Towards Effective, Stealthy, and Persistent Backdoor Attacks Targeting Graph Foundation Models

Jiai Luo[1], Qingyun Sun[1], Lingjuan Lyu[2], Ziwei Zhang[1], Haonan Yuan[1],
Xingcheng Fu[3], Jianxin Li[1*]

[1]SKLCCSE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Sony AI, Zurich, Switzerland
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
{luojy, sunqy, zwzhang, yuanhn, lijx}@buaa.edu.cn, Lingjuan.Lv@sony.com, fuxc@gxnu.edu.cn

Abstract

Graph Foundation Models (GFMs) are pre-trained on diverse source domains and adapted to unseen targets, enabling broad generalization for graph learning. Despite that GFMs have attracted considerable attention recently, their vulnerability to backdoor attacks remains largely underexplored. A compromised GFM can introduce backdoor behaviors into downstream applications, posing serious security risks. However, launching backdoor attacks against GFMs is non-trivial due to three key challenges. (1) *Effectiveness*: Attackers lack knowledge of the downstream task during pre-training, complicating the assurance that triggers reliably induce misclassifications into desired classes. (2) *Stealthiness*: The variability in node features across domains complicates trigger insertion that remains stealthy. (3) *Persistence*: Downstream fine-tuning may erase backdoor behaviors by updating model parameters. To address these challenges, we propose GFM-BA[1], a novel Backdoor Attack model against Graph Foundation Models. Specifically, we first design a label-free trigger association module that links the trigger to a set of prototype embeddings, eliminating the need for knowledge about downstream tasks to perform backdoor injection. Then, we introduce a node-adaptive trigger generator, dynamically producing node-specific triggers, reducing the risk of trigger detection while reliably activating the backdoor. Lastly, we develop a persistent backdoor anchoring module that firmly anchors the backdoor to fine-tuning-insensitive parameters, enhancing the persistence of the backdoor under downstream adaptation. Extensive experiments demonstrate the effectiveness, stealthiness, and persistence of GFM-BA.

1 Introduction

Graph Foundation Models (GFMs) are designed to be pre-trained on various graph data from diverse domains, and subsequently adapted to a wide range of downstream tasks in the target domain (Liu et al. 2025; Mao et al. 2024; Shi et al. 2024a,b). Existing efforts (Zhao et al. 2024; Yu et al. 2024, 2025; Wang et al. 2024b) towards GFMs have demonstrated strong knowledge transfer from pre-training source domains to target domains, achieving superior performance. While the pre-training and adaptation paradigm (Zi et al. 2024;

Tang et al. 2024a; He and Hooi 2024; Lachi et al. 2024) has driven the success of GFMs, they also introduce new potential security vulnerabilities, particularly backdoor attacks (i.e., inserting backdoors into the model that cause it to misbehave when encountering certain triggers). For GFMs, attackers can exploit the pre-training stage to inject backdoors and release compromised pre-trained GNNs to the public. Downstream users who adopt these pre-trained models unknowingly inherit the backdoor, exposing their downstream applications to targeted manipulation. These threats pose risks to critical applications of GFMs such as drug discovery (Bongini, Bianchini, and Scarselli 2021) and financial fraud detection (Cheng et al. 2020).

Backdoor attacks for traditional GNNs have been extensively studied (Zhang et al. 2021; Xi et al. 2021; Dai et al. 2023; Zheng et al. 2023; Xu, Xue, and Picek 2021). However, backdoor attacks against traditional GNNs and GFMs have fundamental differences. As shown in Figure 1, existing backdoor attacks against GNNs operate under three presumed conditions where (1) labels for downstream tasks are accessible during the backdoor injection phase; (2) the training and downstream graphs originate from the same domain; (3) the backdoor model remains unchanged during the downstream usage. In the context of GFMs, all three conditions may not hold, *leading to three key challenges for designing backdoor attacks against GFMs*: (1) *Effectiveness*: During the pre-training stage, downstream task knowledge is inaccessible. How to ensure that the injected trigger consistently induces a specific label that aligns with the attacker's intent? (2) *Stealthiness*: The distribution and semantics of node features can vary significantly across different graph domains (Mao et al. 2024; Shi et al. 2024a). How to design triggers that remain stealthy across diverse downstream domains? (3) *Persistence*: Downstream adaptation may modify the learned model parameters, thereby erasing the backdoor effect (a phenomenon known as backdoor forgetting (Gu et al. 2023)). How can attackers embed backdoors that can remain persistently effective after downstream fine-tuning?

