Fend for Yourself! Backdoor Purification in Federated Graph Learning with an Evolving Knowledge Anchor

Chengcheng Zhu
Nanjing University

Yunlong Mao *
Nanjing University

Jiale Zhang
Yangzhou University

Bosen Rao
Yangzhou University

Sheng Zhong
Nanjing University

Abstract

Federated Graph Learning (FedGL) enables collaborative training on decentralized graph data while preserving privacy, yet its distributed nature makes it highly vulnerable to backdoor attacks. These attacks compromise the integrity of the global model by injecting malicious triggers. Existing defenses, however, are often ineffective on complex graph data or rely on a trusted server, creating an architectural conflict with modern privacy-preserving technologies. To overcome these limitations, we propose GBHINDER, a novel and practical trusted-server-free defense framework where each benign participant defends itself. GBHINDER establishes a virtuous cycle: it leverages its own trusted historical knowledge as a benign anchor to purify the downloaded global model, and in turn, selectively incorporates the global model's benign knowledge to progressively evolve the anchor itself. Specifically, this cycle is driven by two key components. A Historical Channel Attention Regularization module uses the anchor to constrain the global model's representations and disrupt backdoor propagation. To resolve the tension between local trust and global collaboration, an Adaptive Momentum Information Update mechanism enables the anchor to safely evolve by dynamically integrating robust global information, ensuring the anchor remains effective with federated iteration. Extensive experiments on several benchmark datasets demonstrate that GBHINDER significantly outperforms state-of-the-art (SOTA) defenses, successfully reducing the backdoor attack success rate to below 10% while preserving high accuracy on the main task.

1 Introduction

Graph Neural Networks (GNNs) have emerged as a promising avenue for effectively learning from complex graph data [13, 34, 40], which encapsulates rich topological structures and node features. However, such graph data often contains sensitive private information, such as a customer's transaction records in e-commerce [9, 45], user profiles in banking [7, 31], or patient data in healthcare institutions [20]. Due to stringent privacy policies and fierce business competition, sharing raw graph data between platforms is often infeasible [12], leading to the formation of data silos [21]. To overcome this bottleneck, Federated Graph Learning (FedGL) [5, 32] has been proposed, integrating the strengths of GNNs and Federated Learning (FL) [22, 42] to collaboratively train a shared global model while preserving data privacy. Despite its numerous advantages [11], the distributed nature of FedGL introduces additional vulnerabilities, particularly to backdoor attacks [1, 36]. In this threat model, malicious participants can train local models on poisoned samples embedded with triggers, and when these compromised models are aggregated, the global model inherits the backdoor function.

With the objective of better defending against backdoor attacks, a variety of defense methods have been widely studied in FL contexts [3, 10, 23, 24]. A predominant approach is to detect outliers by computing discrepancies between model weights, gradients, or other metrics. However, such general-purpose defenses exhibit inherent fragility when applied to graph data [37]. The complex topological structures and inherent non-independent and identically distributed (non-IID) characteristics of graph data cause gradients of benign clients to be naturally diverse, blurring the distinction between malicious deviations and benign heterogeneity [30]. Furthermore, the arbitrary placement and shape of triggers, facilitated by graphs' structural flexibility, further exacerbate this challenge and make identifying poisoned models exceedingly difficult.

Urgently, defenses specifically designed for FedGL remain in early stages. To our knowledge, only two dedicated defense mechanisms, GNNCert [43] and FedTGE [30], have been proposed. GNNCert offers a certified defense that provides provable robustness, ensuring correct predictions even for inputs with triggers. However, this guarantee is achieved by analyzing localized graph segments, which can lead to a drop in prediction accuracy as crucial global context from the original graph structure is lost. FedTGE enhances detection by using energy-based models to distinguish benign and

*Corresponding author, email: maoyl@nju.edu.cn.

malicious samples at the client level, then applies server-side clustering and weighted aggregation to suppress threats. Nevertheless, its energy model focuses on node-level features and local topologies, failing to discern global structural differences, which degrades performance in graph-level tasks.

Critically, existing FedGL defenses are architecturally dependent on a trusted central server to orchestrate security. This reliance, however, introduces two fundamental flaws. First, it creates a severe performance and scalability bottleneck. The federated server, traditionally a lightweight aggregator, is burdened with computationally intensive defense tasks, such as pairwise model comparison or feature analysis for anomaly detection. This additional overhead fundamentally compromises the efficiency of the FL paradigm, limiting its applicability in large-scale federations with numerous participants. More fundamentally, this server-centric paradigm is in direct philosophical conflict with the evolution of privacy-preserving FL. Server-side verification inherently requires access to, or inspection of, individual client updates. This requirement directly nullifies the guarantees of privacy-enhancing technologies like secure multi-party computation (SMC) [6] or differential privacy (DP) [33], which are expressly designed to conceal these very updates from the server. Consequently, the prevailing server-reliant security paradigm forces a false dichotomy between security and privacy. This architectural impasse severely restricts the practical deployment of robust FedGL in sensitive, real-world applications and necessitates a paradigm shift in defense design.

To overcome this impasse, we pivot from the trusted server-reliant paradigm and propose a novel defense mechanism founded on a new principle: empowering each benign client to defend itself. However, realizing this client-centric vision poses two fundamental technical challenges:

Challenge 1 : *How to defend with limited local information?* From the point of an isolated client, the aggregated global model is an untrusted black box. Lacking visibility into the updates from other participants, how can a client reliably distinguish between benign global knowledge and malicious patterns using only its local data and historical model? To address this, GBHINDER introduces Historical Channel Attention Regularization, which leverages the client's trusted historical knowledge as a benign anchor. This anchor acts as a personalized, trusted baseline to regularize the downloaded global model, thereby disrupting the injection of malicious functionality without needing to inspect other clients.

Challenge 2 : *How to balance local trust with global collaboration?* A purely self-reliant defense risks model divergence and knowledge isolation. If a client over-relies on its static local anchor, it may reject beneficial global updates from other benign peers, fail to generalize, and ultimately undermine the collaborative goal of FL. To resolve this critical tension, GBHINDER introduces an Adaptive Momentum Information Update mechanism. This component enables the benign anchor to dynamically and safely evolve by selectively integrating robust knowledge from the global model. It ensures that the client benefits from the collective knowledge of the federation without inheriting its vulnerabilities. These components work in synergy to create a virtuous cycle of purification and evolution on the client side, delivering a practical and trusted server-free defense for FedGL.

We conduct extensive evaluations on several benchmark datasets for graph tasks to validate the effectiveness of GBHINDER. We compare our method against a wide array of baselines, including general-purpose defenses from traditional FL and state-of-the-art (SOTA) methods specifically designed for FedGL. The experimental results demonstrate that GBHINDER significantly outperforms existing defense mechanisms, successfully reducing the backdoor attack success rate to below 10% while preserving high accuracy on the main task, all without requiring a trusted server or auxiliary data. Furthermore, we systematically investigate the robustness of our method under various challenging scenarios, including different backdoor trigger types and sizes, and varying proportions of poisoned injection and malicious clients. We also assess its applicability across diverse FL settings, architectures, tasks, and overhead. Finally, comprehensive ablation studies and parameter sensitivity analyses are presented to validate the crucial role of each component within our framework.

The main contributions are summarized as follows:

- **A Novel Trusted-Server-Free Defense Paradigm.** We propose GBHINDER, a novel client-side defense mechanism for FedGL that leverages trusted local historical knowledge to mitigate backdoor attacks. By eliminating the dependency on a trusted central server, GBHINDER significantly enhances its practicality and deployability in real-world scenarios.

- **Historical Channel Attention Regularization.** GBHINDER introduces a channel-wise attention mechanism incorporating topological information, which magnifies differences between benign and malicious neurons and regularizes the global model by aligning its deep-layer representations with benign shallow-layer features from historical models. Combined with a topological consistency loss, it effectively disrupts the propagation of backdoor patterns within the GNN.

- **Adaptive Momentum Information Update.** To resolve the tension between local self-defense and global collaboration, we propose an adaptive momentum update strategy, which enables the client's benign anchor to safely evolve by selectively incorporating robust knowledge from the global model based on a perturbation-based stability metric. This ensures the anchor remains reliable and effective throughout the training process.

- **Comprehensive Empirical Validations.** We conduct extensive experiments on several benchmark graph

datasets, demonstrating that GBHINDER reduces backdoor attack success rates to below 10% while maintaining high main task accuracy. Our evaluations, including comparisons with SOTA defenses, robustness tests across various backdoor and federated learning settings, parameter sensitivity studies, ablations, and overhead measurements, confirm its practical feasibility.

The remainder of this paper is organized as follows. Section 2 introduces the related works. The problem formulation and proposed method are discussed in Section 3 and 4. Section 5 evaluates and analyzes the experimental results. Finally, Section 6 concludes the paper.

2 Related Work

2.1 Backdoor Attacks to FedGL

Backdoor attacks on GNNs first emerged in centralized settings, where early research focused on designing subgraph triggers and optimizing their injection [38, 50]. The primary evolutionary trend has been toward enhancing stealth and effectiveness. Trigger design has progressed from simple random subgraphs to structured motifs [52] and instance-specific triggers, as seen in the Graph Trojan Attack (GTA) [35]. To further minimize detectability, advanced attacks such as the Unnoticeable Graph Backdoor Attack (UGBA) [8] and the Distribution-Preserving Backdoor Attack (DPBA) [51] were developed to preserve the statistical distributions of the original graph data, allowing the trigger to blend in seamlessly.

The distributed nature of FedGL presents a fertile new ground for these threats. Initial work adapted centralized attacks to the federated paradigm but still relied on static, randomly generated triggers [37]. A significant leap was made by Opt-GDBA [43], which introduced an adaptive generator to optimize trigger structure and placement specifically for the FedGL context, achieving more effective and stealthy attacks. However, these methods all fall under the category of data-poisoning attacks, which require direct modification of training data and risk degrading the model's main-task performance. A more recent and insidious paradigm is the non-intrusive attack. NI-GDBA [17], for example, circumvents data modification entirely by training a client-side perturbation generator to induce a "natural" backdoor, posing a more fundamental threat to the integrity of the learning process.

2.2 Backdoor Defenses to FedGL

Recent theoretical works have sought to build a foundational understanding of backdoor defense, primarily in the computer vision domain [49]. In a centralized setting for graph data, methods like DShield [47] have proven effective in achieving backdoor-free training by directly identifying and filtering poisoned nodes from the dataset. However, such data purification strategies are fundamentally incompatible with the FL paradigm, where privacy constraints render client data inaccessible to the server. Consequently, research in general FL has shifted towards robust aggregation approaches that aim to mitigate the effect of malicious updates without direct data inspection [3, 10, 18, 23, 24]. These methods typically identify and exclude outlier updates by comparing model weights or gradients. For instance, FoolsGold [10] and Flame [23] employ similarity-based filtering, while RLR [24] adjusts aggregation weights based on directional consistency. However, these general-purpose defenses struggle when applied to FedGL [37]. The complex topologies and inherent non-IID nature of decentralized graph data cause benign client updates to be naturally diverse, making it difficult to distinguish between honest heterogeneity and adversarial manipulations [30].

Recognizing these limitations, research on defenses specifically tailored for FedGL is emerging, though still nascent. One notable approach is GNNCert [43], a certified defense that provides provable robustness guarantees by employing graph partitioning and ensemble voting. However, this theoretical guarantee often comes at the cost of reduced main-task accuracy and significant inference overhead. Another key work, FedTGE [30], uses an energy-based model to distinguish benign and malicious patterns, enabling the server to perform weighted aggregation. While promising, its performance can be unstable across different graph tasks. Crucially, both of these defense paradigms rely on the assumption of a trusted central server to orchestrate the voting or aggregation process. In contrast, this paper aims to explore a new defense paradigm: a client-deployed solution for FedGL that relies solely on local information to defend against backdoor attacks, thereby eliminating the dependency on a trusted server.

