.5±1.7	58.3±2.0	55.3±1.7	66.3±1.2	66.4±2.4	66.6±2.5	67.9±2.4	**69.9** ±1.8
Citeseer	0	81.0±1.4	82.3±2.0	80.3±1.1	81.3±1.1	80.6±3.0	80.9±1.0	81.6±3.0	**83.4** ±1.7
	1	78.4±1.6	81.3±1.4	78.3±0.7	80.3±0.7	79.1±2.8	79.0±2.0	80.1±2.8	**82.9** ±1.9
	2	74.9±3.5	77.4±4.9	75.4±2.0	79.4±2.0	78.3±3.2	78.1±1.6	79.3±1.4	**82.5** ±1.4
	3	64.0±3.7	60.9±3.0	60.3±1.2	78.3±1.2	78.7±3.3	77.6±1.9	78.7±1.5	**81.0** ±1.6
	4	55.4±2.6	61.6±4.6	55.5±1.8	77.5±1.8	77.7±6.5	77.9±2.0	78.1±2.1	**79.5** ±1.6
	5	47.6±5.2	55.6±6.3	47.4±2.0	71.4±2.0	71.0±4.6	72.0±2.8	72.0±1.6	**74.6** ±1.9
Polblogs	0	97.0±0.2	97.3±0.3	97.0±0.1	97.1±0.1	97.2±0.8	97.4±0.4	97.3±0.8	**98.2** ±1.6
	1	96.8±0.2	97.2±0.3	97.0±0.1	96.0±0.1	97.1±0.3	97.3±0.4	97.2±0.5	**98.1** ±1.9
	2	95.6±0.2	96.1±0.7	95.9±0.2	97.0±0.2	95.3±0.7	96.1±0.4	95.9±0.3	**97.8** ±1.5
	3	95.4±0.2	95.8±0.6	95.6±0.3	96.6±0.3	95.4±0.1	96.1±0.3	95.9±0.1	**97.5** ±1.1
	4	94.2±0.2	94.8±0.7	94.4±0.3	96.1±0.3	94.9±0.6	95.6±0.6	95.3±0.5	**97.4** ±1.3
	5	93.0±0.5	93.3±1.4	93.2±0.4	95.6±0.4	93.2±0.4	94.6±0.4	94.6±0.4	**97.2** ±1.5
Pubmed	0	88.1±1.4	87.0±1.1	84.9±1.4	87.3±1.4	85.9±1.1	85.4±1.1	86.8±1.1	**89.6** ±1.9
	1	87.0±1.6	85.0±2.1	83.8±1.7	86.6±1.7	84.6±2.9	84.5±1.8	85.5±2.9	**88.8** ±1.4
	2	84.1±3.5	83.9±1.4	84.2±1.1	85.2±1.1	84.3±3.5	84.2±2.8	85.3±2.5	**87.0** ±1.3
	3	81.3±3.7	81.3±1.3	82.5±1.6	83.9±1.6	84.9±3.3	84.2±3.2	84.2±1.3	**85.6** ±1.9
	4	76.4±2.6	78.5±0.9	80.0±2.7	80.0±2.3	79.4±3.2	78.8±3.2	79.9±1.2	**81.6** ±1.2
	5	68.3±5.2	74.3±2.0	75.1±3.7	75.2±1.4	70.2±1.6	72.9±2.0	70.6±1.6	**76.8** ±1.3

Table 13: Node classification performance (Accuracy±Std) under Topology Attack (Xu et al., 2019a). Comparison between RGA-IB and additional baseline methods, including GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), RGCN (Zhu et al., 2019), UAG (Feng et al., 2021), GIB (Wu et al., 2020), IB-D^{2}GAT (Wang et al., 2025), and RM-GIB (Dai et al., 2023a).