To address the aforementioned challenges, we propose GFM-BA, a novel model for performing Backdoor Attacks against Graph Foundation Models. First, to solve the *effectiveness* challenge and achieve label-specific manipulation without access to downstream knowledge, we design a label-free trigger association module, which links triggers to a

---

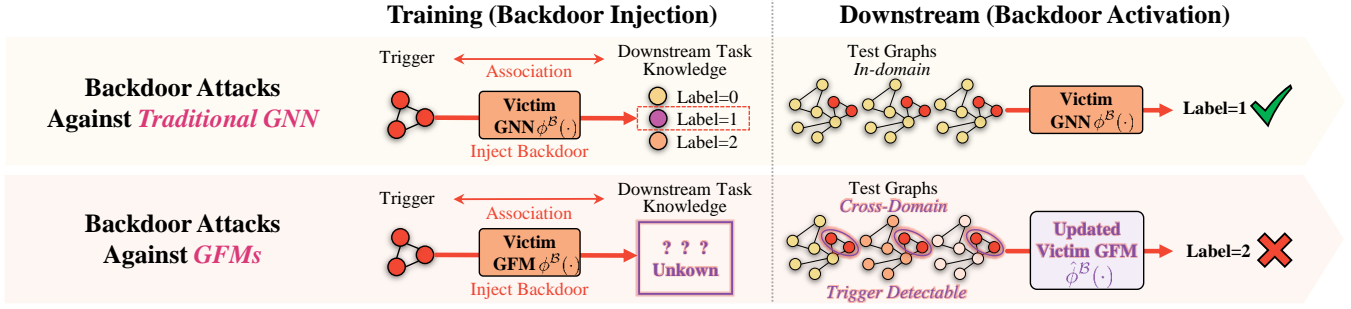[1]Code: https://github.com/RingBDStack/GFM-BA.

Figure 1: Key differences between backdoor attacks against traditional GNNs and GFMs.

set of prototype embeddings during pre-training. During the downstream trigger injection phase, the attacker can identify which prototype embedding aligns with the desired target label and injects the corresponding trigger through a few trial queries. Second, to solve the *Stealthiness* challenge and ensure that the injected trigger remains stealthy across diverse downstream domains, we develop a node-adaptive trigger generator, which dynamically produces context-aware triggers conditioned on the context of each target node instead of using fixed triggers. It enhances stealthiness while ensuring reliable backdoor activation. Besides, GFM-BA does not modify the clean pre-trained model but instead activates the latent backdoor already hidden in the encoder. Lastly, to solve the *persistence* challenge and maintain the functionality of the backdoor after downstream fine-tuning, we introduce a persistent backdoor anchoring module, which anchors the backdoor to parameters in the pre-trained model that are unlikely to change significantly during downstream adaptation. By embedding the backdoor into these stable regions of the model, the trigger-target association becomes less susceptible to being forgotten.

Our main contributions can be summarized as follows:

- We study backdoor attacks against Graph Foundation Models, highlighting the significant trustworthiness concerns in the development of GFMs.

- We propose GFM-BA, a novel backdoor attack model for GFMs containing three tailored modules targeting the *effectiveness*, *stealthiness*, and *persistence* challenges.

- We conduct extensive experiments and show that GFM-BA consistently outperforms existing methods against three representative victim GFMs, demonstrating its superior performance.

2 Related Work

Graph Foundation Models. Graph Foundation Models (GFMs) aim to capture generalizable graph knowledge that enables positive transfer across tasks and domains (Liu et al. 2025; Mao et al. 2024; Shi et al. 2024a; Wang et al. 2024b; Zi et al. 2024; Tang et al. 2024a; He and Hooi 2024; Lachi et al. 2024; Xia, Kao, and Huang 2024). They are typically pre-trained using self-supervised objectives, such as link prediction (Yu et al. 2024; Zhang and Chen 2018) or graph contrastive learning (Zhao et al. 2024; Yu et al. 2025), over multiple source datasets. The resulting model is then adapted to downstream tasks on target graphs through

task-specific fine-tuning (Zhao et al. 2024; Hassani 2022; You et al. 2020) or prompting (Sun et al. 2022a, 2023, 2022b; Fang et al. 2023; Tang et al. 2024b). For example, GCOPE (Zhao et al. 2024) attempts to mitigate the negative transfer by introducing domain-specific virtual nodes that interconnect nodes across domains, aligning the semantic patterns. MDGPT (Yu et al. 2024) introduces a two-stage prompting strategy to adapt target domains by integrating unified multi-domain knowledge with domain-specific information. SAMGPT (Yu et al. 2025) further introduces structure tokens to align various structural distributions. However, the trustworthiness of GFMs remains largely unexplored in the current literature.

Graph Backdoor Attacks. Graph backdoor attacks (Lyu et al. 2024; Zhang et al. 2023, 2021; Xi et al. 2021; Dai et al. 2023) aim to manipulate backdoored GNNs to predict a specified label for any input embedded with triggers (typically small subgraphs (Zheng et al. 2023; Xu, Xue, and Picek 2021; Wang et al. 2024a; Yang et al. 2024; Xu and Picek 2022; Yang, Li, and Li 2024; Feng et al. 2024; Luo et al. 2025)). Early efforts introduced subgraph-based triggers and achieved strong performance (Zhang et al. 2021; Xi et al. 2021). Subsequent methods enhanced stealthiness by leveraging in-distribution triggers to evade outlier detection (Dai et al. 2023; Zhang et al. 2024). However, their focus on supervised GNNs makes them unsuitable for backdoor attacks on GFMs during pre-training. GCBA (Zhang et al. 2023) is the first backdoor attack targeting graph contrastive learning, but it requires knowledge of downstream class labels, limiting its applicability to GFMs. CrossBA (Lyu et al. 2024) introduces the first cross-domain pre-training graph backdoor attack by aligning triggered graphs with a learned trigger embedding while separating them from clean samples. Yet it cannot control the predicted label, making the backdoor attack degenerate to a form of adversarial evasion attack (Sun et al. 2022a; Kwon and Kim 2025). Additionally, the use of a fixed trigger across domains increases its detectability. Overall, prior methods either rely on downstream knowledge or lack control over the attack outcome, limiting their effectiveness and stealthiness in the GFM setting.