3 Problem Formulation

3.1 Federated Graph Learning

We define an attributed graph as $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{X} = \{x_i\}_{i=1}^{|\mathcal{V}|} \subseteq \mathbb{R}^d$ is the set of node features, with x_i representing the feature vector of node v_i. Let $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ denote the adjacency matrix of G, where $\mathbf{A}_{u,v} > 0$ if $(u,v) \in \mathcal{E}$, and 0 otherwise. For unweighted graphs, $\mathbf{A}_{u,v}$ is simply 1 for existing edges. The focal task of this study is graph classification, where each graph G is assigned a label y from the label space $\mathcal{Y}$. A graph classifier F takes a graph G as input to predict its label, formalized as $F : G \to \mathcal{Y}$.

FedGL enables $\mathcal{M}$ clients to collaboratively train a global model w without revealing their local datasets. The training process involves clients training local models on their respective datasets, $\mathcal{D}_k = \{(G_{k,i}, y_{k,i})\}_{i=1}^{|\mathcal{D}_k|}$, and then uploading their model weights to a parameter server. Specifically, the global

training objective is defined as follows:

$$w = \arg\min_{w} \sum_{k=1}^{|\mathcal{M}|} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \ell_k(w, \mathcal{D}_k), \tag{1}$$

$$\ell_k(w, \mathcal{D}_k) = \frac{1}{|\mathcal{D}_k|} \sum_{i=1}^{|\mathcal{D}_k|} f(w, (G_{k,i}, y_{k,i})), \tag{2}$$

where w denotes the optimal global model parameters, $\ell_k(w, \mathcal{D}_k)$ represents the average loss computed over the dataset $\mathcal{D}_k$ for client k, $(G_{k,i}, y_{k,i})$ denotes the i-th sample in $\mathcal{D}_k$, and $|\mathcal{D}| = \sum_{k=1}^{|\mathcal{M}|} |\mathcal{D}_k|$.

At the t-th federated training round, the server randomly selects a client set S_t where $|S_t| = m$ and $0 < m \leq |\mathcal{M}|$, and broadcasts the current model parameters w^t. The selected clients perform local training in the following three steps:

- Global model download. Selected clients download the global model w^t from the server.
- Local training. Each client updates its local model, initialized with w^t, by training on its local dataset: $w_k^t \leftarrow w_k^t - \eta \nabla_{w_k^t} \ell(w_k^t; b)$, where η and b are the learning rate and a local mini-batch, respectively.
- Aggregation. After the clients upload their local models $\{w_k^{t+1} \mid k \in S_t\}$, the server updates the global model by aggregating the local models.

$$w^{t+1} \leftarrow \mathbf{AGG}(\{w_k^{t+1} \mid k \in S_t\}). \tag{3}$$

Note that $\mathbf{AGG}(\cdot)$ is the predefined aggregation method, such as FedAvg [22], etc.

3.2 Threat model

We follow the threat model considered in previous studies on backdoor attacks in FedGL [17, 37, 43]. Specifically, we define the goals, capabilities, and background knowledge for both the adversary and the defender.

3.2.1 Adversary's Goal

The adversary's primary objective is to inject a malicious function, often referred to as a neural trojan, into the global graph model. This embedded trojan is designed to remain dormant and undetectable during standard inference on clean data. However, it becomes activated when a specific, predefined pattern known as a trigger is present in the input, causing the model to misclassify the input to a target label chosen by the adversary. Formally, let F_w represent the clean model and $\tilde{F}_w$ denote the backdoored model. The adversary's objectives can be summarized as:

$$\begin{cases} F_w(G) = \tilde{F}_w(G) \\ \tilde{F}_w(G \oplus \delta) = \hat{y} \end{cases}, \tag{4}$$

where δ is the trigger and $\hat{y}$ is the target output chosen by the adversary. In FedGL, a malicious client, given a local dataset $\mathcal{D}_k$, aims to compromise a fraction r of the data by injecting triggers δ, thereby constructing a backdoor dataset $\mathcal{D}_{k,b}$, such that $\mathcal{D}_k = \mathcal{D}_{k,c} \cup \mathcal{D}_{k,b}$ and $|\mathcal{D}_{k,b}| = r \cdot |\mathcal{D}_k|$, where $\mathcal{D}_{k,c}$ represents the clean subset. The adversary trains a backdoored local model $\tilde{F}_{w_k}$ by minimizing the empirical loss, defined as:

$$\begin{aligned} \mathbb{E}_{(G,y)\sim\mathcal{D}_k}[\ell(F_{w_k}(G), y)] = \mathbb{E}_{(G,y)\sim\mathcal{D}_{k,c}}[\ell(F_{w_k}(G), y)] \\ + \mathbb{E}_{(G,\hat{y})\sim\mathcal{D}_{k,b}}[\ell(F_{w_k}(G), \hat{y})]. \end{aligned} \tag{5}$$

3.2.2 Adversary's Capability and Knowledge

We assume the adversary controls a subset of malicious clients and can fully manipulate their local training processes. This includes modifying training data, altering model parameters, and controlling the optimization procedure. Malicious clients possess their own training graphs and have access to the shared global model during FedGL training. However, the adversary lacks access to or the ability to modify any information associated with benign participants, such as their private local datasets, model updates, or training procedures.

3.2.3 Defender's Capability and Goal

In our framework, every benign participant acts as a defender. Our proposed method is deployed on the client side, empowering each honest client to protect itself. We assume that all benign clients adhere strictly to the proposed FedGL protocol, whereas malicious clients do not. Notably, our defense strategy does not rely on additional assumptions, such as the availability of a trusted server or a public clean dataset, which are often impractical in real-world federated settings. The defender's goal is to produce a final global model w that is not only robust against such backdoor attacks but also maintains high performance on its primary task with clean data.

4 Methodology

4.1 Defense Intuition and Overview

4.1.1 Key Intuition

In FedGL, benign clients possess trustworthy local data, and local training can be intuitively viewed as fine-tuning the downloaded global model using this clean data to mitigate backdoor attacks. Intuitively, such continuous fine-tuning would gradually overwrite malicious patterns embedded in the global model. However, our preliminary investigations reveal this intuition to be deceptive. As shown in Figure 1, we conducted experiments where three types of backdoor attacks were applied for 10 rounds, followed by 200 additional rounds of FL without malicious participants. The results demonstrate that naive fine-tuning has limited effectiveness in eliminating deeply embedded backdoors. This ineffectiveness stems

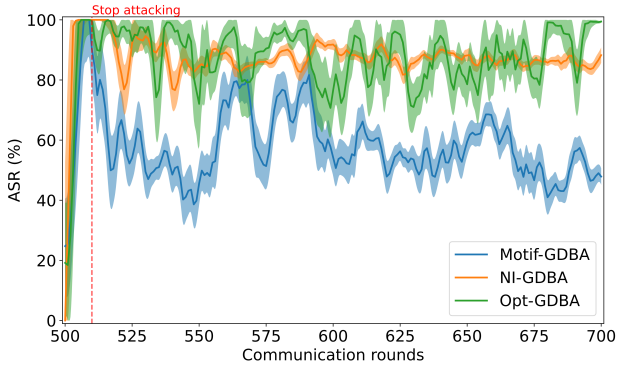

Figure 1: The effectiveness of backdoor attacks in FedGL.

from two primary challenges. ❶ *Backdoor Invisibility*. By design, backdoors remain dormant and trigger-free on a benign client's local data. Consequently, the standard training objective, such as cross-entropy loss, is effectively blind to the malicious functionality. It lacks any incentive or gradient signal to penalize the neurons responsible for the backdoor, as it cannot optimize for a problem it does not observe.❷ *Local Data Insufficiency*. Each client's local dataset offers only a partial and often biased (non-IID) view of the complete data distribution. In many cases, certain classes may be underrepresented or absent, making it difficult to use limited and incomplete clean data to generate gradients powerful enough to overwrite backdoor patterns embedded in the global model.

These motivate our paradigm shift from passive, implicit fine-tuning to an active regularization-based framework. GBHINDER introduces this shift by empowering each benign client to fend for itself, achieved by establishing a virtuous cycle centered around an evolving knowledge anchor.

4.1.2 Overview

As illustrated in Figure 2, our approach, denoted as GBHINDER, operates this cycle through two synergistic components:

- In the Historical Channel Attention Regularization module, the client uses its trusted historical model as a benign anchor to impose explicit constraints on the downloaded global model. By designing a channel attention mechanism that accounts for the topological properties of graph data, we amplify the differences between benign and malicious neuron activations, aligning the locally purified global model's deep-layer representations with benign shallow-layer features from the trusted historical model. Additionally, a topological consistency loss enforces feature alignment for adjacent nodes, mitigating anomalous patterns introduced by backdoor attacks.

- In the Adaptive Momentum Information Update module, to ensure the anchor remains potent and aligned with the benign evolution of the federation, we dynamically update historical model parameters using a momentum-based approach in each training epoch. A perturbation-based metric is introduced to quantify the locally purified global model's susceptibility to backdoor triggers, adjusting the momentum coefficient to selectively incorporate robust, benign knowledge into the anchor.

4.2 Historical Channel Attention Regularization

GBHINDER is built on the principle of leveraging each client's own trusted history. In FedGL, the server initializes and distributes a global model to selected clients for local training. Upon selection in round t, a benign client initializes its historical anchor using its most recent local model, i.e., $h_k^t \leftarrow w_k^{t-1}$. If a client is selected for the first time, it instead trains a clean model on its local data to serve as an inaugural trusted anchor. The historical anchor is used to regularize the downloaded global model during local training. The resulting current local model (locally purified global model), w_k^t, is then preserved to serve as the new anchor for the client's next training round. If backdoors are effectively purified during the local update, the local model can remain benign, providing a trustworthy anchor for the future. This insight motivates our core strategy: explicitly regularize the potentially compromised global model w^t using the previous local model (the historical anchor). However, due to the stealthy nature of backdoors, the representations generated by backdoored and benign models on clean inputs can be nearly indistinguishable, which limits the effectiveness of naive representation-matching and motivates the design of a more targeted regularizer.

To address this, we design a channel attention mechanism whose intuition is that while overall representations might appear similar, the underlying attention patterns of neurons, especially across different channels, will diverge. It is designed to amplify the differences in channel-wise importance between benign and malicious neuronal activations and force the latter to align with the former. Specifically, for a GNN model F, we denote the activation tensor at the l-th layer as $\mathcal{F}^l \in \mathbb{R}^{N \times D_l}$, where D_l is the output channel dimension and N is the number of nodes. We define a channel attention function $\mathcal{H} : \mathbb{R}^{N \times D_l} \to \mathbb{R}^N$, which maps the activation tensor to an attention representation. Recognizing the topological properties of graphs, we incorporate a topological weight to emphasize the contribution of structurally important nodes (i.e., high-connectivity nodes). We use the normalized node degree as a topological weight, formalized as:

$$\mathbf{D}[i] = \frac{\deg(v_i)}{\max_{v \in \mathcal{V}} \deg(v)}, \tag{6}$$

where $\deg(v_i)$ is the degree of node v_i. The attention map for a specific channel d is then computed by calculating, for each node, the squared activation weighted by the node's

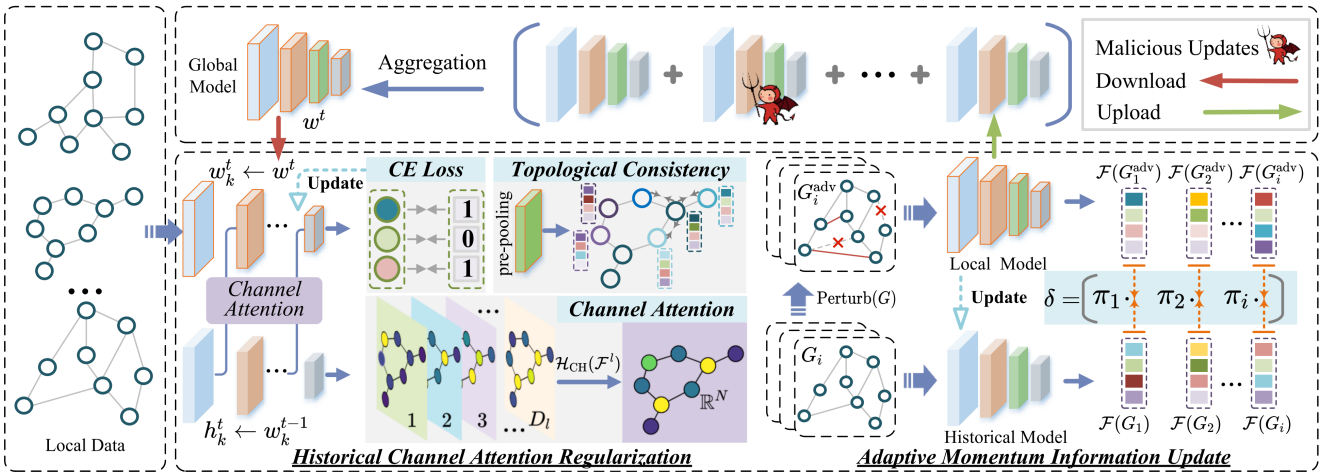

Figure 2: Overview of the proposed GBHINDER framework.

