RGMoE: A Unified Robust Graph Mixture of Experts to Defend Various Graph Adversarial Attacks

Anonymous authors
Paper under double-blind review

Abstract

Graph Neural Networks (GNNs) have achieved great success in modeling graph-structured data. However, recent research has revealed that GNNs are highly vulnerable to adversarial attacks, including manipulation attacks, node injection attacks, and the emerging threat of backdoor attacks. Although a variety of defense strategies have been proposed, most existing methods are tailored to a single attack type and lack a unified framework capable of defending against multiple threats simultaneously. To address this, we leverage the flexibility of the Mixture of Experts (MoE) architecture to design a unified and scalable framework to defend against various graph adversarial attacks. Our analysis shows that the inherent diversity of MoE allows some experts to be naturally unaffected by specific attacks. However, the limited number of robust experts and how to route the perturbed samples to them remain open problems. To this end, we propose the Robust Graph MoE (**RGMoE**), which introduces a novel logic diversity loss to encourage individual experts to focus on distinct neighborhood structures during decision making, thus ensuring a sufficient subset of experts to be unaffected by local structural perturbations. Moreover, RGMoE incorporates a robustness-aware router that identifies perturbed nodes and adaptively routes them to the corresponding unaffected experts, thereby enabling robust predictions for attacked samples. Extensive experiments on large-scale graphs demonstrate the effectiveness of RGMoE against multiple graph adversarial attacks. Our code is available at `https://anonymous.4open.science/r/RGMoE-F870`.

1 Introduction

Graphs are crucial data structures in numerous domains, such as social networks Fan et al. (2019), molecular graphs Mansimov et al. (2019), and finance systems Cheng et al. (2022). In recent years, Graph Neural Networks (GNNs) have shown great success in learning on graphs. With the message-passing mechanism, node representations can capture information from the graph structure, benefiting various downstream tasks such as social networks, drug discovery, and finance systems Fan et al. (2019); Zhou et al. (2020); Scarselli et al. (2008); Gao et al. (2023).

Although promising results have been achieved, recent studies have shown that GNNs are vulnerable to adversarial attacks Jin et al. (2021b); Sun et al. (2022). In other words, the predictions of GNNs can be significantly degraded or deliberately manipulated by unnoticeable perturbations in graphs. Based on the way how the graph is perturbed, adversarial attacks can be broadly categorized into manipulation attacks, node injection attacks, and the recent emerging backdoor attacks Dai et al. (2024b). Representative examples of manipulation attacks include Nettack Zügner et al. (2018) and PRBCD Geisler et al. (2021), which mislead GNNs into producing incorrect predictions on target nodes by adding or deleting a small number of edges in the training graph. TDGIA Zou et al. (2021) is a typical node injection attack that significantly degrades global node classification performance through the injection of a limited number of labeled fake nodes. More recently, backdoor attacks such as GTA Xi et al. (2021), UGBA Dai et al. (2023b), and DPGBA Zhang et al. (2024a) enable adversaries to control model predictions by injecting carefully designed triggers (e.g., specific nodes or subgraphs) into the training graph. During inference, the presence of such backdoor triggers can induce predefined model behaviors.

To address the susceptibility of GNNs to adversarial attacks, many defense methods have been proposed, including adversarial training Xu et al. (2019), graph structure denoising Jin et al. (2020); Dai et al. (2021), robust aggregation Chen et al. (2021); Geisler et al. (2021); Jin et al. (2021a), and backdoor cleaning Dai et al. (2023b); Zhang et al. (2024a;b). However, existing defenses still have two major limitations: (i) *Incomplete coverage of attacks*: As shown in Fig. 1, most existing defenses rely on attack-tailored designs to robustify a **single** GNN model. This inherently limits their ability to defend against different types of attacks due to the significant differences among perturbations, such as backdoor triggers and malicious nodes, let alone providing a unified framework to defend against them simultaneously. (ii) *Lack of scalability*: Many state-of-the-art defense methods against graph adversarial attacks, such as Pro-GNN Jin et al. (2020) and RS-GNN Dai et al. (2022), require dense adjacency matrix computations with a time complexity of $O(N^2)$. Therefore, these methods exhibit poor scalability to large-scale graph datasets. Given the above limitations, it is essential to design a unified and scalable framework that can effectively defend against manipulation attacks, node injection attacks, and backdoor attacks simultaneously.

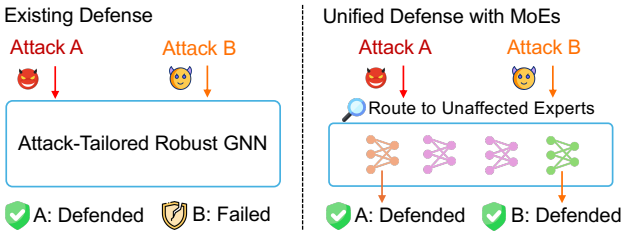


Figure 1: Illustration of existing defense and ours. By training some unaffected experts and routing perturbed nodes to them, MoE enables a unified defense against various attacks.

One promising solution for defending against various perturbations is to adopt the Mixture of Experts (MoE) Jacobs et al. (1991); Wang et al. (2023); Yuan et al. (2024) architecture, which comprises **a group of** GNN models. *Firstly*, pioneering work has shown that the diversity of graph experts in MoE Liu et al. (2023); Wu et al. (2024) can be leveraged to learn more generalizable representations. Inspired by the diversity of experts in MoE, some experts may rely on features unaffected by adversarial attacks, making them naturally robust to certain perturbations. As shown by our preliminary analysis in Fig. 2a, in a vanilla graph MoE model, around 20% of experts are naturally unaffected by backdoor perturbations Dai et al. (2023b) on an OGB-Arxiv graph and have very low attack success rates. Similarly, we observe that some experts are naturally more resistant to the edge manipulation Geisler et al. (2021) and node injection Zou et al. (2021) attacks (See Fig. 2b and Fig. 2c, respectively). *Secondly*, the flexibility of the routing mechanism allows perturbed nodes to be routed to experts resistant to malicious perturbations, thereby enabling accurate predictions for those attacked samples. Therefore, as illustrated in Fig. 1, it is promising to learn a unified Graph MoE to effectively defend against various adversarial attacks by training a subset of unaffected experts and routing the perturbed samples to them. Moreover, due to sparse routing in MoE, this robustness can be achieved without significantly increasing inference-time costs.

Two technical challenges remain to be addressed for successfully applying Graph MoE for defending against various adversarial attacks: (**i**) As illustrated by our preliminary analysis in Fig. 2a, the majority of experts still rely heavily on features influenced by adversarial perturbations, making it difficult to be activated if the number is rather limited. Thus, how can we further diversify the decision logics of the experts to ensure a sufficient number of robust experts that are unaffected by each conducted adversarial attack? (**ii**) How can we accurately route perturbed samples to the corresponding robust experts to obtain the final robust predictions? To this end, we propose the Robust Graph MoE (**RGMoE**), a unified MoE-based defense framework against variou graph adversarial attacks. Specifically, RGMoE incorporates a novel logic diversity loss to encourage individual experts to focus on distinct neighbor nodes during decision making, thereby promoting diverse decision logics and ensuring the existence of sufficient robust experts under each conducted adversarial perturbation. Additionally, RGMoE trains a robustness-aware router that adaptively routes the perturbed nodes to the corresponding expert robust to that specific perturbation. By jointly leveraging logic diversity and robustness-aware routing, RGMoE can effectively defend against a wide range of adversarial attacks while keeping the modest inference-time costs. Our main contributions are summarized as follows:

- We investigate a novel problem of simultaneously defending against various adversarial attacks (i.e., backdoor, manipulation, and node injection attacks) within one framework.

- We propose a unified RGMoE that combines a logic diversity loss with a robustness-aware router, enabling robustness by training unaffected experts and routing perturbed nodes to them.
- Extensive experiments against various adversarial attacks on large-scale graphs demonstrate the effectiveness of our RGMoE.

2 Preliminary Analysis

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote a graph, where $\mathcal{V}$ is the set of N nodes, $\mathcal{E} \subseteq |\mathcal{V}| \times |\mathcal{V}|$ denotes edges, and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ denotes node features. The node set $\mathcal{V}$ can be partitioned into three subsets: $\mathcal{V}_C$ for clean nodes, $\mathcal{V}_P$ for perturbed nodes, and $\mathcal{V}_U$ for unlabeled nodes. The target label for $\mathcal{V}_P$ (if modified) is denoted as y_t. The representation of a node v is $\mathbf{h}_v$ and $\mathcal{N}(v)$ denotes its neighbors.

2.1 Preliminaries

In this subsection, we briefly introduce *graph adversarial attacks* and *Mixture of Experts*.