3 Threat Model

Attacker's Goals: Consistent with prior work (Lyu et al. 2024; Zhang et al. 2023; Xi et al. 2021), we focus on the node classification task. In the context of backdoor attacks

against GFMs, attackers aim to inject a backdoor into the pre-trained encoder $\phi(\cdot)$, resulting in a compromised model $\phi^{\mathcal{B}}(\cdot)$. The goal is to induce any downstream classifiers $f(\cdot)$ built on adapted $\hat{\phi}^{\mathcal{B}}(\cdot)$ to misclassify triggered graphs into a specific label y^*, while maintaining normal performance on clean inputs. Formally, the attack objective is:

$$f(\hat{\phi}^{\mathcal{B}}(\mathcal{G}', v)) = y^*, \quad f(\hat{\phi}^{\mathcal{B}}(\mathcal{G}, v)) = f(\hat{\phi}(\mathcal{G}, v)), \quad (1)$$

where $\mathcal{G}$ is a clean graph, $\mathcal{G}'$ is the corresponding triggered graph, and v denotes the target node.

Attacker's Knowledge and Capabilities: Following CrossBA (Lyu et al. 2024), attackers have full control over the pre-training process but have no access to any downstream information. ***This setting is realistic, as foundation model training is resource-intensive, and downstream users often rely on open-source models or APIs to build their applications.*** *Thus, a malicious adversary can release an open-source GFM, and downstream users who adopt it may unknowingly inherit the embedded backdoor.* During the backdoor activation stage, the attacker can inject triggers into target nodes and observe feedback signals. For example, in paper ranking systems based on citation networks, attackers can publish dummy papers that cite the target paper. Public leaderboards then allow attackers to observe the feedback on the impact. In e-commerce, fake user accounts can manipulate and monitor recommendation outcomes.

4 GFM-BA

In this section, we elaborate GFM-BA, a novel model for <u>B</u>ackdoor <u>A</u>ttacks against <u>G</u>raph <u>F</u>oundation <u>M</u>odels, designed to achieve effectiveness, stealthiness, and persistence simultaneously. The overall architecture of GFM-BA is shown in Figure 2, which consists of three components:

(1) *Label-Free Trigger Association Module (Challenge 1: Effectiveness)*. We apply Farthest Point Sampling to select prototype embeddings from the node embedding space of pre-training graphs generated by the pre-trained GNN encoder, which serve as targets for trigger association.

(2) *Node-Adaptive Trigger Generator (Challenge 2: Stealthiness)*. We design a trigger generator to dynamically produce the personalized trigger for each target node. The generated trigger is optimized to align the triggered target node with the target embedding while maintaining graph homophily, ensuring both effectiveness and stealthiness.

(3) *Persistent Backdoor Anchoring Module (Challenge 3: Persistence)*. We enhance backdoor persistence under downstream adaptation by anchoring the trigger-target association to fine-tuning-insensitive parameters of the pre-trained model. Specifically, we apply graph mixup to simulate distribution shifts and identify fine-tuning-sensitive parameters. Random perturbations are then applied to anchor the trigger-target mapping to the identified stable regions of the model.

4.1 Label-Free Trigger Association Module

During pre-training, attackers lack access to the downstream model architecture and its decision boundaries. As a result, existing graph backdoor attacks that require the downstream task labels are infeasible in the context of GFMs (Zhang

et al. 2023, 2021; Dai et al. 2023). While methods such as CrossBA (Lyu et al. 2024) avoid this limitation by pushing the triggered graph away from clean graphs and toward a learned trigger, they lack control over the resulting label, causing the triggered graph to be classified into a fixed but unknown class, failing to guide it to a specific target label.

To overcome this challenge, we propose associating triggers with a set of prototype embeddings rather than with specific labels. However, naively generating these prototype embeddings at random may result in all of them corresponding to a fixed and undesired class, whereas our goal is to construct prototype embeddings that span diverse downstream classes, enabling the attacker to steer the triggered graph toward a desired label by selecting an appropriate target embedding through several trial queries during the trigger injection phase. Following the assumption in prior work (Zhao et al. 2024; Yu et al. 2024, 2025) that pre-training data contains knowledge relevant to downstream applications, sampling prototype embeddings from the node embedding space of pre-training graphs can effectively reduce the search space. Specifically, we employ the Farthest Point Sampling (FPS) (Eldar et al. 1997), which is a greedy algorithm that iteratively selects the most distant point to ensure that the sampled set preserves the overall structure and distribution of the original data (Eldar et al. 1997; Lang, Manor, and Avidan 2020). Here, we propose Proposition 1.