topological importance:

$$\left[\mathcal{H}_d(\mathcal{F}^l)\right]_i = \mathbf{D}[i] \cdot |\mathcal{F}^{l(i,d)}|^2, \quad \text{for } i = 1, \ldots, N \tag{7}$$

where $\mathcal{F}^{l(i,d)}$ represents the activation of node i in channel d at layer l. A simple approach would be to average these attention maps across all channels. However, this overlooks the fact that backdoor neurons may be concentrated in specific channels. To account for the heterogeneous importance of channels in representing benign versus malicious features, we calculate channel-specific weights to highlight salient channels and suppress noisy ones. The weight score for channel d is computed as:

$$\omega_d = \frac{\|\mathcal{F}^{l(:,d)}\|_1}{\sum_{j=1}^{D_l} \|\mathcal{F}^{l(:,j)}\|_1}, \tag{8}$$

where $\|\mathcal{F}^{l(:,d)}\|_1 = \sum_{i=1}^{N} |\mathcal{F}^{l(i,d)}|$ represents the L_1 norm of the activation values in channel d. The final channel attention operator is the weighted sum of per-channel attention maps:

$$\mathcal{H}_{\text{CH}}(\mathcal{F}^l) = \sum_{d=1}^{D_l} \omega_d \cdot \mathcal{H}_d(\mathcal{F}^l) \quad \in \mathbb{R}^N. \tag{9}$$

To ensure consistent scaling and enable comparison across different layers, we apply L_2 normalization to the resulting attention representations, a critical step for stable feature regulation. This is defined by the function $\psi(\cdot)$:

$$\psi(\mathcal{H}_{\text{CH}}(\mathcal{F}^l)) = \frac{\mathcal{H}_{\text{CH}}(\mathcal{F}^l)}{\|\mathcal{H}_{\text{CH}}(\mathcal{F}^l)\|_2}. \tag{10}$$

Given the message-passing nature of GNNs, shallow layers capture local topological patterns, while deeper layers integrate global information. Backdoor features often accumulate and become more pronounced in deeper layers through multi-hop message propagation. By using the channel attention of shallow layers from the historical anchor to constrain the deep-layer attention of the locally purified global model, we can disrupt this propagation, further preventing amplification of malicious information and suppressing backdoor neurons. For a model with L layers, the loss function for alignment is:

$$\mathcal{L}_{\text{align}}(h_k^t, w_k^t) = \sum_{l=1}^{L-1} \left\| \psi\left(\mathcal{H}_{\text{CH}}(\mathcal{F}_{h_k^t}^l)\right) - \psi\left(\mathcal{H}_{\text{CH}}(\mathcal{F}_{w_k^t}^{l+1})\right) \right\|_2, \tag{11}$$

where h_k^t denotes the historical anchor parameters, and w_k^t represents the client local model, i.e., locally purified global model. Additionally, backdoor attacks often disrupt the topological consistency of graphs by introducing anomalous edges or modifying node features, violating the natural similarity between neighboring nodes in real graph data. To counter this, we enforce consistency in the feature representations of neighboring nodes in the global model using local clean data. The topological consistency loss is defined as:

$$\mathcal{L}_{\text{topo}}(w_k^t) = \frac{1}{\sum_{(i,j)\in\mathcal{E}} \mathbf{A}_{i,j}} \sum_{(i,j)\in\mathcal{E}} \left\| \mathbf{z}_{w_k^t}^{(i)} - \mathbf{z}_{w_k^t}^{(j)} \right\|_2^2 \mathbf{A}_{i,j}, \tag{12}$$

where $\mathbf{z}^{(i)}$ denotes the feature representation of node i from a pre-pooling layer of the model, $\mathcal{E}$ is the set of edges, and $\mathbf{A}_{i,j}$ is the adjacency matrix entry for nodes i and j. To reduce computational overhead, we randomly sample a subset of edges for this loss computation. The overall regularization loss combines the cross-entropy loss $\mathcal{L}_{\text{ce}}$ with the proposed constraints:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda_1 \cdot \mathcal{L}_{\text{align}} + \lambda_2 \cdot \mathcal{L}_{\text{topo}}, \tag{13}$$

where λ_1 and λ_2 are hyperparameters balancing the contributions of each term.

4.3 Adaptive Momentum Information Update

While Historical Channel Attention Regularization is effective, its performance hinges on the quality of the historical model. Relying solely on a static historical anchor is risky. Each client's local data represents only a partial view of the global distribution, and the heterogeneity of graph structures across clients can lead to significant variations in historical knowledge. Moreover, since the random client selection in FedGL, a benign client might be dormant for multiple rounds. Such a biased or outdated anchor is not only an inadequate standard for purifying the evolving global model, but it also forces the client into knowledge isolation, leading to model divergence and undermining the collaborative goal of federation. To address these issues, we propose an Adaptive Momentum Information Update mechanism to dynamically refresh local historical knowledge using robust global information.

Specifically, we update the historical anchor every training epoch using a momentum-based approach:

$$h_k^t = a \cdot h_k^t + (1-a) \cdot w_k^t, \tag{14}$$

where a is the momentum coefficient, controlling the balance between retaining historical knowledge and incorporating updates from the locally purified global model. A larger a prioritizes the historical model, while a smaller a incorporates more global information. The choice of this momentum coefficient becomes a critical factor. We cannot assume that our regularization can completely eliminate the backdoor influence in a single round. Blindly absorbing the updated global model is risky, as residual backdoor information could poison the historical anchor over time. Therefore, we dynamically adjust a based on the perceived stability and trustworthiness of the purified global model, ensuring selective incorporation of global knowledge from it.

To quantify this trustworthiness, we assess the locally purified global model's robustness by measuring its sensitivity to topological perturbations that mimic backdoor attacks. We define a perturbation function $G^{\text{adv}} = \text{Perturb}(G)$ to create an adversarial graph G^{adv}, which randomly adds or removes a certain ratio of edges in the graph G, where each edge modification follows an i.i.d. uniform distribution as in prior work [46]. This simulates backdoor perturbations by testing the robustness of the model against connectivity pattern variances. Moreover, accounting for data heterogeneity, we weight the importance of each graph by its size (number of nodes), as larger graphs contribute more significantly to the overall representation. The graph importance weight is:

$$\pi_i = \frac{|\mathcal{V}_i|}{\sum_{G_j \in \mathcal{D}_k} |\mathcal{V}_j|}, \tag{15}$$

where $|\mathcal{V}_i|$ is the number of nodes in G_i, and $\mathcal{D}_k$ is local dataset of client k. The discrepancy between the purified global and historical models under perturbation is computed as:

$$\delta = \sum_{G_i \in \mathcal{D}_k} \pi_i \cdot \left\| \psi\left(\mathcal{F}_{w_k^t}(G_i^{\text{adv}})\right) - \psi\left(\mathcal{F}_{h_k^t}(G_i)\right) \right\|_2^2, \tag{16}$$

where $\mathcal{F}_w$ denotes the graph-level embedding and δ measures the purified global model's instability under perturbations. To improve efficiency, we approximate the stability score δ by using a randomly sampled mini-batch from local data. A high δ indicates that the locally purified global model is sensitive to topological changes, suggesting potential backdoor contamination, prompting the historical model to rely more on its own knowledge to resist malicious updates. Conversely, a small δ signifies a robust model, allowing the client to safely absorb more of its updated knowledge. Finally, to ensure a stable update, we use a non-linear exponential growth function to

Algorithm 1: GBHINDER Algorithm

Input: Client set $\mathcal{M}$, communication rounds T, local epochs E, learning rate η; Initial momentum coefficient a_0, sensitivity factor γ; Local dataset $\mathcal{D}_k$ for each client k.
Output: Final global model w^T.

Initialize global model parameters w^0;
Server Executes:
for *each communication round* $t \in [0, \ldots, T-1]$ **do**
  Select a random subset of clients S_t from $\mathcal{M}$;
  for *each client* $k \in S_t$ ***in parallel*** **do**
    $w_k^{t+1} \leftarrow \text{ClientUpdate}(k, w^t)$;
  $w^{t+1} \leftarrow \sum_{k \in S_t} \frac{|\mathcal{D}_k|}{\sum_{j \in S_t} |\mathcal{D}_j|} w_k^{t+1}$
Client Executes:
Function ClientUpdate(k, w^t):
Initialize local model: $w_k^t \leftarrow w^t$;
Initialize historical anchor:$h_k^t \leftarrow w_k^{t-1}$;
for *each local epoch* $e \in [1, 2, \ldots, E]$ **do**
  for *each batch* $b \subset \mathcal{D}_k$ **do**
    Compute $\mathcal{L}_{\text{ce}}(w_k^t; b)$;
    Compute $\mathcal{L}_{\text{align}}(h_k^t, w_k^t; b)$ based on Eq. 11;
    Compute $\mathcal{L}_{\text{topo}}(w_k^t; b)$ based on Eq. 12;
    $\mathcal{L}_{\text{total}} = \frac{1}{|b|}\left(\mathcal{L}_{\text{ce}} + \lambda_1 \cdot \mathcal{L}_{\text{align}} + \lambda_2 \cdot \mathcal{L}_{\text{topo}}\right)$;
    $w_k^t \leftarrow w_k^t - \eta \nabla \mathcal{L}_{\text{total}}$;
  `// Adaptive Momentum Update`
  for *each* G_j *in a random mini-batch* $b' \subset \mathcal{D}_k$ **do**
    $G_j^{\text{adv}} \leftarrow \text{Perturb}(G_j)$;
    $\pi_j \leftarrow \frac{|\mathcal{V}_j|}{\sum_{G_m \in b'} |\mathcal{V}_m|}$;
  Compute δ based on Eq. 16 ;
  $a \leftarrow a_0 \cdot (1 - \exp(-\gamma \cdot \delta))$;
  Update anchor: $h_k^t \leftarrow a \cdot h_k^t + (1-a) \cdot w_k^t$;
return $w_k^{t+1} \leftarrow w_k^t$;

map the sensitivity score δ to the momentum coefficient a in a smooth and bounded manner:

$$a = a_0 \cdot (1 - \exp(-\gamma \cdot \delta)), \tag{17}$$

where a_0 is the initial momentum coefficient, and γ is a hyper-parameter controlling the adjustment rate. A larger γ makes a more sensitive to δ. This adaptive mechanism ensures that the historical model selectively incorporates robust global knowledge, enhancing its resilience to backdoor perturbations while maintaining up-to-date historical information for effective regularization. Finally, the participant uploads the fully regularized local model to the server. This model is also preserved and will be used to initialize the historical anchor at the start of the client's next active training round. Accordingly, the overall GBHINDER algorithm for a benign participant can be summarized in Algorithm 1.

5 Experimental Evaluation

In this section, we rigorously evaluate the efficacy of GBHINDER against SOTA attacks in FedGL through a series of comprehensive experiments. Our evaluation is conducted within a simulated FedGL environment using real-world datasets, ensuring practical relevance. We begin by benchmarking the performance of GBHINDER against leading SOTA defense mechanisms in the face of sophisticated backdoor attacks. Subsequently, we assess the robustness and adaptability of GBHINDER under diverse attack configurations and federated settings. Finally, we perform detailed ablation studies to dissect the framework and investigate the impact of its key components, thereby providing deeper insights into the factors that underpin its defensive capabilities.