2.1.1 Graph Adversarial Attacks

Extensive studies have shown that GNNs are vulnerable to adversarial attacks, i.e., the predictions of GNNs can be greatly degraded or manipulated by small perturbations in graphs. A unified formulation of graph adversarial attacks on node classification can be given as:

$$\begin{aligned} &\min_{\hat{\mathcal{G}} \in \Phi(\mathcal{G})} \mathcal{L}_{\text{atk}}(\hat{\mathcal{G}}) = \sum_{v \in \mathcal{V}_P} \ell_{\text{atk}}\big(f_{\theta^*}(\hat{\mathcal{G}})_v,\, y_t\big) \\ &\text{s.t.} \quad \theta^* = \arg\min_{\theta} \mathcal{L}_{\text{train}}\big(f_\theta(\hat{\mathcal{G}})\big), \end{aligned} \tag{1}$$

where l_{atk} is the loss for attacking, $\mathcal{L}_{train}$ is the loss for training the model. $\hat{\mathcal{G}}$ is the perturbed graph and $\Phi(\mathcal{G})$ corresponds to the way of perturbing the graph. Based on different perturbations Φ, the adversarial attacks can be categorized into manipulation attacks, node injection attacks, and the emerging backdoor attacks Dai et al. (2024b):

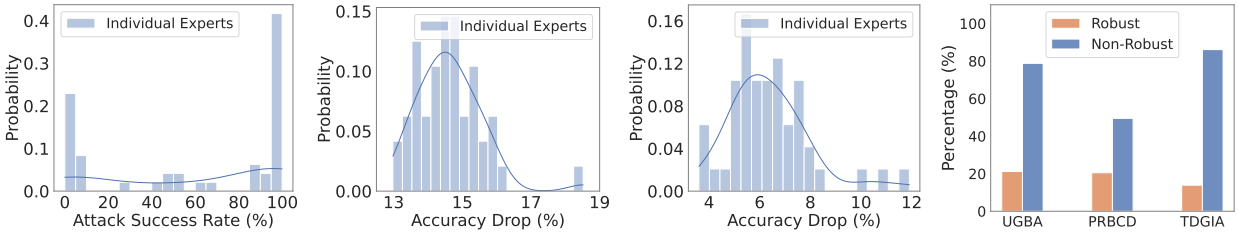
- **Graph Backdoor Attack**: In backdoor attacks, the attacker adds backdoor triggers (i.e., nodes or subgraphs) to a set of selected nodes in the original graph and labels them as target class y_t. The attacked GNN model trained on the poisoned graph should classify target nodes attached with the triggers as target class and behave normally for clean nodes without triggers attached.
- **Manipulation Attack**: In a manipulation attack, an attacker manipulates either the graph structure or node features to decrease the accuracy on the global or locally selected test nodes.
- **Node Injection Attack**: Node injection attacks aim to decrease the global performance by injecting malicious nodes into the training graph.

2.1.2 Mixture of Experts

Sparse mixture of experts Shazeer et al. (2017) decomposes a dense network into multiple experts $\{E_1, E_2, \ldots, E_N\}$. During inference, only a small subset of experts is activated for each input sample. A trainable gating network f_ϕ determines the subset of experts to the activated based on the input sample. Specifically, given the input $\mathbf{x}$, the output of MoE is the weighted sum of the top-K expert predictions:

$$y = \sum_{k \in \mathcal{K}} f_\phi(\mathbf{x})_k E_k(\mathbf{x}). \tag{2}$$

where $\mathcal{K}$ denotes indicates the set of activated top-k expert, and $f_\phi(\mathbf{x})_k$ denotes the gating score for the expert E_k.



(a) Experts under UGBA (b) Experts under PRBCD (c) Experts under TDGIA (d) Robust routing rate

Figure 2: (a)-(c) Robustness of individual experts in a vanilla Graph MoE under backdoor, edge manipulation, and node injection attack; (d) Ability of vanilla router to route perturbed nodes to unaffected experts.

2.2 Robustness Analysis of Graph MoE

In this subsection, we conduct preliminary experiments by training a vanilla graph MoE model on attacked graphs to verify the feasibility of our proposed MoE-based defense. The objective of this preliminary analysis is to examine (i) the robustness of individual experts in MoE and the potential existence of unaffected experts; (ii) the routing behavior and its ability to route perturbed samples to those unaffected experts.

Robustness of Individual Experts. To observe the robustness of individual experts, we train a Graph-MoE Wang et al. (2023) consisting of a router and 48 GNN experts on an OGB-Arxiv graph poisoned by various adversarial perturbations, including the backdoor **UGBA** Dai et al. (2023b), manipulation **PRBCD** Geisler et al. (2021), and node injection **TDGIA** Zou et al. (2021). Fig. 2a shows the performance of individual experts under backdoor attacks and edge manipulation attacks. From Fig. 2a, we observe that approximately 30% of experts achieve an ASR below 20%, which can be regarded as robust experts, under the UGBA attack. Similarly, a small subset of experts exhibits notable robustness under PRBCD and TDGIA, as shown in Fig. 2b and Fig. 2c, respectively. These observations confirm the existence of unaffected experts in Graph MoE across diverse adversarial attacks. On the other hand, the number of robust experts remains limited, highlighting the necessity of developing more effective strategies to diversify GNN experts and ensure a sufficient pool of robust experts.

Analysis of Routing. To observe the router's ability to route perturbed samples to these unaffected experts, we compute and visualize the proportion of perturbed nodes assigned to robust and non-robust experts, respectively. Here, robust experts are defined as those achieving an ASR below 20% for UGBA, an accuracy drop below 10% for PRBCD, and an accuracy drop below 5% for TDGIA. Fig. 2d illustrates the routing rates to robust experts under different attacks. As shown in Fig. 2d, fewer than 20% of perturbed samples are routed to robust experts by the existing router, indicating that the overall robustness of the vanilla Graph MoE is fragile. This observation underscores the necessity of developing a robustness-aware routing mechanism that directs perturbed nodes to those unaffected experts, thereby enhancing the overall robustness of Graph MoE.

2.3 Problem Formulation

With the preliminaries of mixture of experts in Sec. 2.1 and preliminary experiments in Sec. 2.2, the problem of learning a robust graph mixture of experts against various adversarial perturbations can be formally defined as:

Problem 1. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ that may poisoned by various graph adversarial attacks $\mathcal{A}$ (including backdoor attacks, manipulation attacks, and node injection attacks), the robust Graph MoE aims to learn a set of diverse experts $\mathcal{M} = \{E_i\}_{i=1}^{K}$, where there exists a corresponding subset of experts $\mathcal{S}(a(v)) \subseteq \mathcal{M}$ that remain unaffected for the sample v perturbed by an attack $a \in \mathcal{A}$, and a robustness-aware router f_ϕ which can route the perturbed sample $a(v)$ to the corresponding robust experts $\mathcal{S}(a(v))$.*

3 Methodology

Our preliminary analysis has verified the feasibility of the proposed MoE-based defense by demonstrating the existence of unaffected experts and the potential to route perturbed nodes to them. Additionally, the preliminary experiments in Fig. 2 indicate two challenges that remain to be addressed:(i) A vanilla Graph MoE is limited in producing sufficiently diverse experts. How can we further diversify the decision logics of the MoE to ensure the existence of sufficient unaffected experts for each conducted adversarial attack? (ii) how can we accurately route perturbed samples to the corresponding unaffected experts that remain robust to the perturbation?

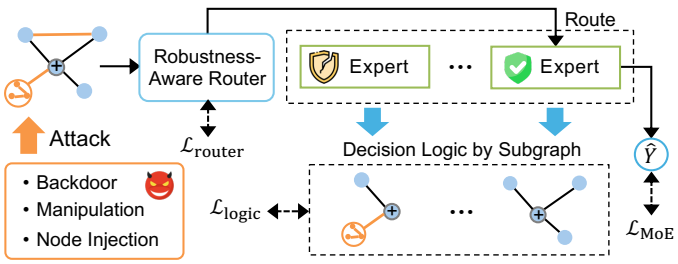


Figure 3: Overall framework of RGMoE.

To address these challenges, we propose RGMoE, a unified MoE-based defense framework against diverse adversarial attacks. As shown in Fig. 3, RGMoE comprises a diverse set of experts. To capture the decision logic of each expert, we measure the contribution of neighbor nodes to the aggregated representation of the center node using the mutual information estimator, followed by a logic diversity loss that encourages different experts to focus on distinct neighbor nodes. This ensures the existence of an unaffected subset of robust experts, each specializing in resistance to specific perturbation patterns. Additionally, RGMoE trains a robustness-aware router to detect the perturbation patterns within samples. Based on these identified patterns, the robustness-aware router adaptively routes each perturbed sample to the corresponding expert that is unaffected by that perturbation. The details of RGMoE are given in the following.

3.1 Mixture of Experts Architecture

As illustrated in Fig. 3, RGMoE contains multiple GNN experts $\{E_1, \ldots, E_n\}$ that have distinct decision logics. A robustness-aware router f_ϕ is designed to identify the perturbed patterns and activate only the robust experts for each perturbed sample. In this section, we present the adopted MoE architecture in RGMoE that enables these objectives.

Following Wang et al. (2023), RGMoE deploys a MoE-based architecture comprising multiple GNN experts. To determine which experts to use for a given node v, a gating network f_ϕ is employed to sparsely route the top-k experts that are denoted as $\mathcal{K}$. Formally, the representation update within the MoE layer can be written as:

$$\mathbf{h}(v) = \sum_{k \in \mathcal{K}} f_\phi(v) \mathrm{E}_k(v, \mathcal{N}_v), \tag{3}$$

where each expert E_k denotes a GNN expert (e.g., GCN Kipf (2016)) that aggregates information from the neighborhood $\mathcal{N}_v$ via the message-passing mechanism. Since the robustness-aware router needs to detect the local structural patterns perturbed by the adversarial attacks, RGMoE adopts a GNN-based gating network as the router. The router leverages both node features and neighborhood information to determine the routing decision:

$$f_\phi(v) = \mathrm{Softmax}(\mathrm{TopK}(Q(v, \mathcal{N}_v))), \tag{4}$$

where Q denotes a separate GNN model.