Proposition 1. *When point density decays monotonically from each class centroid, increasing the separation between centroids raises the probability that FPS will cover more classes in a fixed number of steps.*

The detailed proof can be found in Appendix A. Based on Proposition 1 and prior findings (Chen et al. 2020; Morcos 2022) that self-supervised pre-training enhances class separability, adopting FPS to select prototype embeddings from the node embeddings of the pre-training graphs encourages sampled embeddings to cover a broader range of downstream classes. *Note that while "density decays monotonically from class centroid" is idealized and not strictly satisfied in practice, many real-world datasets exhibit this overall trend (Idrissi et al. 2015), making proposition 1 a useful theoretical motivation for using FPS.* Specifically, let $\mathcal{E}$ denote the set of node embeddings extracted by the pre-trained GNN from the pre-training graphs. The target embedding set $\mathcal{E}_{\text{target}}$ is then constructed by selecting k prototype embeddings from $\mathcal{E}$ using FPS, denoted as $\mathcal{E}_{\text{target}} = \text{FPS}(\mathcal{E}, k)$.

After associating triggers with $\mathcal{E}_{\text{target}}$, in the downstream trigger injection stage, attackers can select a target embedding from $\mathcal{E}_{\text{target}}$ based on the downstream application and generate its corresponding trigger using the trigger generator (see Section 4.2) to activate the desired backdoor behavior.

4.2 Node-Adaptive Trigger Generator

Graphs from different domains exhibit significant feature distribution discrepancy (Shi et al. 2024a). Existing backdoor methods that use fixed, domain-agnostic triggers can introduce feature inconsistencies between the injected trigger node and its neighbors in the downstream graph, violating the graph homophily (Jin et al. 2020), thus making the
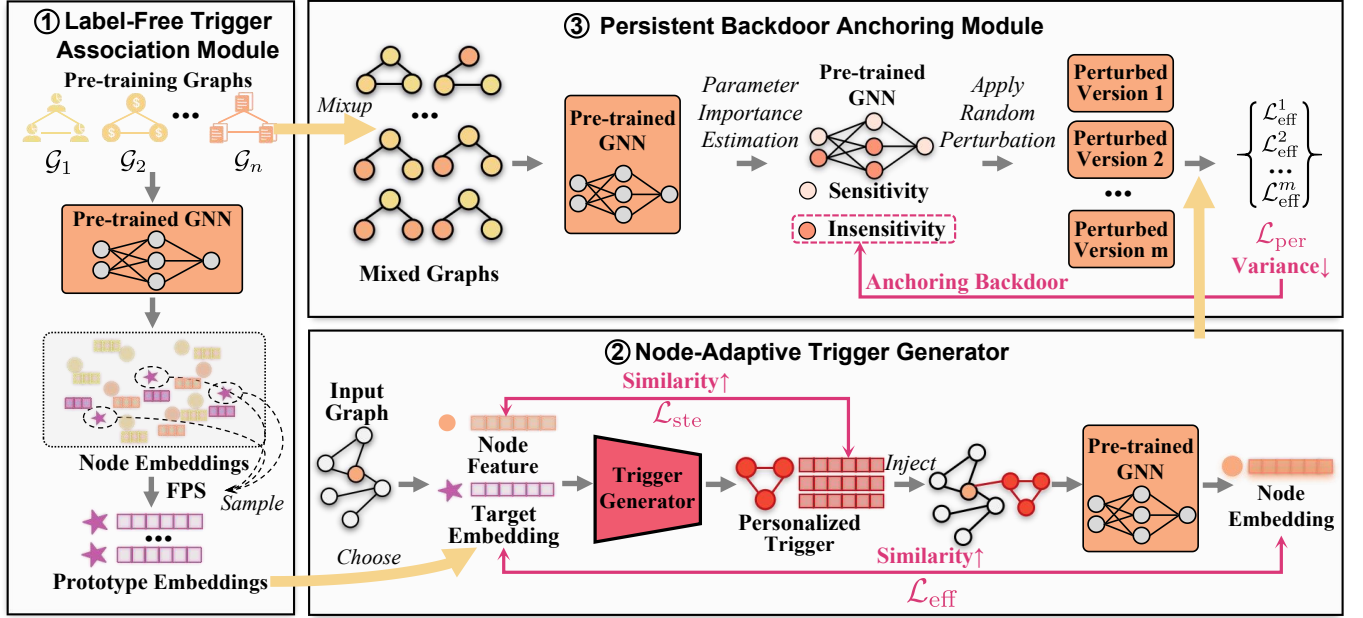
Figure 2: The overall framework of GFM-BA. FPS is first applied to select prototype embeddings as trigger association targets. The node-adaptive trigger generator then produces personalized triggers conditioned on the target embedding and the target node, ensuring both stealthiness and effectiveness. Finally, graph mixup is used to identify fine-tuning-insensitive parameters, allowing the backdoor to be anchored in the stable regions of the model and remain effective after the downstream adaptation.

injected trigger more detectable and easier to remove.