Datasets and Models: Our primary evaluation focuses on the graph classification task. We validate GBHINDER on four real-world benchmarks: NCI1 [27], PROTEINS [2], DD [27], and AIDS [25]. Furthermore, to demonstrate the extensibility of GBHINDER, we conduct experiments on the node classification task on four datasets: CiteSeer [44], PubMed [44], Coauthor-CS [26], and Amazon-Photo [26]. Detailed statistics for all datasets are provided in Appendix A, and the results for the node classification task are presented in Appendix C. Consistent with prior works [17,35], we partition each dataset by randomly sampling 80% of the graphs for the training set and retaining the remaining 20% for the test set.

The Graph Isomorphism Network (GIN) [39], a powerful and widely-recognized architecture for graph representation learning, is employed as our default backbone model. To demonstrate the versatility and model-agnostic nature of GBHINDER, we also extend our evaluation to other prominent GNN architectures (shown in Appendix D), including the GCN [16], GAT [29], and GraphSAGE [14].

Attack and Defense Baselines: To evaluate the robustness of GBHINDER, we compare its performance against four SOTA backdoor attacks: Rand-GDBA [37], Motif-GDBA [52], Opt-GDBA [43], and NI-GDBA [17]. By default, we configure the proportion of malicious clients to 20%, with a trigger size of three nodes and a backdoor injection ratio of 50%. For benchmarking, GBHINDER is compared against five SOTA defense methods: FoolsGold [10], Flame [23], RLR [24], GNNCert [43], and FedTGE [30]. The first three are general-purpose FL defenses, while GNNCert and FedTGE are specifically designed for FedGL, representing the most advanced defenses in this domain. We adhere to the original implementations of these methods to ensure fair comparisons.

Implementation Details: The experiments are implemented within a FedGL framework using the FedAvg aggregation protocol, executed on a single machine equipped with an NVIDIA GeForce RTX 5080 GPU with 32 GB of memory. In each communication round, the server randomly selects five out of ten clients to contribute to the global model update. Moreover, we examine the influence of the total number of participants and the client selection fraction per round in Appendix B. The training dataset is evenly distributed among all clients unless otherwise specified to simulate non-IID scenarios. During local training, each selected client trains its model for ten epochs using the Adam optimizer with a learning rate of 0.01. For GBHINDER, we set the $\lambda_1 = 1$, $\lambda_2 = 1$, the upper bound of the momentum coefficient $a_0 = 0.9$, and the perturbation-based robustness metric scaling factor $\gamma = 10$. By default, to reduce computational overhead, we sample 50% of the edges when computing $\mathcal{L}_{\text{topo}}$ and 50% of the local data for the calculation of δ. A detailed analysis of the computational cost is provided in Appendix G. To ensure robustness of results, each experiment is repeated five times with different random seeds, and the average performance is reported.

Evaluation Metrics: Following established practices, we evaluate GBHINDER using two key metrics: ❶ the accuracy of the main classification task on clean samples (ACC), which measures the model's performance on its primary task; and ❷ the attack success rate (ASR), defined as the proportion of backdoored inputs misclassified to the adversary's target label. It is formally defined as:

$$\text{ASR} = \frac{\#\text{successful attacks}}{\#\text{total trials}}. \tag{18}$$

We report the mean ASR evaluated on the global model across all malicious clients at the final communication round. An ideal and robust FedGL framework means that it simultaneously achieves a high ACC while minimizing the ASR.

5.1 Comparison with SOTA Methods

In this section, we present a rigorous and comprehensive evaluation of GBHINDER 's performance in mitigating backdoor attacks within the FedGL, benchmarked against five leading defense methods. Table 1 summarizes the comparative results, with the baseline condition of no defense (i.e., employing stan-

Table 1: Performance comparison between GBHINDER and SOTA defense methods.

Datasets (ACC *w/o* attack)	Attack	No-Defense		FoolsGold		Flame		RLR		GNNCert		FedTGE		GBHINDER	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
NCI1 (ACC=80.42)	Rand-GDBA	75.72	28.66	78.39	12.37	77.26	13.13	78.17	9.54	70.58	2.03	75.97	39.58	80.56	6.73
	Motif-GDBA	77.36	54.65	78.33	53.81	78.98	44.99	79.06	50.63	71.61	22.74	76.32	37.13	79.35	5.26
	NI-GDBA	80.42	86.94	76.66	88.53	76.66	85.43	77.12	84.27	69.92	57.92	75.09	80.94	80.09	8.64
	Opt-GDBA	76.66	87.11	72.84	86.68	69.08	58.15	71.61	50.74	68.00	73.99	75.97	35.23	79.88	6.51
PROTEINS (ACC=75.95)	Rand-GDBA	66.47	29.89	65.41	26.54	67.11	24.09	70.32	33.78	61.75	5.95	73.28	30.26	76.53	7.54
	Motif-GDBA	67.30	63.78	69.76	22.02	63.39	57.48	67.58	57.48	59.77	27.60	72.27	42.07	74.01	6.72
	NI-GDBA	75.95	88.11	71.09	78.71	74.68	71.61	72.76	80.02	66.95	61.49	73.66	72.91	75.16	9.75
	Opt-GDBA	72.49	92.83	70.74	47.10	69.29	95.25	68.77	90.61	62.46	35.47	72.52	34.7	73.13	6.12
DD (ACC=71.46)	Rand-GDBA	68.82	30.19	69.49	30.68	70.09	28.35	67.86	40.63	62.49	8.47	67.34	10.31	70.13	5.24
	Motif-GDBA	69.86	40.63	68.67	47.92	68.87	40.24	65.87	40.24	62.25	29.08	64.26	21.7	69.72	6.31
	NI-GDBA	71.46	84.86	68.52	78.43	68.52	77.84	70.67	80.32	53.03	73.86	64.96	81.22	70.95	8.39
	Opt-GDBA	71.34	78.53	70.64	75.63	70.81	77.64	62.25	29.08	60.43	61.98	65.53	72.64	70.17	4.32
AIDS (ACC=99.74)	Rand-GDBA	99.56	24.56	99.76	11.56	99.76	4.16	97.12	18.45	95.24	14.05	91.96	11.98	99.72	8.34
	Motif-GDBA	99.88	38.45	98.07	9.05	99.76	7.84	98.45	35.84	92.26	8.90	93.97	5.79	99.17	6.05
	NI-GDBA	99.74	87.33	97.47	88.81	97.50	87.59	98.07	84.05	94.62	84.93	93.41	86.63	99.71	8.78
	Opt-GDBA	99.11	84.62	98.05	50.34	98.21	43.24	92.26	86.90	95.06	37.20	91.85	17.03	99.52	2.35

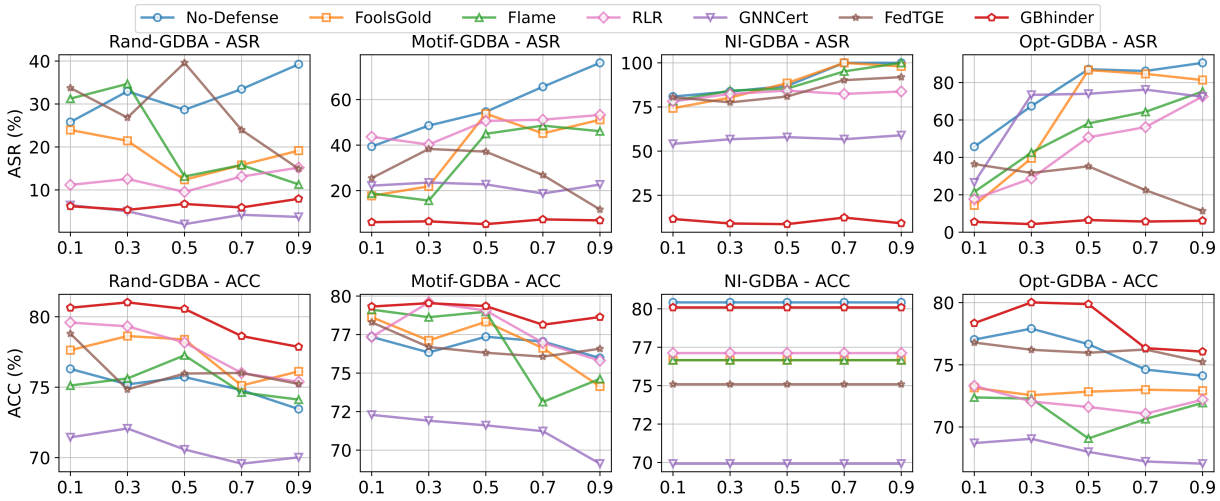

Figure 3: Performance of different defense methods under different poisoning ratios.

dard FedAvg aggregation) denoted as "No-defense". The best results are highlighted in Blue for clarity.

The experimental results demonstrate that GBHINDER consistently reduces the ASR to below 10% across all the datasets, significantly outperforming existing SOTA defense methods. From the attack perspective, backdoor attacks employing randomly generated triggers, such as Rand-GDBA, exhibit the lowest ASR, never exceeding 31% across the datasets. In contrast, attacks with selectively designed or optimized triggers achieve higher ASR, with NI-GDBA proving the most effective. Notably, NI-GDBA leverages "natura" triggers without modifying the training data, exploiting inherent patterns in the graph to embed backdoors, making it particularly stealthy and challenging to defend.

From the defense perspective, traditional defenses exhibit fragility in FedGL due to the topological structures and non-IID settings, particularly when confronting optimized triggers. GNNCert achieves partial success in mitigating backdoor effects, but at the cost of reduced ACC, as its deterministic graph partitioning sacrifices critical contextual information. Similarly, FedTGE's performance is inconsistent across graph-based tasks, particularly against NI-GDBA, where it fails almost entirely. This is likely because FedTGE relies on detecting malicious model updates, whereas NI-GDBA introduces no explicit modifications to the model parameters. Remarkably, GBHINDER demonstrates robust resistance to NI-GDBA's natural triggers. This resilience may stem from its Adaptive Momentum Information Update mechanism, which incorporates adversarial noise during updates to simulate backdoor patterns, indirectly enhancing the model's robustness to such noise-like triggers. In addition, we explored adaptive attacks in Appendix F.

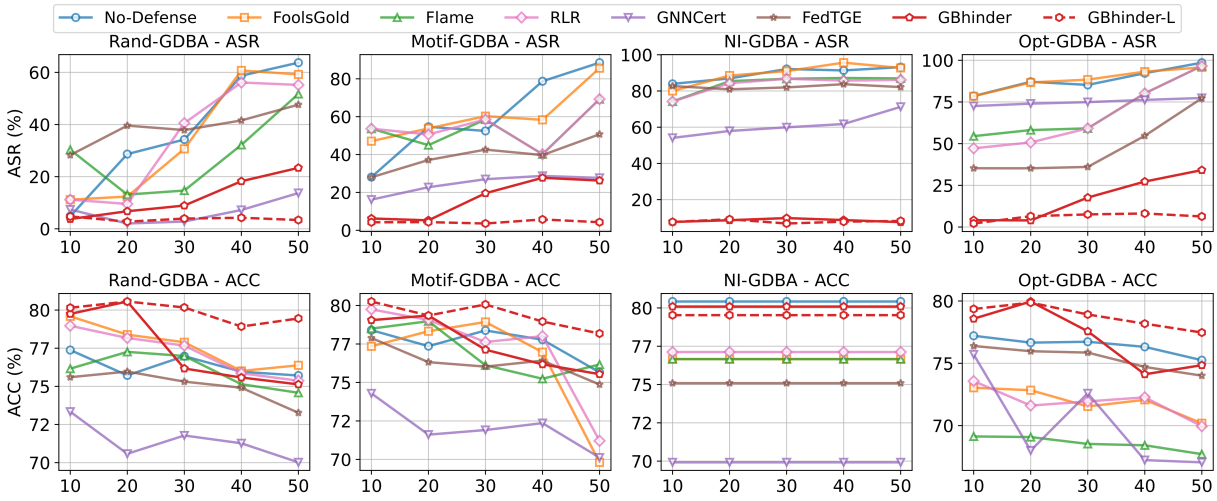

Figure 4: Performance of different defense methods under a varying ratio of malicious clients (%).