3.2 Diversifying Expert Decision Logics

In this subsection, we describe how we diversify the decision logics of GNN experts to ensure a sufficient number of robust experts. Adversarial attacks typically mislead the GNN predictions by perturbing the neighborhood $\mathcal{N}_v$ of a target node v. To mitigate this vulnerability, we encourage different GNN experts to attend to distinct neighbors, thereby enabling a subset of unaffected experts to ignore the perturbed patterns. Specifically, we first capture each expert's decision logic during message passing by measuring the mutual information between the aggregated representation of v and its neighbors. We then introduce a logic

diversity loss to encourage different experts to rely on distinct local neighborhoods when making decisions. Details are presented below.

3.2.1 Capturing Logic by Mutual Information.

To capture an expert's decision logic during message passing, we measure each neighbor's contribution to the aggregated representation of node v by mutual information Hjelm et al. (2018). Specifically, for an expert E_k in a GraphMoE layer, let $\tilde{\mathbf{h}}_v^k$ denote the output representation of node v, and $\mathbf{h}_u^k$ denote the input representation of its neighbor $u \in \mathcal{N}_v$ used by expert E_k, we measure the contribution of neighbor u to node v as

$$I^k(u, v) = \mathcal{I}(\mathbf{h}_u^k; \tilde{\mathbf{h}}_v^k), \tag{5}$$

where $\mathcal{I}(\cdot;\cdot)$ denotes mutual information between two variables. By combining the mutual information contributions of all neighbors, we obtain a decision logic vector that characterizes the message-passing pattern learned by expert E_k:

$$\mathbf{I}_v^k = [I^k(u_1, v), I^k(u_2, v), \ldots, I^k(u_{|\mathcal{N}(v)|}, v)], \quad u_i \in \mathcal{N}(v). \tag{6}$$

However, exact mutual information is computationally intractable. We therefore adopt the Jensen-Shannon divergence-based MI estimator Nowozin et al. (2016) $\hat{\mathcal{I}}_\omega^{\mathrm{JSD}}(\mathbf{h}_u^k; \tilde{\mathbf{h}}_v^k)$ to approximate the MI values in Eq. 6:

$$\hat{\mathcal{I}}_\omega^{\mathrm{JSD}}(u; v) := \mathbb{E}_{\mathbb{P}_{u,v}}[-\mathrm{sp}(-T_\omega(\mathbf{h}_u^k, \tilde{\mathbf{h}}_v^k))] - \mathbb{E}_{\mathbb{P}_u \times \mathbb{P}_v}[\mathrm{sp}(T_\omega(\mathbf{h}_u^k, \tilde{\mathbf{h}}_v^k))], \tag{7}$$

where $\mathbb{P}_{u,v}$ denotes the joint distribution of neighbor representations and corresponding center-node outputs, and $\mathbb{P}_u \times \mathbb{P}_v$ denotes the product of their marginal distributions. Here, sp denotes the softplus function. Following Nowozin et al. (2016), we train the discriminator network by optimizing the following loss:

$$\min_\omega \mathcal{L}_{\mathrm{MI}} = -\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \hat{\mathcal{I}}_\omega^{\mathrm{JSD}}(u; v). \tag{8}$$

3.2.2 Logic Diversity Loss.

With the expert decision logic vector $\mathbf{I}_v^k$ in Eq. 6, we further diversify expert behaviors by minimizing the similarity between their decision logics. Specifically, we introduce a logic diversity loss that penalizes high cosine similarity between the decision logics of the selected K experts:

$$\mathcal{L}_{\mathrm{logic}} = \frac{2}{|\mathcal{V}|K(K-1)} \sum_{v \in \mathcal{V}} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \max\left(0,\ s(\mathbf{I}_v^i, \mathbf{I}_v^j) - m\right), \tag{9}$$

where $s(\cdot,\cdot)$ denotes the cosine similarity between two logic vectors, and m is a margin hyperparameter controlling the degree of diversity among experts. Minimizing $\mathcal{L}_{\mathrm{logic}}$ encourages each expert to capture distinct local neighborhood structures, thereby ensuring enough unaffetced experts under each adversarial perturbation.

3.2.3 Diverse Experts Training

With the above MoE architecture, MI estimator in Eq. 8 and the logic diversity loss in Eq. 9, we jointly train the router and experts end-to-end by optimizing:

$$\min_{\phi,\theta,\omega} \mathcal{L}_E = \mathcal{L}_{MoE} + \mathcal{L}_{MI} + \lambda \mathcal{L}_{\mathrm{logic}}, \tag{10}$$

where $\mathcal{L}_{\mathrm{MoE}}$ includes the classification loss and the auxiliary load-balancing loss detailed in Appendix C, and λ is a hyperparameter controlling the strength of the logic diversity regularization.

3.3 Robustness-Aware Router

While the logic diversity loss ensures RGMoE to maintain a sufficient number of robust experts unaffected by certain adversarial perturbations, a robustness-aware router is still necessary to accurately route perturbed nodes to the corresponding robust experts. Intuitively, perturbed nodes tend to induce larger disagreement, since robust experts rely on features that are unaffected by perturbations, whereas non-robust experts do not. Motivated by this, we treat nodes with large expert disagreement as potential perturbed samples. For these potentially perturbed nodes, we fine-tune the router to reduce the routing of highly confident but 'manipulated' experts. This strategy suppresses the participation of non-robust experts while increasing the contribution of robust experts that exhibit more consistent predictive distributions. We present the details of the robustness-aware router training below. We present the details of our robustness-aware router below.

3.3.1 Identification of Potential Perturbed Samples.

We identify potential perturbed nodes by measuring disagreement among experts in the final GraphMoE prediction layer. Specifically, for a GraphMoE with N experts, let $\mathbf{y}_v^n \in \mathbb{R}^C$ represent the corresponding predictive distribution. The disagreement among experts for node v is quantified as the prediction variance:

$$s(v) = \sum_{n=1}^{N} \mathrm{KL}(\mathbf{y}_v^n \parallel \tilde{\mathbf{y}}_v), \quad \tilde{\mathbf{y}}_v = \frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_v^k, \tag{11}$$

where $\tilde{\mathbf{y}}_v$ is the aggregated prediction probability y_v^k of the expert E_k, and $\mathrm{KL}(\cdot \parallel \cdot)$ denotes the Kullback-Leibler divergence. We empirically observe that perturbed nodes $\mathcal{V}_P$ generally exhibit significantly larger expert disagreement than clean nodes $\mathcal{V}_C$. Therefore, we define the set of identified potential perturbed nodes $\hat{\mathcal{V}}_P$ as those whose prediction variance exceeds a threshold derived from the distribution of $s(v)$ over all training nodes $\mathcal{V}_L \cup \mathcal{V}_U$.

$$\hat{\mathcal{V}}_P = \{v \in \mathcal{V}_L \cup \mathcal{V}_U \mid s(v) > \mu_s + \sigma_s\}, \tag{12}$$

where μ_s and σ_s denote the mean and standard deviation of $s(i)$ over all nodes.

3.3.2 Encouraging to Route to Robust Experts.

Given the identified perturbed nodes $\hat{\mathcal{V}}_P$, we further design an objective to route them toward robust experts. For these nodes, which exhibit large prediction disagreement across experts, we reduce the confidence in their original predictions. This discourages the participation of highly confident but manipulated experts and encourages routing toward experts with alternative predictions, i.e., the unaffected experts. To this end, we adopt a soft label $\tilde{\mathbf{y}}_v$ that aggregates predictions from all experts in the final prediction layer:

$$\tilde{\mathbf{y}}_v = \frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_v^k, \tag{13}$$

this soft label reduces the router's reliance on experts with confident but potentially manipulated predictions, thus encouraging a diversified routing strategy that emphasizes consistent and robust experts. Moreover, to further discourage reliance on the original prediction and increase the likelihood of routing to robust experts, we optionally refine the soft label $\tilde{\mathbf{y}}_v$ by applying a shrinkage factor ρ_c (where $0 < \rho_c \leq 1$) to the originally predicted class and then renormalizing:

$$\tilde{\mathbf{y}}_v(c) = \frac{\rho_c \tilde{\mathbf{y}}_v(c)}{\sum_{j=1}^{C} \rho_j \tilde{\mathbf{y}}_v(j)}, \quad c \in \{1, \ldots, C\}. \tag{14}$$

here, ρ_c controls the degree of confidence reduction toward the original predicted class, thereby promoting diversified routing to robust experts. With the constructed soft label $\tilde{\mathbf{y}}_v$ for perturbed nodes $\hat{\mathcal{V}}_P$ and hard label $\mathbf{y}_v$ for clean nodes $\mathcal{V}_C = \mathcal{V}_L \setminus \hat{\mathcal{V}}_P$, we train the robustness-aware router by minimizing:

$$\min_{\phi} \mathcal{L}_{\text{router}} = \frac{1}{|\hat{\mathcal{V}}_P|} \sum_{v \in \hat{\mathcal{V}}_P} l(\hat{\mathbf{y}}_v, \tilde{\mathbf{y}}_v) + \gamma \frac{1}{|\mathcal{V}_C|} \sum_{v \in \mathcal{V}_C} l(\hat{\mathbf{y}}_v, \mathbf{y}_v), \tag{15}$$

where $\hat{\mathbf{y}}_v$ denotes the aggregated predictions from the final MoE layer, l denotes the cross-entropy loss, and γ balances the routing supervision from clean and perturbed nodes.