To avoid this problem, we replace the universal trigger with a trigger generator that dynamically produces a personalized trigger conditioned on both the target node and the selected target embedding. Specifically, we implement MLP as the trigger generator. Consistent with prior work (Lyu et al. 2024), we design the trigger as a fully connected 3-node graph with identical features, which is much smaller than both pre-training and downstream graphs. Let $\mathbf{x}_i$ denote the feature of the target node and $\mathbf{e}_j$ the target embedding chosen from $\mathcal{E}_{\text{target}}$. The feature of the injected trigger node, denoted as $\mathbf{x}_{ij}^{\text{tri}}$, is generated as follows:

$$\mathbf{x}_{ij}^{\text{tri}} = \text{MLP}\left([\mathbf{x}_i\|\mathbf{e}_j]\right), \qquad (2)$$

where $[\cdot \| \cdot]$ denotes vector concatenation. We have two objectives for the trigger: (1) ensuring that the triggered graph is associated with the target embedding to guarantee effectiveness, and (2) maintaining feature similarity with the connected target node to enhance stealthiness. Let $\mathcal{G}_i$ denote a clean graph from the pre-training dataset $\{\mathcal{G}_u\}_{u=1}^n$, and $\tilde{\mathcal{G}}_{ij}$ denote the 3-node trigger graph feature $\mathbf{x}_{ij}^{\text{tri}}$. We optimize the trigger generator with $\mathcal{L}_{\text{eff}}$ and $\mathcal{L}_{\text{ste}}$:

$$\mathcal{L}_{\text{eff}} = -\mathbb{E}_{u\sim[n],j\sim[k],i\sim[|\mathcal{G}_u|]}\text{sim}\left(\phi(\text{In}(\mathcal{G}_u,\tilde{\mathcal{G}}_{ij},i)),e_j\right), \quad (3)$$

$$\mathcal{L}_{\text{ste}} = -\mathbb{E}_{u\sim[n],j\sim[k],i\sim[|\mathcal{G}_u|]}\text{sim}\left(x_i, x_{ij}^{\text{tri}}\right). \qquad (4)$$

where $|\mathcal{G}_u|$ denotes the number of nodes in $\mathcal{G}_u$, $\text{In}(\mathcal{G}_u, \tilde{\mathcal{G}}_{ij}, i)$ denotes inserting the trigger graph $\tilde{\mathcal{G}}_{ij}$ into the clean graph $\mathcal{G}_u$ with the i-th node as the target node, $\phi(\cdot)$ is the frozen
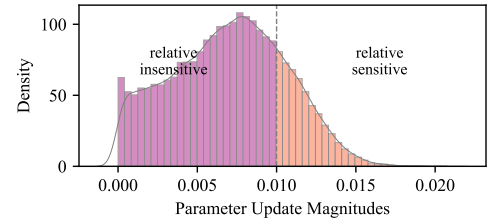

Figure 3: An analysis of the distribution of parameter update magnitudes after downstream fine-tuning.

pre-trained GNN encoder, and $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity. Here, $\mathcal{L}_{\text{eff}}$ promotes the association between the embedding of the triggered graph and the target embedding, while $\mathcal{L}_{\text{ste}}$ encourages the feature similarity between the trigger node and the connected target node. Notably, the backdoor vulnerability is introduced without the need to modify the parameters of the pre-trained GNN, but instead leverages the inherent latent backdoor logic already present in the encoder, enabling the GFMs to preserve their performance on clean graphs without any degradation.

4.3 Persistent Backdoor Anchoring Module

During the downstream model construction, downstream users may fine-tune the pre-trained GNN, and such parameter updates can invalidate the backdoor, known as backdoor forgetting (Gu et al. 2023).

To mitigate this negative effect, we draw inspiration from the backdoor attacks in NLP (Cheng et al. 2025) where

triggers are crafted using rare tokens to minimize the likelihood of related parameters being updated during downstream fine-tuning. Our key idea is to anchor the association between the trigger and the target embedding to parameters of the pre-trained GNN that are insensitive to fine-tuning in downstream applications. To investigate this possibility, we conducted a preliminary experiment in which we measured the magnitude of parameter updates of a pre-trained GCN (Kipf and Welling 2017) during downstream fine-tuning, with the details can be found in Appendix C. As shown in Figure 3, most parameters are insensitive (i.e., change slightly after fine-tuning), while only a few are significantly updated. This motivates us to anchor the trigger–target association to these insensitive parameters.

Proposition 2. *Consider a GNN with parameters* Θ. *Let* $\mathbf{X}$ *be the node features,* $\mathbf{A}$ *be the adjacency matrix, and* $\mathbf{Z}$ *be the output. Assume the Jacobian matrix* $J_{\partial \mathbf{Z} / \partial \theta_k}(\mathbf{X})$ *is rank-deficient. For any direction* $\Delta \Theta \in \bigcup_{\mathbf{X}} \ker J_{\partial \mathbf{Z} / \partial \theta_k}(\mathbf{X})$, *there exists an input* $\mathbf{X}$ *such that* $\mathbf{Z}$ *is first-order insensitive to* $\Delta \Theta$, *where* ker *denotes the null space of the Jacobian matrix, and* $\bigcup$ *denotes the union operator.*