5.2 Robust to Different Attack Settings

Impact of Poisoning Ratio. In this section, we investigate the impact of the poisoning ratio on the performance of various defense methods within the FedGL framework. We define the poisoning ratio r as the fraction of each malicious client's training data injected with triggers. For the non-intrusive graph NI-GDBA, which does not embed triggers into clean samples during training, we define r as the fraction of training data used for learning perturbation-based triggers. The results of this comparative analysis are illustrated in Figure 3.

As expected, results show that in the absence of any defense, an increase in the poisoning ratio leads to a significant rise in ASR, indicating that a higher proportion of poisoned data exacerbates the contamination of the global model. Interestingly, this increased level of malicious influence also makes the attack more conspicuous, thereby enhancing the performance of certain server-side, detection-based defenses like FedTGE. Although a more heavily poisoned global model poses a greater challenge for GBHINDER's regularization process, this issue is effectively mitigated by our Adaptive Momentum Information Update mechanism. By dynamically adjusting the incorporation of global knowledge based on a robustness metric, this component becomes more conservative in updating the trusted local historical model when the global model is likely to be heavily compromised. Consequently, GBHINDER demonstrates robust and consistent defense effectiveness across the entire range of tested poisoning ratios.

Impact of Malicious Participation Rates. In this section, we explore the robustness of different defense mechanisms across different malicious participation rates. Specifically, we define the malicious participation rate M_a as the proportion of malicious clients relative to the total number of clients, varying M_a from 10% to 50%. The experimental results, presented in Figure 4, compare GBHINDER against SOTA defenses, with the "No-Defense" curve representing the baseline scenario using Fedavg. Additionally, we report the average ASR of the local models of benign clients in the final communication round (before aggregation), providing insight into the self-protection capabilities of benign clients.

As observed from the results, an increasing malicious client ratio M_a generally leads to a degradation in ACC and a significant increase in the ASR for the global model. This trend poses a fundamental challenge, as standard aggregation algorithms are essentially a weighted average of all client updates; a higher M_a inherently increases the aggregate weight of malicious parameters being introduced into the global model. In contrast, GBHINDER does not rely on server-side filtering but requires only benign clients to adhere to the proposed protocol, effectively mitigating the impact of malicious gradients by leveraging the gradients of benign models to purify the global model. However, when the proportion of malicious participants is too high, the defense performance does degrade, because we cannot prevent a large number of malicious gradients from continuously participating in the global aggregation. Fortunately, the consistently low ASR of the pre-aggregation local models confirms that GBHINDER effectively enables each benign client to achieve self-preservation, safeguarding its own model's integrity. Additionally, we explore the impact of trigger type and size on GBHINDER in the Appendix E.

5.3 Applicability Analysis

Non-IID Data Distributions. In FedGL scenarios, diverse data distributions represent a critical and realistic setting. Following established methodologies [15, 48], we adopt the Dirichlet distribution Dir(α) to simulate varying degrees of data heterogeneity, where smaller values of α correspond to higher levels of heterogeneity. Specifically, we configure α to take values of 0.5, 1, 5, 10, and 1000 across three benchmark

Table 2: Impact of non-IID data distributions.

Datasets	Setting	No-Defense		GBHINDER	
		ACC	ASR	ACC	ASR
NCI1	0.5	68.32	72.56	73.21 (↑4.89)	9.13 (↓63.43)
	1	75.26	79.63	78.56 (↑3.30)	7.32 (↓72.31)
	5	76.05	84.17	79.33 (↑3.28)	6.98 (↓77.19)
	10	77.12	87.21	80.12 (↑3.00)	6.01 (↓81.20)
	1000	76.71	87.52	79.96 (↑3.25)	6.44 (↓81.08)
PROTEINS	0.5	62.86	85.74	64.12 (↑1.26)	7.93 (↓77.81)
	1	70.93	90.17	72.05 (↑1.12)	8.06 (↓82.11)
	5	71.01	91.72	73.41 (↑2.40)	7.79 (↓83.93)
	10	71.98	93.56	74.92 (↑2.94)	7.71 (↓85.85)
	1000	72.49	92.83	75.05 (↑2.56)	7.68 (↓85.15)
DD	0.5	57.03	77.63	62.52 (↑5.49)	6.26 (↓71.37)
	1	67.78	76.73	69.71 (↑1.93)	5.98 (↓70.75)
	5	68.46	80.12	70.04 (↑1.58)	6.01 (↓74.11)
	10	68.82	80.64	71.19 (↑2.37)	5.16 (↓75.48)
	1000	71.02	78.53	70.78 (↓0.24)	5.29 (↓73.24)
AIDS	0.5	92.22	80.42	93.78 (↑1.56)	11.15 (↓69.27)
	1	96.12	82.77	98.05 (↑1.93)	9.78 (↓72.99)
	5	99.27	84.56	99.61 (↑0.34)	9.65 (↓74.91)
	10	98.58	86.39	99.45 (↑0.87)	7.98 (↓78.41)
	1000	99.11	84.62	99.58 (↑0.47)	8.16 (↓76.46)

datasets, with $\alpha = 1000$ representing the IID scenario.

As presented in Table 2, GBHINDER exhibits remarkable robustness to data heterogeneity. This resilience stems from the fundamental design of our client-centric defense paradigm. Many server-side defenses struggle in non-IID environments because the natural statistical drift between benign clients' updates can be mistaken for malicious activity, blurring the lines for anomaly detection. GBHINDER, however, circumvents this issue entirely. Its defense mechanism does not compare a client's update to those of other clients. Instead, it regularizes the incoming global model against the client's own trusted historical knowledge.

5.4 Parameter Sensitivity and Ablation

Role of Different Loss Terms. As delineated in Eq. 13, the deep-shallow alignment loss, $\mathcal{L}_{\text{align}}$, constrains the propagation of backdoor features in deeper layers by aligning the global model's deep-layer representations with the shallow-layer features of the local historical model. Concurrently, the topological consistency loss, $\mathcal{L}_{\text{topo}}$, mitigates anomalous topological patterns introduced by backdoor attacks by enforcing consistency in feature representations among neighboring nodes. The cross-entropy loss, $\mathcal{L}_{\text{ce}}$, serves as the primary optimization objective for the local task. In this section, we systematically investigate the practical contributions of each loss component to the overall efficacy of GBHINDER.

Our experiments presented in Table 3 reveal that the cross-entropy loss is fundamental to the local training task, driving the model to learn discriminative feature representations for distinguishing between classes. In the absence of $\mathcal{L}_{\text{ce}}$, direct

Table 3: Ablation of different loss terms.

$\mathcal{L}_{\text{ce}}$	$\mathcal{L}_{\text{align}}$	$\mathcal{L}_{\text{topo}}$	ACC	ASR
✓	✗	✗	76.66 (↓3.21)	87.11 (↑80.60)
✗	✓	✗	68.42 (↓11.45)	35.74 (↑29.23)
✗	✗	✓	65.39 (↓14.48)	48.75 (↑42.24)
✗	✓	✓	62.52 (↓17.35)	11.73 (↑5.22)
✓	✗	✓	75.94 (↓3.93)	40.53 (↑34.02)
✓	✓	✗	77.53 (↓2.34)	28.79 (↑22.28)
✓	✓	✓	**79.87**	**6.51**

regularization of the model using $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{topo}}$ results in a noticeable degradation of ACC, with $\mathcal{L}_{\text{topo}}$ exhibiting a more pronounced impact due to its focus on enforcing topological consistency, which may overly constrain the model's flexibility in capturing diverse graph structures. Both $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{topo}}$ play critical roles in suppressing backdoor attacks, albeit through complementary mechanisms. $\mathcal{L}_{\text{align}}$ operates from a feature-centric perspective, effectively mitigating backdoor patterns that rely on manipulating feature representations to embed malicious triggers. In contrast, $\mathcal{L}_{\text{topo}}$ addresses backdoor attacks from a topological perspective, countering attempts to implant triggers through alterations in graph structure. Our empirical results demonstrate that the synergistic interplay between these two loss components yields superior defense performance compared to their contributions.

Component Contributions. The GBHINDER framework is fundamentally composed of two core modules: the Historical Channel Attention Regularization (HCAR) and the Adaptive Momentum Information Update (AMIU). To empirically validate the specific contribution of each, we conduct a comprehensive ablation study by designing and evaluating several framework variants. These include: (i) the full GBHINDER; (ii) a variant that removes only the AMIU module ("*w/o* AMIU"); (iii) a variant that removes the HCAR module ("*w/o* HCAR"); and (iv) a baseline that removes both components ("*w/o* both"), which effectively degenerates to the standard FedAvg protocol. A crucial design consideration arises for the "*w/o* HCAR" variant: since AMIU's primary function is to update the historical model for HCAR, removing HCAR would render AMIU's role meaningless. Therefore, to construct a meaningful ablation, we replace the HCAR mechanism in this variant with a simpler regularization that directly constrains the model's intermediate layer features. We evaluate the performance of these variants under varying degrees of data heterogeneity. Experimental results shown in Table 4 demonstrate that HCAR plays a pivotal role in backdoor mitigation.

Notably, the "*w/o* HCAR" variant, which relies on the AMIU-updated model for direct feature regularization, still provides a limited degree of defense. This observation aligns with findings from the previous section, suggesting that AMIU indirectly enhances model robustness to backdoors by dynam-

Table 4: Ablation of different components of GBHINDER.

Setting	*w/o* Both		*w/o* HCAR		*w/o* AMIU		GBHINDER	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
$\alpha = 0.5$	68.32 (↓4.89)	72.56 (↑63.43)	68.79 (↓4.42)	51.37 (↑42.24)	62.50 (↓10.71)	41.72 (↑32.59)	73.21	9.13
$\alpha = 1$	75.26 (↓3.30)	79.63 (↑72.31)	72.57 (↓5.99)	46.71 (↑39.39)	69.75 (↓8.81)	25.71 (↑18.39)	78.56	7.32
$\alpha = 5$	76.05 (↓3.28)	84.17 (↑77.19)	75.86 (↓3.47)	42.24 (↑35.26)	74.61 (↓4.72)	30.27 (↑23.29)	79.33	6.98
$\alpha = 10$	77.12 (↓3.00)	87.21 (↑81.20)	78.28 (↓1.84)	29.93 (↑23.92)	77.76 (↓2.36)	14.06 (↑8.05)	80.12	6.01
$\alpha = 1000$	76.71 (↓3.25)	87.52 (↑81.08)	81.09 (↑1.13)	33.19 (↑26.75)	78.15 (↓1.81)	12.72 (↑6.28)	79.96	6.44

Table 5: Impact of layer alignment strategies.

	ACC	ASR
No-Defense	76.66	87.11
$< l, l >$	80.53 (↑3.87)	17.39 (↓69.72)
$< l, l+1 >$	79.88 (↑3.22)	6.51 (↓80.60)
$< l, l+2 >$	72.34 (↓4.32)	15.24 (↓71.87)
$< l, l+3 >$	57.56 (↓19.10)	47.71 (↓39.40)

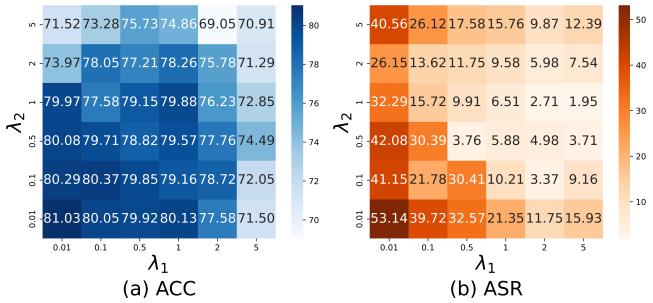

Figure 5: Balance of λ_1 and λ_2.

ically updating historical knowledge, even in the absence of tailored regularization. Furthermore, our analysis reveals the critical importance of AMIU in realistic, non-IID settings. While the impact of removing AMIU is marginal under a balanced data distribution, its function becomes indispensable as data heterogeneity intensifies. Severe heterogeneity conditions cause local anchors to become biased, and constraining the global model solely with such anchors leads to a significant decline in ACC. AMIU mitigates this issue by enabling clients to selectively incorporate robust global knowledge, dynamically updating and refining their historical anchors. This adaptive mechanism effectively alleviates the bias in local anchors, ensuring that GBHINDER maintains robust backdoor mitigation and high ACC across diverse data distributions.