3.4 Final Objective Function

Combining the diversity loss in Eq.(9) and the training of robustness-aware router by Eq.(15), the final objective function can be formulated as:

$$\min_{\phi,\theta,\omega} \mathcal{L}_{\text{total}} = \mathcal{L}_E + \mathcal{L}_{\text{router}}. \tag{16}$$

The training procedure of RGMoE is divided into two phases: we first train the graph MoE model with diverse experts using $\mathcal{L}_E$ to obtain robust experts; then we fine-tune the router with $\mathcal{L}_{\text{router}}$ to route the perturbed nodes to these robust experts. We present the detailed training procedure of method in Algorithm 1.

4 Experiments

In this section, we evaluate RGMoE on large-scale graph datasets under various attacks to answer the following questions:

- **RQ1:** How effective is RGMoE in defending against various GNN adversarial attacks?
- **RQ2:** How does scaling the model capacity of RGMoE impact the robustness and accuracy?
- **RQ3:** How do the MI-based diversity loss and robustness-aware router contribute the robustness of RGMoE?

Table 1: Robustness of RGMoE under backdoor attacks. We report the ASR on attacked nodes and ACC on clean nodes.

Attacks	Defense	Cora		Pubmed		Flickr		OGB-arxiv	
		ASR (%) ↓	ACC(%) ↑	ASR (%) ↓	ACC (%) ↑	ASR (%) ↓	ACC (%) ↑	ASR (%) ↓	ACC (%) ↑
GTA	GCN	100.00 ± 0.00	89.14 ± 0.46	99.76 ± 0.03	84.29 ± 0.33	100.00 ± 0.00	38.21 ± 0.35	95.80 ± 0.33	57.45 ± 0.48
	GNNGuard	98.81 ± 0.37	79.26 ± 0.60	59.97 ± 2.32	80.65 ± 0.13	58.71 ± 38.58	40.69 ± 0.25	40.51 ± 4.80	60.07 ± 1.76
	SoftMedian	23.99 ± 1.47	89.63 ± 0.30	22.57 ± 0.20	82.78 ± 0.12	22.76 ± 0.02	40.32 ± 0.00	40.96 ± 0.66	60.63 ± 0.68
	Prune	4.74 ± 0.21	88.77 ± 1.06	5.98 ± 2.08	84.39 ± 0.09	0.98 ± 0.02	41.10 ± 0.19	8.57 ± 0.43	61.06 ± 0.46
	OD	32.89 ± 26.51	66.54 ± 1.67	19.27 ± 17.15	84.71 ± 0.32	2.67 ± 3.77	40.45 ± 0.12	11.34 ± 4.77	**61.26 ± 0.21**
	RIGBD	2.07 ± 2.93	87.65 ± 1.49	2.36 ± 2.76	83.68 ± 0.43	0.00 ± 0.00	41.23 ± 0.14	49.75 ± 8.39	59.35 ± 0.70
	DP-MoE	71.47 ± 21.03	86.54 ± 3.66	76.89 ± 15.08	85.80 ± 3.84	70.20 ± 36.88	34.45 ± 2.29	30.68 ± 24.64	50.34 ± 5.39
	Ours	**0.74 ± 0.76**	**89.68 ± 3.50**	**1.52 ± 0.94**	**87.47 ± 0.22**	**0.00 ± 0.00**	**42.11 ± 2.1**	**3.24 ± 1.42**	57.50 ± 1.36
UGBA	GCN	98.67 ± 0.00	89.90 ± 0.30	92.88 ± 1.53	80.25 ± 0.24	99.96 ± 0.01	39.53 ± 0.21	99.60 ± 0.03	62.47 ± 0.43
	GNNGuard	99.41 ± 0.21	80.74 ± 0.00	98.67 ± 1.61	82.02 ± 0.37	98.89 ± 0.12	43.05 ± 0.26	98.41 ± 0.36	58.30 ± 4.25
	SoftMedian	29.63 ± 2.55	80.37 ± 0.52	13.88 ± 0.11	82.65 ± 0.21	15.95 ± 1.92	40.32 ± 0.00	53.06 ± 1.77	61.49 ± 0.29
	Prune	97.04 ± 1.17	87.78 ± 0.60	93.44 ± 1.51	82.85 ± 0.15	99.85 ± 0.03	40.48 ± 0.01	99.14 ± 0.24	61.58 ± 0.65
	OD	36.00 ± 34.17	87.53 ± 0.30	33.01 ± 6.70	83.22 ± 0.37	2.29 ± 3.07	41.17 ± 0.02	64.12 ± 36.55	60.49 ± 0.47
	RIGBD	5.67 ± 1.43	87.90 ± 1.06	5.81 ± 3.71	**83.83 ± 0.52**	0.00 ± 0.00	41.33 ± 0.11	0.11 ± 0.08	60.68 ± 0.55
	DP-MoE	84.74 ± 4.39	80.99 ± 1.97	93.63 ± 2.36	81.21 ± 2.36	99.02 ± 0.18	39.42 ± 1.12	83.74 ± 2.12	59.14 ± 6.95
	Ours	**1.48 ± 1.51**	**89.75 ± 1.06**	**1.57 ± 1.70**	82.61 ± 1.70	**0.00 ± 0.00**	**45.71 ± 0.53**	**0.03 ± 0.03**	**63.38 ± 0.50**
DPGBA	GCN	89.63 ± 2.48	88.02 ± 0.17	90.65 ± 1.71	83.26 ± 0.13	99.98 ± 0.01	38.60 ± 0.04	95.92 ± 0.19	60.78 ± 1.05
	GNNGuard	37.48 ± 2.00	79.59 ± 0.80	49.16 ± 20.32	83.49 ± 0.21	84.61 ± 6.81	43.03 ± 0.91	96.39 ± 0.17	60.26 ± 1.46
	SoftMedian	31.70 ± 0.55	80.37 ± 0.30	21.66 ± 2.08	82.82 ± 0.15	99.06 ± 0.09	40.32 ± 0.00	39.59 ± 0.51	61.88 ± 0.44
	Prune	93.77 ± 1.43	85.49 ± 0.46	53.84 ± 0.65	82.04 ± 0.27	65.01 ± 45.24	37.36 ± 4.14	21.31 ± 0.42	61.49 ± 0.56
	OD	94.96 ± 1.34	81.99 ± 0.46	88.65 ± 1.95	83.27 ± 0.15	10.94 ± 15.48	40.36 ± 0.02	98.33 ± 0.17	58.59 ± 0.45
	RIGBD	0.47 ± 0.33	86.42 ± 0.46	1.81 ± 0.19	83.22 ± 0.48	0.00 ± 0.00	40.96 ± 0.38	0.74 ± 0.79	61.14 ± 0.55
	DP-MoE	89.78 ± 0.63	88.15 ± 1.51	91.59 ± 0.29	79.91 ± 0.38	99.26 ± 0.10	40.77 ± 0.48	99.75 ± 0.11	61.60 ± 0.49
	Ours	**0.30 ± 0.42**	**88.27 ± 4.06**	**0.62 ± 0.33**	**83.75 ± 0.85**	**0.00 ± 0.00**	**45.14 ± 2.55**	**0.02 ± 0.02**	**62.11 ± 0.48**

4.1 Experimental Settings

4.1.1 Datasets

To validate the effectiveness of our proposed methods, we conduct experiments on four real-world benchmark datasets widely used for node classification, i.e., Cora *(small-scale)*, Pubmed *(medium-scale)* Sen et al. (2008), Flickr *(medium-scale)* Zeng et al. (2019) and OGB-Arxiv *(large-scale)* Hu et al. (2020). The statistics of the datasets are in Appendix F.

4.1.2 Attack Methods

To demonstrate the robustness of our RGMoE against graph backdoor, manipulation, and node injection attacks, we employ the following representative attack methods for each type of graph adversarial attack:

- **Graph Backdoor Attack**: we evaluate our method on three state-of-the-art graph backdoor attack methods, i.e., **GTA** Zhang et al. (2021), **UGBA** Dai et al. (2023b), and **DPGBA** Zhang et al. (2024b). **GTA** is the first backdoor attack on GNNs that defines the graph-oriented triggers as specific subgraphs. The backdoored GNN should predict the target label for any test samples attached with the trigger and behave normally otherwise. **UGBA** selects representative nodes as poisoned nodes and optimizes the adaptive trigger generator with similarity-constrained loss to conduct unnoticeable graph backdoor attacks with a limited attack budget. **DPGBA** further introduces adversarial learning to generate in-distribution triggers.

- **Manipulation Attack**: We evaluate **PRBCD** Geisler et al. (2021), a representative edge manipulation attack on large graphs. Its attack does not scale quadratically with the number of nodes.