The proof is provided in Appendix A. Proposition 2 suggests the possibility that the association between the trigger and the target embedding can be anchored to insensitive parameters of GNNs. To achieve this goal, we first apply graph mixup (Ling et al. 2023) to explore potential downstream graph patterns. We then adopt the parameter importance estimation method from model pruning (Molchanov et al. 2019), leveraging the pre-training loss on mixed graphs to identify insensitive parameters. Finally, we introduce random perturbations to the identified sensitive parameters during the training process of the trigger generator, guiding the generator to anchor the trigger-target link to the insensitive parameters. Specifically, for each $(\mathcal{G}_i = \{\mathbf{X}_i, \mathbf{A}_i\}, \mathcal{G}_j = \{\mathbf{X}_j, \mathbf{A}_j\})$ in the pre-training dataset with n graphs, we synthesize a mixed graph $\mathcal{G}^{\mathrm{mix}} = \{\mathbf{X}^{\mathrm{mix}}, \mathbf{A}^{\mathrm{mix}}\}$ as follows:

$$\begin{aligned} \mathbf{A}^{\mathrm{mix}} &= \lambda \mathbf{A}_i + (1 - \lambda) \mathbf{M} \mathbf{A}_j \mathbf{M}^\top, \\ \mathbf{X}^{\mathrm{mix}} &= \lambda \mathbf{X}_i + (1 - \lambda) \mathbf{M} \mathbf{X}_j. \end{aligned} \quad (5)$$

where λ is the hyperparameter, $\mathbf{M}$ is the alignment matrix calculated as $\mathbf{M} = \mathrm{softmax}(\mathrm{sim}(\mathbf{R}_i, \mathbf{R}_j))$, with $\mathbf{R}_i$ and $\mathbf{R}_j$ denoting node representations of $\mathcal{G}_i$ and $\mathcal{G}_j$ computed via two rounds of message-passing.

We treat the mixed graph set $\{\mathcal{G}_i^{\mathrm{mix}}\}_{i=1}^{n(n-1)}$ as a proxy for potential downstream patterns, based on the widely accepted assumption that pre-training graphs encode knowledge relevant to downstream applications (Mao et al. 2024; Shi et al. 2024a; Zhao et al. 2024). To estimate parameters sensitive to fine-tuning, we adopt the importance-based sensitivity measure from model pruning (Molchanov et al. 2019). Given the pre-training loss $\mathcal{L}_{\mathrm{pre}}$ evaluated on the mixed graph set, the sensitivity of parameter θ_k is defined as:

$$\mathcal{I}(\theta_k) = \left(g_k \theta_k - \frac{1}{2} \theta_k \mathbf{H}_k \Theta \right)^2, \quad (6)$$

where $g_k = \partial \mathcal{L}_{\mathrm{pre}} / \partial \theta_k$ and $\mathbf{H}_k$ represents the k-th row of the corresponding Hessian matrix. A larger value of $\mathcal{I}(\theta_k)$

implies that the parameter θ_k is more sensitive. We then introduce random perturbations to the top $s\%$ most sensitive ones. For each selected sensitive parameter θ_k, the perturbation is applied as $\theta_k \leftarrow \theta_k + \epsilon |\theta_k|$, where ϵ is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$. We perform m perturbations and obtain a set of perturbed parameters $\{\Theta^{(i)}\}_{i=1}^m$ of the pre-trained GNN. These perturbed parameters will yield a set of effective loss $\mathcal{L}_{\mathrm{eff}}$ as defined in Eq. (3), denoted as $\{\mathcal{L}_{\mathrm{eff}}^j\}_{j=1}^m$. To mitigate the degradation of backdoor effectiveness caused by sensitive parameters during downstream adaptation, we introduce a persistence loss $\mathcal{L}_{\mathrm{per}}$:

$$\mathcal{L}_{\mathrm{per}} = \mathrm{Var}(\{\mathcal{L}_{\mathrm{eff}}^j\}_{j=1}^m) + \mathrm{Mean}(\{\mathcal{L}_{\mathrm{eff}}^j\}_{j=1}^m). \quad (7)$$

Minimizing $\mathcal{L}_{\mathrm{per}}$ enhances the persistence of trigger-target association under downstream fine-tuning.

The overall pipeline of GFM-BA is in Appendix B.

5 Experiments
5.1 Experiments Settings
Datasets. We evaluate GFM-BA on five widely used node-level classification datasets for GFMs. Cora (Yang, Cohen, and Salakhudinov 2016), CiteSeer (Yang, Cohen, and Salakhudinov 2016), and PubMed (Yang, Cohen, and Salakhudinov 2016) are citation networks where nodes correspond to scientific publications and edges represent citation links. Computers (Shchur et al. 2018) and Photos (Shchur et al. 2018) are the Amazon co-purchase dataset, where edges indicate frequent co-purchase relationships between products. For dataset splitting, we use the standard PyG (Fey and Lenssen 2019) splits for Cora, CiteSeer, and PubMed. Following (Yu et al. 2024), we designate the last 100 nodes as the test set for Computers and Photo, and randomly sample m labeled nodes from the remaining nodes to form the m-shot training set.

Victim GFMs. We evaluate the backdoor performance of GFM-BA against three state-of-the-art GFMs designed for multi-domain graph pre-training and adaptation: (1) GCOPE (Zhao et al. 2024) employs virtual nodes to interconnect graphs across domains, aligning the semantics of graphs. (2) MDGPT(Yu et al. 2024) introduces dual prompts to adapt to target domains while integrating unified multi-domain knowledge with a tailored mixture of domain-specific prompts. (3) SAMGPT (Yu et al. 2025) incorporates structural tokens to unify multi-domain structural knowledge and adapt it effectively to unseen domains.