Layer Alignment Strategies. In this section, we investigate the impact of different layer alignment strategies within GBHINDER on its effectiveness in defense. Specifically, the default GBHINDER employs a cross-layer alignment strategy, where features from the l-th layer of the historical model constrain the $(l+1)$-th layer of the global model (denoted as $\langle l, l+1 \rangle$). To explore the efficacy of alternative alignment configurations, we evaluate four combinations: $\langle l, l \rangle$, $\langle l, l+1 \rangle$, $\langle l, l+2 \rangle$, and $\langle l, l+3 \rangle$. Experiments are conducted using the GIN model on the NCI1 dataset, with performance assessed against the Opt-GDBA backdoor attack. The results, summarized in Table 5, indicate that both same-layer alignment ($\langle l, l \rangle$) and cross-layer alignments within a span of two layers ($\langle l, l+1 \rangle$ and $\langle l, l+2 \rangle$) are effective in mitigating backdoor effects, with $\langle l, l+1 \rangle$ achieving the optimal balance of backdoor suppression and main-task accuracy.

This finding aligns with our hypothesis that backdoor triggers, initially manifesting as localized patterns, undergo amplification and consolidation during the GNN's message-passing process, forming stable, malicious representations in deeper layers that dominate classification outcomes. The core principle of GBHINDER is to leverage clean, foundational shallow-layer features from the historical model (at layer l) to proactively regulate and constrain the global model's formation of complex deep-layer features (at layer $l+1$), thereby disrupting the "amplification chain" of backdoor signals. In contrast, same-layer alignment ($\langle l, l \rangle$) functions as a passive constraint, attempting to correct the l-th layer's representations after they have already been influenced by potentially compromised information from the $(l-1)$-th layer. This approach fails to address the fundamental issue of malicious signal propagation from earlier layers. Conversely, large-span alignments, such as $\langle l, l+2 \rangle$ and $\langle l, l+3 \rangle$, introduce training instability and negatively impact ACC. Forcing alignment between features from layers that are too far apart is impractical and conflicts with the objectives of graph learning, as it disrupts the natural hierarchical feature extraction process of GNNs.

Balance of λ_1 and λ_2. In this section, we conduct a sensitivity analysis to investigate the influence of the hyperparameters λ_1 and λ_2, which control the weights of the alignment loss $\mathcal{L}_{\text{align}}$ and the topological loss $\mathcal{L}_{\text{topo}}$, respectively. We systematically vary both λ_1 and λ_2 across the range $\{0.01, 0.1, 0.5, 1.0, 2, 5\}$ and evaluate the performance of GBHINDER on the NCI1 dataset against the Opt-GDBA attack. Figure 5 presents heatmaps that visualize the ACC and ASR across the grid of hyperparameter combinations. The results reveal a clear trade-off. When both λ_1 and λ_2 are set to low values (e.g., 0.01), the regularization strength is insufficient. The weak constraints imposed by $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{topo}}$ fail to effectively suppress the backdoor, leading to a high ASR, though the ACC remains stable. Conversely, excessively high values

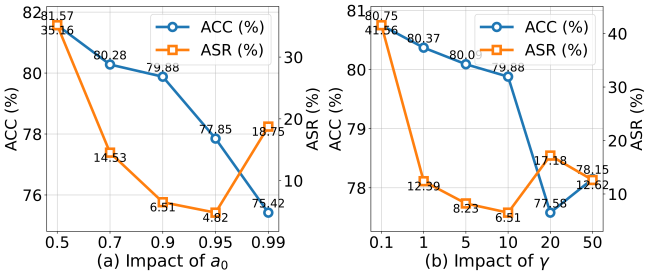

Figure 6: Impact of hyperparameters a_0 and γ.

for either hyperparameter are detrimental to the main task. An overly large λ_1 over-emphasizes the deep-shallow alignment, potentially forcing the model to disregard useful, benign information from the global model and thus impairing its generalization on clean data. Similarly, an exceedingly high λ_2 imposes an overly strict topological smoothness, which can erase natural and informative structural differences within the graphs. The optimal performance is achieved with a balanced configuration, where the parameters are large enough to mitigate the attack but not so large as to degrade ACC. Overall, our analysis indicates that an effective setup generally requires a relatively strong weight for $\mathcal{L}_{\text{align}}$ to decisively counter the deep-layer backdoor representations, complemented by a smaller weight for $\mathcal{L}_{\text{topo}}$, which serves as an effective auxiliary regularizer.

Impact of a_0 and γ. In this section, we analyze the influence of two key hyperparameters within the Adaptive Momentum Information Update module: a_0 and γ. The parameter a_0 defines the upper bound of the momentum coefficient, dictating the maximum proportion of the historical model's parameters that are retained during an update. We evaluate its impact by varying a_0 within the set $\{0.5, 0.7, 0.9, 0.95, 0.99\}$. As shown in Figure 6 (a), a low value for a_0 results in a higher ASR. This occurs because a low a_0 allows the historical model to absorb a significant proportion of global model knowledge, even when substantial differences exist between the global and historical models. Since a single training round cannot fully purify the global model of backdoor influences, this continuous incorporation of potentially malicious knowledge diminishes the effectiveness of the historical anchor as a regularization baseline. While this continuous integration of global knowledge helps maintain a high ACC, it compromises security. Conversely, an excessively high a_0 (e.g., 0.99) is also suboptimal, as it hinders the historical anchor from keeping pace with the benign evolution of the global model, causing it to become stale and weakening the overall performance.

The hyperparameter γ governs the responsiveness of the momentum coefficient a to the perturbation sensitivity score δ. A high γ makes the system highly sensitive to detected threats, while a low γ results in smoother, less reactive adjustments. We test its effect by varying γ across the set $\{0.1, 1.0, 5, 10, 20, 50\}$. The results, presented in Figure 6 (b), illustrate a critical balance. When γ is too small (e.g., 0.1), the system largely ignores the threat signal from δ, causing the momentum coefficient to remain low. This leads to the historical anchor being aggressively overwritten and polluted by malicious updates, thereby increasing the ASR. In contrast, a very large γ over-amplifies the sensitivity, causing the system to over-react even to benign model fluctuations. This can trigger the momentum a to jump to its maximum value a_0 unnecessarily, leading to the local anchor chronically rejecting new information from the global model. This isolation makes the anchor stale and ultimately degrades defense performance. Therefore, a moderate value for γ is essential to ensure the system is responsive to genuine threats while remaining stable against normal variations in the training process.

6 Summary and Future Work

In this paper, we introduced GBHINDER, a novel trusted-server-free framework for defending against backdoor attacks in FedGL. GBHINDER is built upon a virtuous defense cycle, driven by two synergistic components: Historical Channel Attention Regularization, which leverages a historical anchor to purify the global model, and Adaptive Momentum Information Update, which strengthens this anchor over time by selectively incorporating robust global knowledge. By operating without reliance on a trusted server, GBHINDER offers a practical and deployable solution for real-world federated systems. Our extensive experiments demonstrate that GBHINDER successfully thwarts backdoor attacks, reducing ASR to below 10% while maintaining high ACC and outperforming SOTA methods. Its robustness and effectiveness are further validated through comprehensive evaluations across diverse attack, federated learning settings, and non-IID data distributions, supported by detailed ablation studies, parameter sensitivity analyses, and overhead measurements. Nonetheless, we acknowledge a trade-off, as detailed in our overhead analysis in Appendix G, the robust protection offered by GBHINDER comes with moderate computational and storage costs. A key direction for future work, therefore, is to explore more lightweight defense architectures.

Acknowledgments

The authors would like to thank reviewers for the time and effort they have kindly made in this paper. This work was supported in part by the National Natural Science Foundation of China under Grant (NSFC62272222, NSFC-62272215), the Jiangsu Province Outstanding Youth Fund (No. BK20230080), the Fundamental Research Funds for the Central Universities (No. 2024300401), the Natural Science Foundation of Jiangsu Province Grant (No. BK20251911), and the China Postdoctoral Science Foundation (No. 2025T180438).

Ethical Considerations

Our research is dedicated to enhancing the security and trustworthiness of decentralized machine learning systems. The primary focus of GBHINDER is purely defensive, designed to protect Federated Graph Learning models from malicious backdoor attacks. The ethical implications of this work are positive, as it aims to secure a technology often used in safety-critical applications like finance and healthcare. We exclusively utilized public, open-source benchmark datasets, ensuring that no private or personally identifiable information was handled or compromised during our research. By highlighting the persistent threat of backdoors and providing a practical defense, our study intends to raise awareness within the community and contribute to the collective effort of building more secure, reliable, and trustworthy AI systems.

We considered the following stakeholders and how they may be affected by the research. **Deployers and Operators.** Organizations (e.g., hospitals, financial institutions) deploying FedGL benefit from GBHINDER by being able to protect their global models without relying on a centralized trusted server, which is often legally or practically infeasible to establish. **Adversaries.** Malicious actors attempting to compromise FedGL systems are negatively impacted as GBHINDER increases the cost and complexity of launching successful backdoor attacks.

Potential Risks. We acknowledge that research into defenses can inherently inform adversaries. Specifically, by detailing the mechanics of GBHINDER, sophisticated attackers might attempt to design adaptive attacks specifically engineered to bypass our regularization. However, we reasoned that the benefits of publication outweigh these risks. Security through obscurity is fragile, and only by transparently publishing the defense mechanics can the community analyze its limits and develop even stronger iterations.

Open Science

In alignment with the principles of open and reproducible science, we are committed to making our research fully accessible. We provide a comprehensive artifact package that includes the complete source code for GBHINDER, configurations required to replicate all experiments presented in this paper, including baseline comparisons and ablation studies. All materials are available on Zenodo (https://doi.org/10.5281/zenodo.17898578).

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[2] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[3] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

[4] Cen Chen, Tiandi Ye, Li Wang, and Ming Gao. Learning to generalize in heterogeneous federated networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 159–168, 2022.

[5] Chuan Chen, Ziyue Xu, Weibo Hu, Zibin Zheng, and Jie Zhang. Fedgl: Federated graph learning framework with global self-supervision. *Information Sciences*, 657:119976, 2024.

[6] Lvjun Chen, Di Xiao, Xiangli Xiao, and Yushu Zhang. Secure and efficient federated learning via novel authenticable multi-party computation and compressed sensing. *IEEE Transactions on Information Forensics and Security*, 2024.

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

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

[9] 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*, pages 417–426, 2019.

[10] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020.

[11] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[12] Michelle Goddard. The eu general data protection regulation (gdpr): European regulation that has a global

impact. *International Journal of Market Research*, 59(6):703–705, 2017.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[15] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[17] Ken Li, Bin Shi, Jiazhe Wei, and Bo Dong. Ni-gdba: Non-intrusive distributed backdoor attack based on adaptive perturbation on federated graph learning. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pages 852–862, New York, NY, USA, April 2025. Association for Computing Machinery.

[18] Songze Li and Yanbo Dai. {BackdoorIndicator}: Leveraging {OOD} data for proactive backdoor detection in federated learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4193–4210, 2024.

[19] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[20] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

[21] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE transactions on neural networks and learning systems*, 2024.

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, April 2017.

[23] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX security symposium (USENIX Security 22)*, pages 1415–1432, 2022.

[24] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9268–9276, 2021.

[25] Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[26] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[27] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[28] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

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

[30] Guancheng Wan, Zitong Shi, Wenke Huang, Guibin Zhang, Dacheng Tao, and Mang Ye. Energy-based backdoor defense against federated graph learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE international conference on data mining (ICDM)*, pages 598–607. IEEE, 2019.