- **Node Injection Attack**: We evaluate **TDGIA** Chen et al. (2021), a typical graph injection attack that injects a small amount of labeled fake nodes progressively around topologically vulnerable nodes in the graph. The authors introduced homophily unnoticeability to promote unnoticeability while maintaining the damage.

4.1.3 Baseline Methods

To evaluate the performance of RGMoE, we compare against three categories of baseline approaches.

- **Backdoor Defense Methods.** We compare with state-of-the-art backdoor defense methods: **Prune** Dai et al. (2023b) detects and removes triggers by identifying edges with low cosine similarity; **OD** Zhang et al. (2024a) uses a graph auto-encoder to filter out-of-distribution trigger nodes based on reconstruction loss; **RIGBD** Zhang et al. (2024b) uses random edge dropping to detect and purify backdoored nodes with large prediction variance.

- **Robust GNN Methods.** We also compare with SOTA and representative robust GNN approaches designed to defend against adversarial attacks: **RGCN** Zhu et al. (2019) employs Gaussian distributions with variance-based attention to absorb adversarial effects; **GNNGuard** Zhang & Zitnik (2020) deletes adversarial edges based on node similarity and adjusts edge weights during training; **SoftMedian** Geisler et al. (2021) uses weighted mean aggregation based on distance to dimension-wise median; **SimPGCN** Jin et al. (2021a) adaptively integrates graph structure and node features through similarity-preserving aggregation.

- **MoE-based Defense.** Finally, we also include **DP-MoE** Yuan et al. (2024), which combines differential privacy and MoE to apply adaptive noises to defend against node injection attacks.

4.1.4 Evaluation Metrics

Here, we present the details of how we evaluate the robustness against various graph adversarial attacks.

- **Graph Backdoor Attacks**. Following the existing representative graph backdoor studies Dai et al. (2023b); Zhang et al. (2024b), we conduct experiments on the inductive node classification task, where 20% of the nodes are masked out from the original graph as test nodes. The attacker can inject 5% nodes out of the original graph as poisoned nodes on Cora, Pubmed, Flickr, and 3% on Arxiv. The target class is set to 0, and the trigger size is limited to 3 on all datasets. During defense, we attach half of the test nodes (10%) with backdoor triggers to test the attack success rate **(ASR)**. The remaining 10% are kept clean to test the clean accuracy **(ACC)**.

Table 2: Robustness of RGMoE under manipulation attack (P-RBCD). We report the ACC on attacked nodes.

Dataset	Ptb Rate (%)	GCN	RGCN	GNNGuard	SoftMedian	SimPGCN	DP-MoE	Ours
Flickr	3	42.27 ± 0.17	47.32 ± 0.24	43.96 ± 1.61	43.31 ± 0.02	42.80 ± 0.13	43.96 ± 0.02	**51.04 ± 0.21**
	5	41.32 ± 0.74	46.84 ± 0.36	42.44 ± 1.74	42.71 ± 0.01	42.80 ± 0.14	43.71 ± 0.01	**50.98 ± 0.18**
	10	38.02 ± 2.53	43.46 ± 0.77	40.43 ± 2.45	42.71 ± 0.00	42.79 ± 0.12	43.23 ± 0.03	**50.16 ± 0.63**
OGB-Arxiv	3	48.79 ± 0.63	50.71 ± 0.15	49.06 ± 0.61	50.46 ± 0.36	50.17 ± 0.22	49.49 ± 0.65	**53.55 ± 0.02**
	5	46.13 ± 0.55	48.10 ± 0.12	46.43 ± 0.37	48.10 ± 0.11	47.38 ± 0.21	47.27 ± 0.69	**50.71 ± 0.09**
	10	34.52 ± 0.44	36.59 ± 0.16	34.50 ± 0.60	37.64 ± 0.20	35.34 ± 0.21	35.72 ± 0.60	**39.94 ± 0.04**

Table 3: Robustness of RGMoE node injection attack (TDGIA). We report the ACC on test nodes.

Dataset	Inj.Node/Edge	GCN	RGCN	GNNGuard	SoftMedian	SimPGCN	DP-MoE	Ours
Flickr	3000, 5	43.84 ± 0.07	49.18 ± 0.18	45.24 ± 2.35	44.38 ± 0.04	43.62 ± 0.23	44.59 ± 0.49	**50.63 ± 0.16**
	5000, 10	43.20 ± 0.06	49.16 ± 0.07	44.91 ± 1.52	42.71 ± 0.00	42.82 ± 0.13	42.67 ± 0.04	**49.62 ± 0.24**
	8000, 20	39.17 ± 0.24	46.88 ± 0.27	41.25 ± 2.67	40.56 ± 0.07	40.46 ± 0.21	39.64 ± 0.78	**47.08 ± 0.73**
OGB-Arxiv	100, 50	58.79 ± 0.19	59.68 ± 0.09	60.32 ± 0.14	62.87 ± 0.50	60.53 ± 4.36	61.76 ± 0.37	**63.33 ± 0.46**
	1500, 60	55.71 ± 0.90	58.54 ± 0.45	59.21 ± 0.56	61.18 ± 0.35	59.12 ± 0.35	59.86 ± 0.31	**61.19 ± 0.53**
	2000, 80	52.49 ± 0.86	56.42 ± 0.26	58.72 ± 0.66	59.03 ± 0.56	57.06 ± 0.71	56.95 ± 1.17	**59.14 ± 0.37**

- **Manipulation and Node Injection Attacks**. Since PRBCD and TDGIA both aim to decrease the overall performance, we follow the previous study Chen et al. (2022) to report the accuracy **(ACC)**. 20% nodes are masked and selected as test nodes. Moreover, we vary different attack budgets (i.e., edge perturbation rate for PRBCD, number of injected nodes or edges for TDGIA) to evaluate the consistent efficacy of RGMoE.

4.1.5 Implementation Details

We deploy a 2-layer GCN as the model architecture across all experiments. Each experiment is conducted 3 times and the average results are reported. More implementation details are specified in Appendix G. We use an Adam optimizer with a fixed learning rate of 0.01 for all baseline methods. All experiments are conducted on a single A6000 GPU.

4.2 Robustness of RGMoE (RQ1)

To answer **RQ1**, we compare RGMoE with baseline defense methods across the datasets under various graph adversarial attacks.

Defend Against Graph Backdoor Attacks. To investigate the robustness of RGMoE against backdoor attacks, we evaluate its performance across the datasets under three state-of-the-art attack methods, i.e., GTA, UGBA and DPGBA Following the above-mentioned evaluation protocol, we report the ASR and ACC in Table 1. From Table 1, we make two key observations: (**i**) Across all baselines, few of them can give effective defenses across all attacks. The existing MoE-based architecture, DP-MoE gives poor performance on backdoor attacks with ASR>80% on many datasets. In contrast, our RGMoE consistently achieves the lowest ASR, often approaching 0% across the datasets and backdoor attack methods. This demonstrates the effectiveness of our RGMoE in defending against various backdoor attacks. (**ii**) Our method often attains slightly higher clean accuracy compared to vanilla GCN and most defense baselines. This is because that logic diversity loss encourages different experts to learn complementary local structures. As a result, the cooperation of the diverse experts in Graph MoE facilitates both robustness and accuracy on clean nodes.

Defend Against Manipulation and Node Injection Attacks. We then evaluate RGMoE on large-scale datasets under PRBCD *(edge manipulation)* and TDGIA *(node injection)*. These two attack methods are selected due to their representativeness and scalability to large-scale datasets. The results of defending against manipulation and node injection attacks on large-scale datasets are presented in Table 2 and Table 3, respectively. Additional results on small-scale datasets are presented in Appendix I. Different attack budgets are applied to evaluate the consistent efficacy of RGMoE. From the tables, we observe: (**i**) In Table 2, RGMoE consistently achieves higher accuracy than all baselines across perturbation rates from 3% to 10%,

(**ii**) Similarly in Table 3, our method not only surpasses all baselines under different numbers of injected nodes and edges. These results demonstrate the efficacy of our RGMoE in producing robust experts and routing perturbed nodes under various perturbations, including backdoors, manipulation, and node injection.

4.3 Impact of MoE Capacity to Robustness (RQ2)

To answer **RQ2**, we scale the capacity of RGMoE by varying the number of experts $N \in \{12, 18, 24, 30, 36, 42, 48\}$, while consistently routing the top $N/6$ experts. This setup allows us to investigate whether increasing MoE capacity leads to improvements in robustness or accuracy. For fair comparison, other parameters (i.e., λ_{div} and γ) are tuned till best. Results across datasets are summarized in Figure 4. More results under other attacks can be found in Appendix I. From the figure, we observe that: (i) Scaling the model capacity greatly enhances robustness. For example, with only $N = 12$ experts, the ASR under DPGBA exceeds 25% on Cora, Pubmed, and Arxiv, with large variance across random seeds. As the number of experts increases, the ASR on all three datasets converges to nearly 0 when $N = 48$. This indicates that a larger pool of experts can result in more experts unaffected by backdoor triggers. Consequently, the robustness-aware router can reliably route poisoned nodes to these unaffected experts. (ii) Dataset scale effect at fixed capacity. At the *same* number of experts, Arxiv exhibits lower ASR than the smaller datasets (Cora and Pubmed). This is because, on larger datasets, MoE can exploit richer data to learn more diverse features that are less aligned with adversarial patterns, yielding stronger robustness. In contrast, the clean accuracy remains stable or slightly improves with scaling, this implies the diversity of experts wouldn't harm the overall accuracy of the MoE framework.