Baselines. We adopt CrossBA (Lyu et al. 2024), the state-of-the-art graph backdoor method against cross-domain graph pre-training, as our primary baseline. Following (Lyu et al. 2024), we also include two adapted variants of GCBA (Zhang et al. 2023): GCBA_R, which randomly selects a cluster center as the target embedding, and GCBA_M, which selects the most isolated cluster center.

Implement Details. Following prior works (Zhao et al. 2024; Yu et al. 2024, 2025), GFMs are trained across all datasets except the one held out for testing. All experiments are conducted in a 5-shot node classification setting. A grid search over α and β is performed in the range $[1e^{-2}, 1e^0]$ using a logarithmic step size of 5. Each experiment is repeated

Dataset		Cora		CiteSeer		PubMed		Photo		Computers	
Victim	**Threaten**	**ASR** (*Scen.1*)	**ASR** (*Scen.2*)	**ASR** (*Scen.1*)	**ASR** (*Scen.2*)	**ASR** (*Scen.1*)	**ASR** (*Scen.2*)	**ASR** (*Scen.1*)	**ASR** (*Scen.2*)	**ASR** (*Scen.1*)	**ASR** (*Scen.2*)
GCOPE	GCBA_R	$26.78_{\pm8.89}$	$3.83_{\pm1.27}$	$29.64_{\pm8.67}$	$4.94_{\pm1.45}$	$62.10_{\pm12.81}$	$20.70_{\pm4.27}$	$26.80_{\pm8.79}$	$3.35_{\pm1.10}$	$35.40_{\pm19.13}$	$3.54_{\pm1.91}$
	GCBA_M	$33.42_{\pm1.89}$	$4.77_{\pm0.27}$	$35.87_{\pm6.50}$	$5.98_{\pm1.08}$	$64.94_{\pm15.18}$	$21.65_{\pm5.06}$	$27.80_{\pm9.26}$	$3.48_{\pm1.16}$	$46.20_{\pm12.34}$	$4.62_{\pm1.23}$
	CrossBA	$\mathbf{100.00_{\pm0.00}}$	$\underline{14.29_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\underline{16.67_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\underline{33.33_{\pm0.00}}$	$\underline{74.00_{\pm13.44}}$	$\underline{9.25_{\pm1.68}}$	$\underline{79.80_{\pm7.26}}$	$\underline{7.98_{\pm0.73}}$
	GFM-BA	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{90.40_{\pm12.10}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{89.06_{\pm8.02}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{93.20_{\pm7.05}}$	$\mathbf{84.53_{\pm1.79}}$	$\mathbf{94.40_{\pm7.70}}$	$\mathbf{78.54_{\pm7.28}}$
SAMGPT	GCBA_R	$61.66_{\pm8.82}$	$8.81_{\pm1.26}$	$51.60_{\pm13.57}$	$8.60_{\pm2.26}$	$89.02_{\pm18.39}$	$29.67_{\pm6.13}$	$52.00_{\pm9.95}$	$6.50_{\pm1.24}$	$66.60_{\pm6.66}$	$6.66_{\pm0.67}$
	GCBA_M	$61.44_{\pm13.41}$	$8.78_{\pm1.92}$	$64.42_{\pm15.49}$	$10.74_{\pm2.58}$	$94.82_{\pm4.64}$	$31.61_{\pm1.55}$	$57.40_{\pm9.58}$	$7.18_{\pm1.20}$	$68.00_{\pm12.83}$	$6.80_{\pm1.28}$
	CrossBA	$\underline{95.24_{\pm10.64}}$	$\underline{13.61_{\pm1.52}}$	$\mathbf{100.00_{\pm0.00}}$	$\underline{16.67_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\underline{33.33_{\pm0.00}}$	$\underline{96.80_{\pm4.55}}$	$\underline{12.10_{\pm0.57}}$	$\underline{92.00_{\pm11.75}}$	$\underline{9.20_{\pm1.17}}$
	GFM-BA	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{99.80_{\pm0.23}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$
MDGPT	GCBA_R	-	-	-	-	-	-	-	-	-	-
	GCBA_M	-	-	-	-	-	-	-	-	-	-
	CrossBA	$\underline{95.14_{\pm5.26}}$	$\underline{13.59_{\pm0.75}}$	$\underline{92.36_{\pm15.87}}$	$\underline{15.39_{\pm2.65}}$	$\mathbf{100.00_{\pm0.00}}$	$\underline{33.33_{\pm0.00}}$	$\underline{82.60_{\pm19.86}}$	$\underline{10.32_{\pm2.48}}$	$\underline{82.00_{\pm19.21}}$	$\underline{8.20_{\pm1.92}}$
	GFM-BA	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{96.61_{\pm6.17}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{99.43_{\pm0.95}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{98.40_{\pm2.30}}$	$\mathbf{97.68_{\pm2.57}}$	$\mathbf{99.20_{\pm1.30}}$	$\mathbf{93.19_{\pm3.66}}$

Table 1: The results of attack effectiveness. *Scen.1* refers to *target-uncontrolled attack*, and *Scen.2* refers to *target-controlled attack*. "-" indicates that the method is not applicable to perform the backdoor attack (GCBA (Zhang et al. 2023) is tailored for the graph contrastive learning method). The best results are shown in **bold** and the runner-ups are underlined.