[32] Zhen Wang, Weirui Kuang, Yuexiang Xie, Liuyi Yao, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4110–4120, 2022.

[33] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.

[34] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[35] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *30th USENIX security symposium (USENIX Security 21)*, pages 1523–1540, 2021.

[36] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

[37] Jing Xu, Rui Wang, Stefanos Koffas, Kaitai Liang, and Stjepan Picek. More is better (mostly): On the backdoor attacks in federated graph neural networks. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 684–698, 2022.

[38] Jing Xu, Minhui Xue, and Stjepan Picek. Explainability-based backdoor attacks against graph neural networks. In *Proceedings of the 3rd ACM workshop on wireless security and machine learning*, pages 31–36, 2021.

[39] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[41] Han Yang, Binghui Wang, Jinyuan Jia, et al. Gnncert: Deterministic certification of graph neural networks against adversarial perturbations. In *The Twelfth International Conference on Learning Representations*, 2024.

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[43] Yuxin Yang, Qiang Li, Jinyuan Jia, Yuan Hong, and Binghui Wang. Distributed backdoor attacks on federated graph learning and certified defenses. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2829–2843, 2024.

[44] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[45] Xiaoyan Yin, Wanyu Lin, Kexin Sun, Chun Wei, and Yanjiao Chen. A 2 s 2-gnn: Rigging gnn-based social status by adversarial attacks in signed social networks. *IEEE Transactions on Information Forensics and Security*, 18:206–220, 2022.

[46] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[47] Hao Yu, Chuan Ma, Xinhang Wan, Jun Wang, Tao Xiang, Meng Shen, and Xinwang Liu. Dshield: Defending against backdoor attacks on graph neural networks via discrepancy learning. In *NDSS*, 2025.

[48] Jiale Zhang, Chengcheng Zhu, Xiaobing Sun, Chunpeng Ge, Bing Chen, Willy Susilo, and Shui Yu. *FLPurifier* : Backdoor Defense in Federated Learning via Decoupled Contrastive Training. *IEEE Transactions on Information Forensics and Security*, 19:4752–4766, 2024.

[49] Kaiyuan Zhang, Siyuan Cheng, Guangyu Shen, Guanhong Tao, Shengwei An, Anuran Makur, Shiqing Ma, and Xiangyu Zhang. Exploring the orthogonality and linearity of backdoor attacks. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2105–2123, 2024.

[50] 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*, pages 15–26, 2021.

[51] 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*, pages 4386–4397, 2024.

[52] Haibin Zheng, Haiyang Xiong, Jinyin Chen, Haonan Ma, and Guohan Huang. Motif-backdoor: Rethinking the backdoor attack on graph neural networks via motifs. *IEEE Transactions on Computational Social Systems*, 11(2):2479–2493, 2023.

A Details on Datasets

In this work, we focus on both graph and node classification tasks. We select four widely used datasets for each task. The basic statistics for each dataset are shown in Table 6 and 7. For graph classification, the datasets include: NCI1 [27], PROTEINS_full [2], DD [27], and AIDS [25]. For node classification, the datasets include: CiteSeer [44], PubMed [44], Coauthor-CS [26], and Amazon-Photo [26].

Table 6: **Dataset statistics for graph-level tasks.**

Datasets	NCI1	PROTEINS	DD	AIDS
#Graphs	4,110	1,113	1,178	2,000
Avg. #Nodes	29.87	39.06	284.32	15.69
Avg. #Edges	32.3	715.66	72.82	32.3
#Classes	2	2	2	2
#Target label	1	1	0	1

Table 7: **Dataset statistics for node-level tasks.**

Datasets	CiteSeer	Pubmed	Coauthor-CS	Amazon-Photo
#Nodes	3,327	19,717	18,333	7,650
#Edges	4,732	44,338	81,894	119,081
#Feature	3,703	500	6,805	745
#Classes	6	3	15	8
#Target label	1	1	1	0

B Impact of Different Server Settings

In this section, we investigate the impact of various federated settings on GBHINDER. We first evaluate the performance impact of deploying GBHINDER in a benign environment, i.e., without attacks. As shown in Table 8, the results demonstrate that the main-task accuracy remains on par with standard FedAvg. Subsequently, we examine the influence of the total number of participants and the client selection fraction per round. Following established conventions in FedGL [43], where scaling to thousands of clients remains an open challenge, we configure the total number of clients at 20 and 40, with random selection ratios of 20%, 50%, and 100%. The results in Table 9 indicate that increasing the client population and participation rate further elevates the difficulty of the attack due to the stronger dilution effect of benign updates. Crucially, GBHINDER proves consistently effective across these varying configurations.

C Applicability to Node Classification Tasks

To validate the task-agnostic nature of GBHINDER, we extend the evaluation to the node classification task. This assessment is conducted on four prominent datasets against a diverse set of attack methodologies. For this experimental setting, we make two key adjustments to our baseline comparisons. First, GNNCert is excluded from the evaluation, as its certified defense mechanism is designed specifically for graph-level predictions. Second, to ensure a rigorous evaluation against a more relevant and challenging threat model, we incorporate GTA-GDBA and UGBA-GDBA, which are the FedGL adaptations of the potent node-level backdoor attacks, GTA [35] and UGBA [8].

Table 8: Performance of GBHINDER without attacks.

	NCI1	PROTEINS	DD	AIDS
Fedavg	80.42	75.95	71.46	99.74
GBHINDER	79.87	73.26	72.08	99.02

Table 9: Performance of GBHINDER with different total number of participants and participation ratios.

Num	Participation Rate (ACC *w/o* attack)	Opt-GDBA		GBHINDER	
		ACC	ASR	ACC	ASR
20	20% (75.18)	69.82	91.17	74.32	13.63
	50% (78.34)	71.74	86.53	76.93	6.98
	100% (78.58)	74.93	69.71	78.12	9.72
40	20% (77.37)	73.78	89.72	77.57	10.32
	50% (77.73)	75.71	72.97	76.59	12.59
	100% (78.83)	75.18	59.72	78.92	8.17

The results are summarized in Table 10. The findings reveal that defenses from general FL are largely unreliable against these specialized attacks. While they may exhibit sporadic effectiveness on certain datasets or against specific attacks, their performance is inconsistent, underscoring their lack of robustness and generalizability when confronted with the unique structural and feature-based vulnerabilities in graph data. As anticipated, FedTGE, with its energy-based model being more attuned to node-level feature distributions, demonstrates improved efficacy here compared to its performance in graph classification. Crucially, while GBHINDER exhibits a marginal performance degradation relative to the graph classification task, it consistently delivers strong and satisfactory defensive outcomes across all tested scenarios. This result substantiates our claim that GBHINDER's defense mechanism is fundamentally task-agnostic.

D Impact of Different Model Architectures

To validate the generalizability of GBHINDER, we conducted a comprehensive set of experiments across four widely-used GNN architectures: GCN, GIN, GraphSAGE, and GAT, while keeping all other hyperparameters consistent with our default configuration. We first established a baseline by confirming

Table 10: Performance comparison between GBHINDER and SOTA defense methods on node classification tasks.

Datasets (ACC w/o attack)	Attack	No-Defense		FoolsGold		Flame		RLR		FedTGE		GBHINDER	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CiteSeer (ACC=76.05)	**Rand-GDBA**	74.36	94.12	73.54	95.26	72.25	33.46	72.17	46.54	73.97	8.58	75.08	7.53
	GTA-GDBA	71.79	96.51	70.32	78.62	69.76	85.51	69.06	61.63	72.32	13.13	72.18	11.58
	UGBA-GDBA	75.26	86.71	67.98	6.78	73.76	26.86	74.12	84.27	74.09	18.14	74.86	5.67
Pubmed (ACC=88.89)	**Rand-GDBA**	84.87	74.87	83.23	63.71	84.35	41.48	80.59	80.76	85.75	22.26	86.53	13.74
	GTA-GDBA	83.58	92.08	82.85	70.01	85.45	59.49	81.58	82.48	87.34	12.07	87.19	8.92
	UGBA-GDBA	85.58	87.51	79.32	8.57	84.05	1.87	82.76	80.02	83.66	32.91	86.76	9.51
Coauthor-CS (ACC=91.25)	**Rand-GDBA**	86.89	93.95	85.72	89.41	86.01	32.91	84.07	84..58	85.63	27.95	88.52	15.24
	GTA-GDBA	91.39	71.29	89.59	52.56	88.92	21.50	91.35	24..54	90.53	12.18	91.13	10.54
	UGBA-GDBA	90.16	83.56	87.95	26.57	89.76	85.51	81.67	39.32	88.96	8.22	90.7	6.32
Amazon-Photo (ACC=84.08)	**Rand-GDBA**	84.25	95.33	84.82	94.98	81.08	31.08	78.26	92.44	80.81	2.08	83.18	7.24
	GTA-GDBA	81.84	72.93	77.18	57.73	78.68	8.82	76.87	48.24	80.26	3.17	82.05	2.11
	UGBA-GDBA	83.54	95.43	72.80	5.42	75.01	19.72	80.57	80.32	82.96	15.23	83.92	8.39

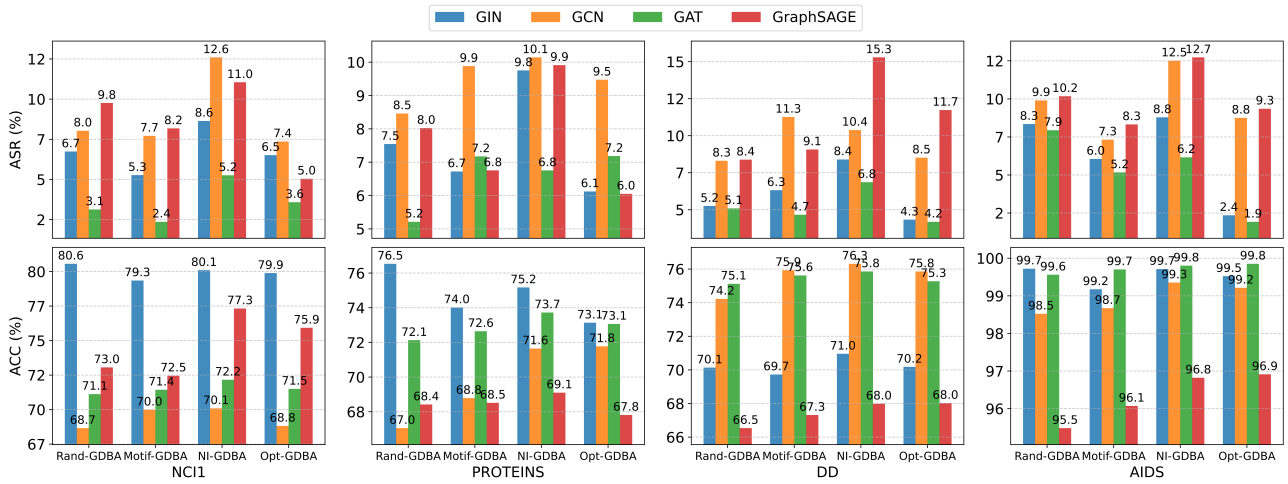

Figure 7: Performance of GBHINDER under different model architectures.

the efficacy of the backdoor attacks across these diverse architectures (as detailed in Table 11). Subsequently, we evaluated the defensive performance of GBHINDER when integrated with each of these GNN backbones.

The results, depicted in Figure 7, demonstrate that GBHINDER consistently and effectively mitigates backdoor attacks regardless of the underlying GNN architecture. This architectural agnosticism can be attributed to the core mechanism of GBHINDER. Our regularization losses, $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{topo}}$, operate on intermediate layer activations $\mathcal{F}^l$ and node representations of the pre-pooling layer $\mathbf{z}^{(i)}$, which are fundamental and ubiquitous components across virtually all GNN models. Consequently, whether the hierarchical representations are generated via GCN's mean aggregation, GraphSAGE's sampling-based aggregation, or GIN's expressive isomorphism-aware aggregation, they are all susceptible to the constraints imposed by the GBHINDER framework.