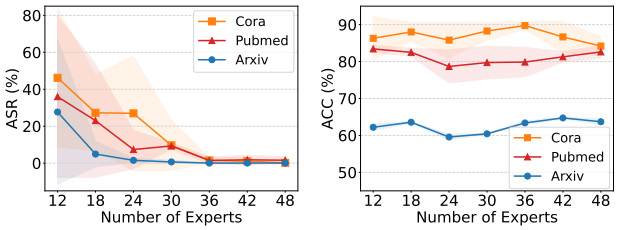


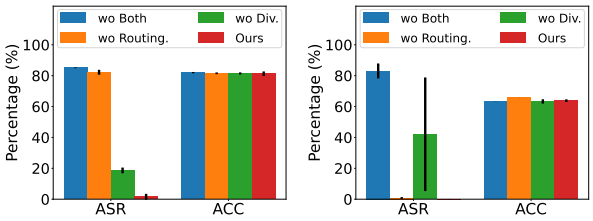
Figure 4: Impact of model capacity under UGBA attack.

4.4 Ablation Study and Hyperparameter Analysis (RQ3)

Ablation Study. To answer **RQ3**, we conduct ablation studies to examine the contributions of the proposed logic diversity loss and the robustness-aware routing strategy. Specifically, we (i) set $\lambda_{\text{div}} = 0$ to remove the diversity loss, and (ii) replace the robustness-aware routing with random top-k routing, and (iii) removing both the diversity loss and robustness-aware router. Figure 5 reports results on Pubmed and Arxiv under the UGBA attack. Please refer to Appendix I for other datasets and attacks. From the figure, we observe:

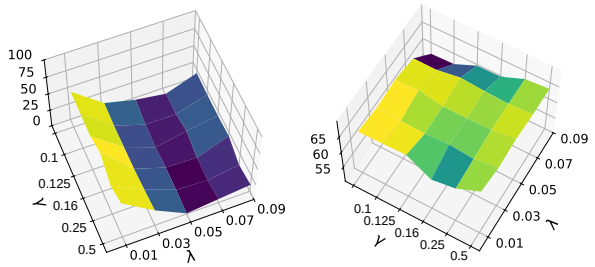
- With $\lambda_{\text{div}} = 0$, the ASR drops considerably (e.g., from above 80% to about 20% on Pubmed and 40% on Arxiv), showing that the logic diversity loss is essential in ensuring a number of experts are robust to various attacks, thus enabling the router to select robust experts without scarifying the accuracy.
- When removing the robustness-aware router, ASR increases significantly on small-scale datasets (Cora and Pubmed), indicating that the router effectively redirects perturbed nodes to robust experts and thus enhances overall robustness. On large-scale datasets (Flickr and Arxiv), the ASR shows a relatively smaller increase, which implies the existence of more robust experts.
- When combining the robustness-aware router and logic diversity loss, ASR further decreases to nearly 0% on both datasets. This highlights that the diversity loss and the robustness-aware router are complementary: the loss encourages experts to learn distinct, less-correlated features, while the router ensures that backdoored nodes are directed to the most robust experts.

Hyperparameter Analysis. We further study the effect of the two key hyperparameters: λ_{div} in Eq. (9), which controls the strength of the MI-based logic diversity loss, and γ in Eq. (15), which controls the degree of training router with soft labels for robustness-aware routing. To explore their effects, we vary the values of λ as $\{0.01, 0.03, 0.05, 0.07, 0.09\}$; and γ as $\{0.1, 0.125, 0.16, 0.25, 0.5\}$. Figure 6 reports results on Arxiv under the DPGBA attack. We observe that once $\lambda_{\text{div}} > 0.03$, RGMoE consistently achieves robust performance



(a) Pubmed (b) Arxiv

Figure 5: Ablation studies under UGBA across datasets.



(a) ASR (%) (b) ACC (%)

Figure 6: Hyperparameter analysis under DPGBA on Arxiv.

with ASR<20%. Under some parameters, RGMoE can reach near-zero ASR. Meanwhile, increasing λ has little impact on the clean accuracy, implying the logic diversity can enhance the robustness without sacrificing the performance. On the other hand, increasing γ can bring some benefit in robustness, with the observed decreased ASR. Its impact on the clean accuracy is also small.

5 Related Work

Graph Adversarial Attacks and Defense. Extensive studies have revealed that GNNs are vulnerable to adversarial attacks via edge manipulation Zügner & Günnemann (2019); Geisler et al. (2021), node feature perturbation Sun et al. (2020); Zügner et al. (2018), and node injection Zou et al. (2021); Chen et al. (2022). For example, PRBCD Geisler et al. (2021) can fool GNNs to give false predictions on target nodes by adding or deleting an unnoticeable amount of edges of the training graph. TDGIA Zou et al. (2021) manages to significantly reduce the global node classification performance of GNNs by injecting a small amount of fake nodes. Emerging studies Xi et al. (2021); Zhang et al. (2024a;b) also highlighted the significance of graph backdoors. UGBA Dai et al. (2023b) controls the model's prediction by injecting specifically designed unnoticeable triggers to selected nodes. DPGBA Zhang et al. (2024a) further proposes to generate in-distribution triggers. To address the susceptibility of GNNs to adversarial attacks, many defense methods have been proposed, including adversarial training Xu et al. (2019), graph structure denoising Jin et al. (2020), robust aggregation Chen et al. (2021); Geisler et al. (2021); Jin et al. (2021a), and backdoor cleaning Dai et al. (2023b); Zhang et al. (2024a;b). However, most existing defenses rely on attack-tailored designs to robustify a single GNN. There lacks a unified framework to defend against various adversarial attacks.

Mixture of Experts. The Sparse Mixture of Experts (MoE) Jacobs et al. (1991); Shazeer et al. (2017); Li et al. (2022) architecture determines a group of experts for each input sample, therefore greatly increasing the model's capacity without losses in efficiency. Despite the success of MoE in scaling up model capacity Shazeer et al. (2017); Fedus et al. (2022); Dai et al. (2024a), existing studies have also investigated the potential of MoE architecture beyond efficiency. In the graph domain, Wu et al. (2024) designed GraphMetro to address complex distribution shifts in graphs. Liu et al. (2023) diversified the representations of graph experts to learn fair representations. Yuan et al. (2024) explores the potential of MoE to defend node injection attacks. Different from previous papers, we firstly utilize MoE to develop a unified framework against various graph adversarial attacks by training a subset of unaffected experts and routing perturbed nodes to them.

6 Conclusion and Future Work

In this paper, we identify two key properties of the MoE: the *diversity* in capturing representations and the *flexibility* in routing samples to certain experts. Building on this, we introduce two novel designs (i.e., logic diversity loss and robustness-aware router) to propose RGMoE, a unified robust framework by training unaffected experts and routing perturbed nodes to them. RGMoE can defend against various graph adversarial attacks with the mildest inference-time costs. In future work, we will explore the broader potential of MoE and build GNNs that are both powerful and trustworthy.

References

Liang Chen, Jintang Li, Qibiao Peng, Yang Liu, Zibin Zheng, and Carl Yang. Understanding structural vulnerability in graph convolutional networks. 2021.

Yongqiang Chen, Han Yang, Yonggang Zhang, MA KAILI, Tongliang Liu, Bo Han, and James Cheng. Understanding and improving graph injection attack by promoting unnoticeability. In *International Conference on Learning Representations*, 2022.

Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *ACL (1)*, 2024a.

Enyan Dai, Charu Aggarwal, and Suhang Wang. Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 227–236, 2021.

Enyan Dai, Wei Jin, Hui Liu, and Suhang Wang. Towards robust graph neural networks for noisy graphs with sparse labels. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pp. 181–191, 2022.

Enyan Dai, Limeng Cui, Zhengyang Wang, Xianfeng Tang, Yinghan Wang, Monica 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*, pp. 368–379, 2023a.

Enyan Dai, Minhua Lin, Xiang Zhang, and Suhang Wang. Unnoticeable backdoor attacks on graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pp. 2263–2273, 2023b.

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. *Machine Intelligence Research*, 21(6):1011–1061, 2024b.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39, 2022.

Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. *Advances in Neural Information Processing Systems*, 34:7637–7649, 2021.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

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. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

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.

Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 148–156, 2021a.

Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. Adversarial attacks and defenses on graphs. *ACM SIGKDD Explorations Newsletter*, 22(2):19–34, 2021b.

TN Kipf. Semi-supervised classification with graph convolutional networks. *ICLR*, 2016.

Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. *arXiv preprint arXiv:2206.04046*, 2022.

Zheyuan Liu, Chunhui Zhang, Yijun Tian, Erchi Zhang, Chao Huang, Yanfang Ye, and Chuxu Zhang. Fair graph representation learning via diverse mixture-of-experts. In *Proceedings of the ACM Web Conference 2023*, pp. 28–38, 2023.

Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1):20381, 2019.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S Yu, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35 (8):7693–7711, 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.

Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. *Advances in Neural Information Processing Systems*, 36:50825–50837, 2023.

Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples for graph data: deep insights into attack and defense. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4816–4823, 2019.