Dataset	Ptb Rate	GCN	GAT	RGCN	UAG	GIB	IB-D^{2}GAT	RM-GIB	RGA-IB (Ours)
Cora	0	83.5±0.4	84.0±0.7	83.1±0.4	82.1±0.5	82.2±0.7	83.1±0.5	82.1±0.6	**85.0** ±1.3
	5	75.5±0.4	77.0±0.7	75.0±1.3	76.2±1.2	75.8±1.2	76.1±1.2	75.3±1.2	**79.6** ±1.9
	10	72.0±1.3	74.1±1.8	73.1±1.3	76.0±1.8	74.0±1.8	74.9±1.7	73.0±1.8	**79.3** ±1.2
	15	69.4±1.7	70.6±2.7	71.6±1.3	74.9±1.3	72.9±1.5	74.0±1.6	72.4±1.5	**76.9** ±1.7
	20	67.6±1.0	68.8±2.3	67.9±2.2	71.9±2.2	70.0±1.8	70.9±1.6	69.6±1.8	**74.5** ±1.1
	25	64.8±1.1	65.5±2.5	66.0±2.4	68.0±2.4	67.9±2.2	68.2±2.2	67.2±2.2	**71.0** ±1.1
Citeseer	0	72.0±0.6	73.3±0.8	71.2±0.8	72.1±0.6	71.5±0.2	72.5±0.2	71.8±0.1	**74.5** ±1.0
	5	68.0±0.6	68.2±0.9	68.2±0.8	71.3±0.8	68.7±1.3	69.6±1.7	67.7±1.3	**72.7** ±2.0
	10	64.1±1.3	66.7±1.4	65.9±1.9	67.9±1.9	66.6±1.4	66.6±1.6	65.6±1.4	**69.7** ±1.7
	15	61.1±1.4	63.6±1.5	64.6±1.7	66.4±1.7	65.9±1.2	65.9±1.4	64.9±1.2	**68.5** ±1.7
	20	60.3±1.4	61.9±1.4	62.9±2.0	64.0±2.0	64.7±1.4	65.5±1.7	64.5±1.4	**67.4** ±1.9
	25	59.0±1.4	59.1±2.2	60.9±2.4	62.9±2.4	63.2±2.2	64.3±2.5	64.2±2.2	**66.7** ±1.9
Polblogs	0	95.7±0.4	95.4±0.2	95.2±0.1	90.1±2.2	95.4±0.8	95.1±0.9	95.5±0.8	**96.5** ±1.1
	5	87.7±0.4	88.4±0.2	89.2±0.1	90.3±0.1	90.2±0.2	90.4±0.8	90.5±0.2	**92.0** ±1.2
	10	84.6±1.7	85.9±1.4	85.9±1.9	87.0±1.9	86.0±1.3	86.2±1.3	85.8±1.3	**89.1** ±1.4
	15	71.6±1.7	72.0±1.1	72.1±1.8	81.9±1.8	79.5±1.9	80.8±1.8	79.2±1.9	**85.0** ±1.3
	20	65.0±1.0	67.1±1.2	67.3±1.2	71.5±1.2	69.4±1.3	70.9±1.7	69.2±1.3	**73.8** ±1.3
	25	64.0±2.4	64.2±2.1	66.1±2.1	69.0±2.1	67.2±1.4	69.0±1.9	65.2±1.4	**71.9** ±1.6
Pubmed	0	87.2±0.1	83.7±0.4	86.2±0.2	87.1±0.1	85.7±0.1	85.2±0.2	85.3±0.1	**88.4** ±1.3
	5	79.2±0.1	80.7±0.4	80.2±0.2	82.4±0.2	81.4±0.2	81.7±1.1	81.2±0.2	**84.2** ±1.9
	10	75.4±1.4	76.9±1.6	77.9±1.4	80.5±1.4	80.3±1.3	80.4±1.3	80.1±1.3	**82.5** ±1.7
	15	72.0±1.6	72.3±1.6	75.8±1.6	76.9±1.6	75.2±1.2	76.7±1.3	76.0±1.2	**78.7** ±1.2
	20	68.5±0.7	69.0±1.7	70.0±1.5	71.6±1.5	72.0±1.5	72.7±2.1	73.0±1.5	**74.6** ±1.0
	25	69.0±2.0	68.2±1.0	69.0±2.2	70.9±2.2	71.0±2.3	72.1±2.2	71.5±2.3	**73.9** ±1.3