Notably, the defensive performance was particularly pronounced when using GIN and GAT as backbones. The high structural sensitivity of GIN, which makes it a powerful expressive model, also implies that an effective backdoor must induce more significant perturbations to its internal activation patterns. These larger deviations make the malicious behavior more conspicuous and thus easier for our regularization terms to detect and rectify. Similarly, for GAT, which is inherently based on attention mechanisms, a successful backdoor must manipulate its internal node-level attention weights. This creates a scenario where our channel-wise attention regularization $\mathcal{L}_{\text{align}}$ acts in concert with GAT's own mechanisms, resulting in a potent synergistic defensive effect.

E Impact of Trigger Types and Sizes

Backdoor Trigger Types. In this part, we investigate the performance of GBHINDER when confronted with different backdoor triggers, including those generated by EXPBA [38], GTA [35], UGBA [8], and DPBA [51]. These advanced at-

Table 11: The effectiveness of backdoor attacks under different model architectures.

Datasets	Attack	GIN		GCN		GAT		GraphSAGE	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
NCI1	**Rand-GDBA**	75.72	28.66	67.15	41.16	65.32	48.06	71.53	34.81
	Motif-GDBA	77.36	54.65	68.97	70.93	69.98	58.48	68.46	48.63
	NI-GDBA	80.42	86.94	71.04	88.28	72.14	81.68	77.49	85.56
	Opt-GDBA	76.66	87.11	69.71	81.76	70.58	87.98	72.42	96.68
PROTEINS	**Rand-GDBA**	66.47	29.89	65.02	43.56	67.05	30.69	65.47	24.75
	Motif-GDBA	67.30	63.78	69.97	76.31	71.52	68.72	67.94	59.72
	NI-GDBA	75.95	88.11	73.54	85.31	73.99	84.92	69.51	87.09
	Opt-GDBA	72.49	92.83	70.13	90.12	72.78	95.05	67.70	96.08
DD	**Rand-GDBA**	68.82	30.19	71.94	54.46	67.79	61.51	66.94	49.11
	Motif-GDBA	69.86	40.63	74.56	65.73	72.57	57.26	66.53	70.57
	NI-GDBA	71.46	84.86	77.12	92.86	76.69	90.07	68.22	93.27
	Opt-GDBA	71.34	78.53	72.39	83.58	74.06	82.86	67.78	87.66
AIDS	**Rand-GDBA**	99.56	24.56	98.35	20.73	99.01	47.06	95.25	46.08
	Motif-GDBA	99.88	38.45	99.14	35.24	98.93	46.72	95.87	57.68
	NI-GDBA	99.74	87.33	99.75	86.18	99.61	91.57	97.25	92.54
	Opt-GDBA	99.11	84.62	98.92	90.05	97.93	79.52	96.95	90.16

Table 12: Performance of GBHINDER with different trigger types.

Attack	No-defense		GBHINDER	
	ACC	ASR	ACC	ASR
EXPBA	72.66	78.19	78.54 (↑5.88)	6.11 (↓72.08)
GTA	74.53	92.67	77.23 (↑2.70)	12.39 (↓80.28)
UGBA	78.98	89.12	80.18 (↑1.20)	2.65 (↓86.47)
DPBA	78.57	93.54	78.78 (↑0.21)	9.50 (↓84.04)

tacks employ distinct methodologies to select trigger injection sites and optimize trigger structures. The results presented in Table 12 demonstrate that GBHINDER achieves robust performance across all trigger types. It is noteworthy that certain advanced attacks, such as UGBA and DPBA, meticulously attempt to integrate triggers seamlessly into the feature or structural distributions of benign data. Such camouflage can partially diminish the effectiveness of $\mathcal{L}_{\text{topo}}$. However, these attacks fundamentally rely on constructing a malicious neural pathway within the model to manipulate predictions when triggers are present, which inherently induces distinct neuron activation patterns, particularly in channel importance. $\mathcal{L}_{\text{align}}$ is specifically engineered to exploit this discrepancy and identifies this functional anomaly by contrasting the deep-layer channel attention of the current global model against the trusted shallow-layer attention patterns from the historical model. Consequently, even when an attack like DPBA successfully preserves the original feature distribution, the specific neural activation signature it requires to execute the attack is inevitably captured and rectified by $\mathcal{L}_{\text{align}}$.

Backdoor Trigger Size. Moreover, we investigate the robustness of GBHINDER against backdoor triggers of varying sizes. We configured attacks by inserting a variable number of malicious nodes, ranging from 3 to 5, and evaluated the performance of GBHINDER against multiple attacks. The detailed experimental results are presented in Table 13.

Table 13: Performance of GBHINDER with different trigger sizes.

Attack	Trigger Size	No-Defense		GBHINDER	
		ACC	ASR	ACC	ASR
Rand-GDBA	**2**	77.32	20.26	79.72 (↑2.40)	6.32 (↓13.94)
	3	75.72	28.66	80.56 (↑4.84)	6.73 (↓21.93)
	4	74.73	56.28	80.12 (↑5.39)	2.31 (↓53.97)
Motif-GDBA	**2**	78.23	10.16	79.62 (↑1.39)	5.71 (↓4.45)
	3	77.36	54.65	79.35 (↑1.99)	5.26 (↓49.39)
	4	77.19	79.77	79.32 (↑2.13)	4.58 (↓75.19)
NI-GDBA	**2**	80.42	72.70	80.09 (↓0.33)	8.12 (↓64.58)
	3	80.42	86.94	80.09 (↓0.33)	8.64 (↓78.30)
	4	80.42	90.41	80.09 (↓0.33)	8.35 (↓82.06)
Opt-GDBA	**2**	79.32	42.71	79.42 (↑0.10)	6.71 (↓36.00)
	3	76.66	87.11	79.87 (↑3.21)	6.51 (↓80.60)
	4	75.18	82.2	78.92 (↑3.74)	1.98 (↓80.22)

From the perspective of benign clients, the primary objective is to mitigate the impact of the backdoor in the global model, irrespective of the specific trigger characteristics. However, results reveal a counterintuitive trend: the defensive efficacy of GBHINDER is actually enhanced when faced with larger triggers. We attribute this counterintuitive phenomenon to the fact that larger triggers introduce more significant and conspicuous perturbations to both the graph's topology and its node feature space. These more pronounced anomalies are more readily detected and penalized by regularization terms.

Table 14: Performance of GBHINDER under adaptive attacks.

Data	UGBA+		GBHINDER	
	ACC	ASR	ACC	ASR
NCI1	70.40	69.11	77.53	12.31
PROTEINS	72.86	44.62	74.07	6.71
DD	64.97	68.43	68.72	18.62
AIDS	99.11	38.45	99.52	3.57

F Adaptive Attacks

In this section, we investigate the resilience of GBHINDER against adaptive attacks. Given that UGBA already incorporates $\mathcal{L}_{\text{topo}}$ during trigger optimization, we construct an enhanced attacker, denoted as UGBA+, by integrating $\mathcal{L}_{\text{align}}$ into the optimization objective. The results presented in Table 14 reveal that the attacker faces a conflicting multi-objective optimization dilemma. The additional constraints required to bypass GBHINDER, specifically satisfying both inter-layer attention regularization and topological consistency, fundamentally degrade the effectiveness of backdoors.

G Computational Overhead

In this section, we analyze the computational overhead introduced by GBHINDER. We acknowledge that GBHINDER requires each benign participant to maintain its local model state from the previous round, which results in an additional memory footprint. However, we argue that this practice of retaining historical state is becoming a common and necessary strategy in advanced FL algorithms designed to tackle challenges like non-IID data and personalization [4, 19, 28]. Our work innovatively repurposes this stateful client paradigm for the crucial task of security.

Our primary focus here is to quantify the computational overhead incurred at the client side. We establish the standard FedAvg protocol as our baseline and compare GBHINDER against two FedGL-specific defense frameworks, FedTGE and GNNCert. The core computational costs in GBHINDER stem from two operations: (1) the computation of the $\mathcal{L}_{\text{topo}}$ loss, which involves iterating over the edge set of each graph in a batch, and (2) the calculation of the stability score δ for the adaptive update, which requires traversing the local dataset. To manage this overhead, we introduced sampling strategies. For simplicity, we evaluate two configurations by uniformly setting the sampling rates for the training batch, edge set, and the data subset for the δ calculation to 5% and 25%, respectively (down from a default of 50%). We measure the computational overhead on both the client and server sides, presenting it as an increase relative to FedAvg. For FedTGE, we adhere to its original configuration, which includes 10 additional energy epochs for its client-side calibration. Following the setup in [41], GNNCert requires computationally intensive local training on numerous subgraphs to maintain its effectiveness; otherwise, its defense performance degrades significantly. To simulate a challenging scenario, the experiments are conducted under GTA-GDBA attack with 30% malicious clients and a 50% data poisoning ratio. All experiments were executed on a single machine equipped with an NVIDIA GeForce RTX 5080 GPU with 32 GB of memory.

Table 15: Computational overhead of GBHINDER.

DataSets	Methods	ASR (%)	Client (s)	Server (s)
NCI1	Fedavg	92.67	0.5012	–
	FedTGE	18.96 (↓73.71)	2.0965 (↑1.5953)	+0.019
	CNNCert	13.75 (↓78.92)	1.7474 (↑1.2462)	+0.521
	GBHINDER (%5)	21.80 (↓70.87)	1.8686 (↑1.3674)	–
	GBHINDER (%25)	17.57 (↓75.10)	2.2751 (↑1.7739)	–
	GBHINDER (%50)	12.39 (↓80.28)	3.8260 (↑3.3248)	–
AIDS	Fedavg	86.96	0.2067	–
	FedTGE	12.97 (↓73.99)	1.1423 (↑0.9356)	+0.021
	CNNCert	31.47 (↓55.49)	0.8064 (↑0.5997)	+0.298
	GBHINDER (%5)	18.13 (↓68.83)	0.6932 (↑0.4865)	–
	GBHINDER (%25)	11.59 (↓75.37)	1.6507 (↑1.4440)	–
	GBHINDER (%50)	6.96 (↓80.00)	2.0160 (↑1.8093)	–
PROTEINS	Fedavg	90.62	0.1340	–
	FedTGE	30.18 (↓60.44)	1.0432 (↑0.9092)	+0.032
	CNNCert	20.35 (↓70.27)	0.4071 (↑0.2731)	+0.309
	GBHINDER (%5)	18.75 (↓71.87)	0.5146 (↑0.3806)	–
	GBHINDER (%25)	9.31 (↓81.31)	1.1395 (↑1.0055)	–
	GBHINDER (%50)	11.39 (↓79.23)	2.7264 (↑2.5924)	–
DD	Fedavg	79.97	0.1224	–
	FedTGE	18.97 (↓61.00)	0.7658 (↑0.6434)	+0.026
	CNNCert	35.47 (↓44.50)	0.6957 (↑0.5733)	+0.698
	GBHINDER (%5)	19.13 (↓60.84)	0.4932 (↑0.3708)	–
	GBHINDER (%25)	9.72 (↓70.25)	0.9507 (↑0.8283)	–
	GBHINDER (%50)	4.66 (↓75.31)	2.2160 (↑2.0936)	–

The experimental results reveal a clear trade-off. As shown in Table 15, GBHINDER introduces a discernible computational overhead on the client side compared to other defenses, yet critically, it imposes zero additional burden on the server, whereas other methods require extra server-side computation. Fortunately, the overhead of GBHINDER can be gracefully managed. By reducing the sampling rates, the computational cost can be substantially alleviated. While this entails a slight degradation in defensive performance, the overall robustness remains at a highly acceptable level. Furthermore, our ablation studies indicate that the contributions of the $\mathcal{L}_{\text{topo}}$ loss and the Adaptive Momentum Information Update (which requires the δ calculation) can become less pronounced under certain conditions, such as near-IID data distributions. This suggests that benign participants can dynamically adjust or even disable these modules based on their specific security requirements and data characteristics, offering a flexible balance between security and computational efficiency.