Shirley Wu, Kaidi Cao, Bruno Ribeiro, James Y Zou, and Jure Leskovec. Graphmetro: Mitigating complex graph distribution shifts via mixture of aligned experts. *Advances in Neural Information Processing Systems*, 37:9358–9387, 2024.

Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *30th USENIX security symposium (USENIX Security 21)*, pp. 1523–1540, 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*, 2019.

Xiangchi Yuan, Chunhui Zhang, Yijun Tian, Yanfang Ye, and Chuxu Zhang. Mitigating emergent robustness degradation while scaling graph learning. The Twelfth International Conference on Learning Representations (ICLR), 2024.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

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

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM symposium on access control models and technologies*, pp. 15–26, 2021.

Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4386–4397, 2024a.

Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness-inspired defense against backdoor attacks on graph neural networks. *arXiv preprint arXiv:2406.09836*, 2024b.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1399–1407, 2019.

Xu Zou, Qinkai Zheng, Yuxiao Dong, Xinyu Guan, Evgeny Kharlamov, Jialiang Lu, and Jie Tang. Tdgia: Effective injection attacks on graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2461–2471, 2021.

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 Appendix

B Case Study (RQ3)

To further analyze the efficacy of logic diversity loss and robustness-aware router in **RQ3**, we conduct a case study by training RGMoE on Arxiv under the UGBA attack to demonstrate (i) the existence of unaffected individual experts in RGMoE; and (ii) the routing choices of perturbed nodes.

Fig. 7 visualizes ASR of 48 experts and the gating weights of 10 random perturbed nodes. Each perturbed node can be routed to 8 experts. From the figure, we observe that the logic diversity loss successfully promotes nearly 50% of experts to be unaffected with ASR closing to 0%. Meanwhile, almost all perturbed samples are successfully routed to robust experts that are unaffected by the perturbation.

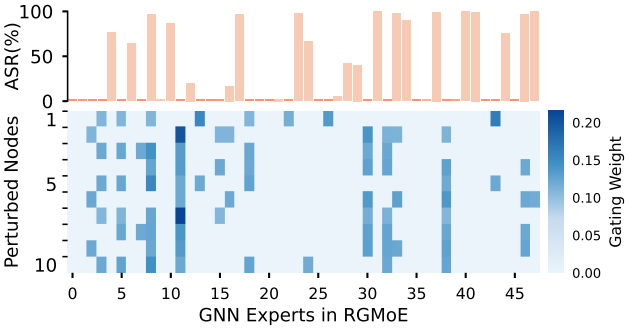


Figure 7: Existence of robust experts and routing choices.

C Auxiliary loss of RGMoE

Following Shazeer et al. (2017), we define the *importance* of each expert as the batch-wise sum of its gate values:

$$I_k = \sum_{v \in \mathcal{V}} f_\phi(v)_k, \quad \bar{I} = \frac{1}{N} \sum_{k=1}^{N} I_k, \tag{17}$$

where $\mathcal{V}$ denotes the batch of nodes and N is the total number of experts. The importance loss $\mathcal{L}_{\text{importance}}$ is computed as the squared coefficient of variation (CV) over all experts:

$$\mathcal{L}_{\text{importance}} = \frac{\frac{1}{N} \sum_{k=1}^{N} (I_k - \bar{I})^2}{\bar{I}^2}. \tag{18}$$

To further ensure a balanced assignment of samples across experts, we adopt a *load-balancing loss* Shazeer et al. (2017):

$$\mathcal{L}_{\text{load}} = \frac{N \cdot \sum_{k=1}^{N} I_k L_k}{\left(\sum_{k=1}^{N} I_k\right)\left(\sum_{k=1}^{N} L_k\right)}, \tag{19}$$

where L_k denotes the expected number of samples routed to expert k. These two auxiliary losses encourage all experts to be both equally important and equally utilized, preventing expert imbalance and training collapse.

D Training Algorithm of RGMoE

The training algorithm of RGMoE is proposed in Algorithm 1. The whole training process is composed of two phases. Specifically, we first train the diverse GNN experts with the initial router from line 1 to line 5 based on Eq. (10). Then we fine-tune to get the robustness-aware router by identifying $\hat{\mathcal{V}}_P$ and constructing the smooth labels. The router is optimized from line 7 to line 9 using Eq. (15).

E Time Complexity Analysis

In this subsection, we analyze the time complexity of RGMoE. Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, let L be the number of GCN layers, h the hidden dimension, d the average node degree and N the number of experts.

Algorithm 1 Algorithm of RGMoE.

Input: Poisoned Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, $\mathcal{Y}_L$, N, K. λ, ω.
Output: MoE model f_θ, robust router f_ϕ and MI estimator T_ω.

```
Randomly initialize f_θ, f_φ and T_ω;
while not converged yet do
    Update f_θ, f_φ and T_ω by descent on ∇_{θ,φ,ω}(L_c + λL_logic) based on Eq. (10);
end while
Identify V̂_P via expert disagreement based on Eq.(12);
Obtain smooth labels ỹ for V̂_P based on Eq.(14) ;
while not converged yet do
    Update f_φ by descent on ∇_φ L_router based on Eq. (15);
end while
return f_θ and f_φ;
```

In the training phase, the cost of the router is approximately $O(Lh|\mathcal{V}|d + |\mathcal{V}|N)$ for calculating the message-passing and gating score. The cost of GNN experts is approximately $O(Lh|\mathcal{V}|dK)$, where K is the number of activated experts and $K < N$. According to Eq. (7), the cost of the MI estimator (i.e., an MLP) is approximately $O(Lh|\mathcal{V}|d)$ by neglecting the number of negative samples, which is a small constant. The cost of calculating the logic diversity loss is approximately $O(L|\mathcal{V}|dK^2)$ according to Eq. (9). When training the robustness-aware router, the process of calculating expert disagreement is $O(|\mathcal{V}|N)$, which is linear to the number of experts, and the process of constructing the smooth label is $O(|\mathcal{V}|N)$. In the test phase, the cost is only $O(Lh|\mathcal{V}|d + |\mathcal{V}|N)$ for the router and $O(Lh|\mathcal{V}|dK)$ for the activated experts. Our time complexity analysis proves that RGMoE has great potential in conducting scalable defense on large-scale graphs.

Table 4: Dataset Statistics

Datasets	Nodes	Edges	Feature	Classes
Cora	2,708	5,429	1,443	7
Pubmed	19,717	44,338	500	3
Flickr	89,250	899,756	500	7
OGB-arxiv	169,343	1,166,243	128	40

Table 5: Additional results in defending against manipulation attack (PRBCD) on Cora.

Dataset	**Ptb (%)**	**GCN**	**RGCN**	**GNNGuard**	**SoftMedian**	**SimPGCN**	**DP-MoE**	**Ours**
Cora	0 (Clean)	85.34 ± 0.46	85.45 ± 0.33	84.33 ± 0.03	83.81 ± 0.87	86.34 ± 0.18	85.88 ± 0.56	86.08 ± 0.26
	3	72.09 ± 0.00	73.81 ± 0.09	76.43 ± 0.83	77.70 ± 0.98	77.20 ± 1.00	73.51 ± 0.41	76.65 ± 0.44
	5	65.50 ± 0.57	68.15 ± 0.17	74.05 ± 0.40	74.55 ± 0.61	70.92 ± 0.09	66.41 ± 2.58	70.06 ± 0.54
	10	55.95 ± 0.23	58.60 ± 0.52	66.07 ± 0.38	66.91 ± 0.80	63.46 ± 0.09	56.58 ± 0.69	64.88 ± 1.06

F Dataset Statistics

Cora and Pubmed are citation networks where nodes denote papers, and edges depict citation relationships. In Cora, each node is described using a binary word vector, indicating the presence or absence of a corresponding word from a predefined dictionary. In contrast, PubMed employs a TF/IDF weighted word vector for each node. Nodes are categorized based on their respective research areas for both datasets.

Flickr represents images uploaded to Flickr as nodes in a graph. An edge is created between two nodes if the corresponding images share certain attributes, such as location, gallery, or user comments. Each image is described by a 500-dimensional bag-of-words feature vector derived from the NUS-wide dataset. Labels are based on 81 tags assigned to the images, which were manually grouped into 7 distinct classes. Each image is categorized into one of these classes.

Table 6: Additional results in defending against node injection attack (TDGIA) on Cora.

Dataset	Inj.Node/Edge	GCN	RGCN	GNNGuard	SoftMedian	SimPGCN	DP-MoE	Ours
Cora	40,3	69.75 ± 0.61	78.74 ± 0.54	80.45 ± 0.09	77.88 ± 0.23	77.02 ± 0.46	78.80 ± 0.53	80.59 ± 0.69

Table 7: Additional results in defending against feature-based attack (Nettack) on Cora.

Dataset	Ptb (%)	GCN	RGCN	GNNGuard	SoftMedian	SimPGCN	DP-MoE	Ours
Cora	15	69.89 ± 1.01	78.56 ± 0.16	76.77 ± 0.89	80.78 ± 0.57	80.35 ± 0.66	73.70 ± 2.01	81.18 ± 1.01

OGB-Arxiv is a citation network consisting of all Computer Science papers from arXiv cataloged in the Microsoft Academic Graph. Each paper is represented by a node, with a 128-dimensional feature vector derived by averaging the skip-gram word embeddings from the paper's title and abstract. The nodes are classified according to their respective research areas. The statistics details of these datasets are summarized in Tab. 4.

G Implementation Details

Train, validation, and test splits. In this paper, we conduct semi-supervised node classification experiments. For backdoor attacks, 20% of nodes are selected as training nodes on Pubmed, Flickr, and OGB-Arxiv, and 70% on Cora due to its smaller size. 10% are used as the validation set, while 20% are masked as test nodes, split equally for ASR (10%) and ACC (10%) evaluation. For manipulation and node injection attacks, 20% of nodes are randomly selected as training nodes across all datasets, with 10% for validation and 20% masked as test nodes.

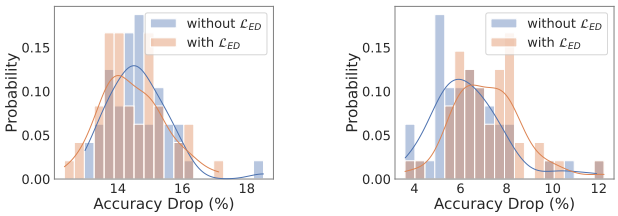
Attack budgets for manipulation and node injection. For PRBCD, we evaluate different perturbation rates (3%, 5%, and 10%) as shown in Tables 2 and 5. For TDGIA, we inject 40 malicious nodes with 3 edges each on Cora, 5000 malicious nodes with 10 edges each on Flickr, and 1000 malicious nodes with 50 edges each on OGB-Arxiv.

Parameter configuration. Following previous studies Dai et al. (2023b), we use a 2-layer GCN as each MoE expert as well as the gating model. The hidden dimension is set to 32 for all datasets. The learning rate is set to 0.01 with 5e-4 weight decay and we use an Adam optimizer to train 200 epochs for all baseline methods. All experiments are conducted on 8 A6000 GPUs.

H Additional robustness analysis under diversity regularization

In this section, we give the formulation of the representation diversity loss $\mathcal{L}_{ED}$ proposed in Liu et al. (2023) as well as its impact on experts' robustness distribution under PRBCD and TDGIA attacks. Given a node v and its L-hop neighbors, this loss enriches the diversity of learned node embeddings by maximizing the agreement between its neighborhood nodes while pushing away the irrelevant nodes that are far away. The procedure is formulated as follows:

$$\mathcal{L}_{ED} = -\log \frac{\sum_{j \in \mathcal{N}(v)} \exp(\mathrm{sim}(z_v, z_j))}{\sum_{k \notin \mathcal{N}(v)} \exp(\mathrm{sim}(z_v, z_k))}, \qquad (20)$$



(a) PRBCD (b) TDGIA

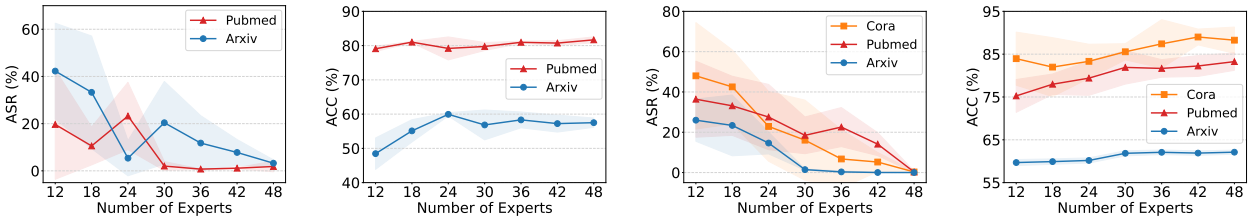
Figure 8: Impacts of existing diversity loss on the proportion of robust individual experts under PRBCD and TDGIA.

Table 8: Additional results in defending against feature-based attack (SPEAR) on OGB-Arxiv.

Dataset	Metric	GCN	Prune	OD	RIGBD	RGCN	DP-MoE	Ours
Arxiv	ASR	97.38 ± 0.01	98.45 ± 0.37	64.23 ± 15.23	45.27 ± 0.30	96.39 ± 0.17	75.83 ± 8.25	1.72 ± 0.77
	ACC	64.30 ± 0.30	60.45 ± 0.66	61.29 ± 0.38	55.52 ± 1.20	60.26 ± 1.46	62.86 ± 0.66	64.39 ± 4.93

Table 9: Additional ablation across datasets under DPGBA.

Method	Cora		Pubmed		Flickr		OGB	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
wo Both	77.78 ± 3.57	91.48 ± 0.30	82.20 ± 0.18	82.36 ± 0.59	77.34 ± 6.62	44.57 ± 1.42	97.56 ± 0.56	61.29 ± 1.58
wo Re-routing	61.19 ± 2.72	86.17 ± 2.81	25.25 ± 2.47	79.55 ± 1.44	76.09 ± 5.82	44.87 ± 1.63	2.00 ± 0.27	61.34 ± 0.93
wo Div	75.41 ± 4.54	91.23 ± 0.63	82.00 ± 2.71	81.11 ± 0.63	6.31 ± 8.92	45.60 ± 2.81	90.87 ± 7.25	60.79 ± 1.56
Ours	0.30 ± 0.42	88.27 ± 4.06	0.62 ± 0.33	82.61 ± 1.70	0.01 ± 0.01	45.14 ± 2.55	0.02 ± 0.02	62.11 ± 0.48



(a) ASR under GTA (b) ACC under GTA (c) ASR under DPGBA (d) ACC under DPGBA

Figure 9: Impact of model capacity under GTA and DPGBA attacks.

where z represents the learned node embedding. The function sim() calculates the similarity between two vectors, i.e.:

$$\text{sim}(z_i, z_j) = \frac{z_i^{\mathrm{T}} z_j}{||z_i||_2 ||z_j||_2}. \tag{21}$$

The impact of $\mathcal{L}_{ED}$ to the robustness of individual experts is illustrated in Fig. 8.

I Additional Experimental Results

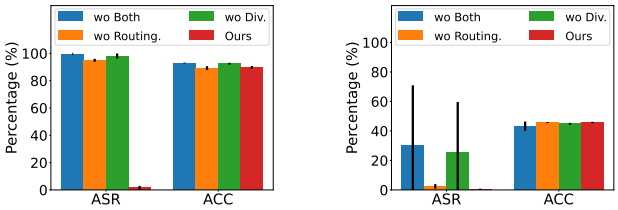
Additional Results of Edge Manipulation and Node Injection Attacks. In this subsection, we report the additional experimental results under PRBCD and TDGIA on the small Cora dataset. The results are reported in Tab.5 and Tab. 6, respectively.

Additional Results of Feature-based Attacks. In this subsection, we report the additional experimental results of RGMoEon feature-based adversarial attacks, i.e., Nettack and SPEAR. The results are reported in Tab. 7 and Tab. 8.

Additional Results of Impact of Model Capacity. In this subsection, we report the additional results of impact of model capacity under GTA and DPGBA attack in Fig. 9.

Additional Results of Ablation Study.

In this subsection, we report the additional ablation study results on Cora and Flickr under UGBA in Fig. 10. The ablation results under DPGBA are reported in Tab. 9.



(a) Cora (b) Flickr

Figure 10: Additional ablation on UGBA on Cora and Flickr.

J Detailed Related Work on Defense against Graph Adversarial attacks

As GNNs are vulnerable to adversarial attacks, various robust graph neural networks against adversarial attacks have been proposed Zhang et al. (2019). GCN-Jaccard Wu et al. (2019) defends against adversarial attacks by eliminating the edges connecting nodes with low Jaccard similarity of node features. RobustGCN Zhu et al. (2019) is to model Gaussian distributions as hidden layers to absorb the effects of adversarial attacks in the variances. Inspired by the fact that adversarial attacks will lead to high-rank adjacency matrix, ProGNN Jin et al. (2020) proposes to learn a low-rank clean adjacency matrix where the adversarial

edges are likely to be removed. GNNGuard Zhang & Zitnik (2020) computes the attention scores based on the cosine similarity of node representations from last layer. With the similarity-based attention, the adversarial edges are likely to be assigned with small weights since they generally link dissimilar nodes. In MedianGNN Chen et al. (2021), a median aggregation mechanism is designed to improve the robustness of GNNs. RSGNN Dai et al. (2022) adopts the node attributes and supervision from the noisy edges to denoise and dense graph, ensuring the robustness with noisy edges and limited labels. Dai et al. (2021) introduces to learn a robust GNN on sparse and noisy labels. Dai et al. (2023a) proposes a unified information bottleneck framework for robustness against adversarial attack and membership inference.

To address the susceptibility of GNNs to the emerging backdoor attacks, basic defense methods have been recently proposed to remove the trigger from the poisoned dataset. Prune Dai et al. (2023b) reomve edges that connect nodes with low similarity. OD Zhang et al. (2024a) which involves training a graph auto-encoder and filtering out nodes with high reconstruction loss. RIGBD Zhang et al. (2024b) identifies the large prediction variance of poisoned nodes under edge dropping and designs a convolution operation to distinguish poisoned nodes from clean nodes. They also propose a novel training strategy to train a backdoor-robust GNN model. However, the majority of robust GNNs focus on the general manipulation attacks/node injection attacks. The defense on graph backdoor is rather limited let alone a unified framework to defend them simultaneously.