Dataset		Cora		CiteSeer		PubMed		Photo		Computers	
Victim	**Threaten**	**ACC** (*Clean*)	**ASR** (*Purified*)	**ACC** (*Clean*)	**ASR** (*Purified*)	**ACC** (*Clean*)	**ASR** (*Purified*)	**ACC** (*Clean*)	**ASR** (*Purified*)	**ACC** (*Clean*)	**ASR** (*Purified*)
GCOPE	GCBA_R	$59.18_{\pm3.93}$	$20.87_{\pm9.57}$	$56.30_{\pm2.86}$	$27.28_{\pm12.83}$	$55.00_{\pm4.39}$	$42.18_{\pm13.59}$	$57.00_{\pm3.24}$	$26.60_{\pm9.56}$	$44.60_{\pm3.71}$	$35.00_{\pm20.04}$
	GCBA_M	$59.78_{\pm7.20}$	$23.27_{\pm8.89}$	$56.12_{\pm2.13}$	$22.62_{\pm6.51}$	$51.10_{\pm5.79}$	$48.71_{\pm22.34}$	$58.80_{\pm7.33}$	$26.80_{\pm8.90}$	$47.20_{\pm7.26}$	$43.20_{\pm13.55}$
	CrossBA	$\underline{60.52_{\pm1.68}}$	$\underline{52.04_{\pm2.42}}$	$\underline{59.06_{\pm4.45}}$	$\underline{57.58_{\pm2.80}}$	$\underline{51.36_{\pm5.37}}$	$\underline{90.65_{\pm2.52}}$	$\underline{65.60_{\pm3.51}}$	$\underline{67.20_{\pm9.73}}$	$\underline{50.40_{\pm4.83}}$	$\underline{48.80_{\pm15.85}}$
	GFM-BA	$\mathbf{61.46_{\pm3.25}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{60.10_{\pm5.72}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{54.44_{\pm4.43}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{65.80_{\pm4.97}}$	$\mathbf{100.00_{\pm0.00}}$	$\mathbf{54.80_{\pm7.56}}$	$\mathbf{100.00_{\pm0.00}}$
SAMGPT	GCBA_R	$60.36_{\pm1.89}$	$38.24_{\pm6.24}$	$44.10_{\pm4.47}$	$43.92_{\pm10.22}$	$58.92_{\pm4.35}$	$77.80_{\pm17.21}$	$79.20_{\pm4.32}$	$17.80_{\pm3.11}$	$69.00_{\pm6.00}$	$26.20_{\pm4.15}$
	GCBA_M	$61.94_{\pm3.56}$	$29.32_{\pm5.45}$	$48.76_{\pm4.09}$	$48.38_{\pm12.30}$	$63.02_{\pm2.60}$	$86.26_{\pm9.05}$	$76.20_{\pm5.67}$	$18.20_{\pm3.11}$	$\mathbf{71.20_{\pm2.59}}$	$23.20_{\pm4.82}$
	CrossBA	$\underline{62.82_{\pm2.85}}$	$\underline{70.28_{\pm11.22}}$	$\underline{59.04_{\pm2.86}}$	$\underline{75.90_{\pm1.95}}$	$\underline{65.64_{\pm4.40}}$	$\underline{74.08_{\pm10.75}}$	$\underline{80.40_{\pm4.34}}$	$\underline{67.80_{\pm2.39}}$	$67.20_{\pm7.12}$	$\underline{50.60_{\pm6.95}}$
	GFM-BA	$\mathbf{63.54_{\pm3.73}}$	$\mathbf{88.28_{\pm3.08}}$	$\mathbf{61.72_{\pm3.61}}$	$\mathbf{86.04_{\pm1.56}}$	$\mathbf{65.92_{\pm5.65}}$	$\mathbf{93.62_{\pm2.20}}$	$\mathbf{80.60_{\pm3.58}}$	$\mathbf{86.00_{\pm2.12}}$	$\underline{69.20_{\pm4.09}}$	$\mathbf{84.60_{\pm2.19}}$
MDGPT	GCBA_R	-	-	-	-	-	-	-	-	-	-
	GCBA_M	-	-	-	-	-	-	-	-	-	-
	CrossBA	$\underline{42.36_{\pm7.08}}$	$\underline{47.76_{\pm13.84}}$	$\underline{37.82_{\pm5.00}}$	$\underline{61.12_{\pm20.64}}$	$\underline{50.20_{\pm6.87}}$	$\underline{69.10_{\pm20.81}}$	$\underline{68.20_{\pm9.09}}$	$\underline{35.40_{\pm5.73}}$	$\underline{50.20_{\pm6.10}}$	$\underline{37.00_{\pm8.69}}$
	GFM-BA	$\mathbf{60.88_{\pm4.83}}$	$\mathbf{81.30_{\pm3.60}}$	$\mathbf{60.58_{\pm2.82}}$	$\mathbf{85.78_{\pm2.96}}$	$\mathbf{62.48_{\pm4.71}}$	$\mathbf{92.36_{\pm1.97}}$	$\mathbf{79.20_{\pm6.14}}$	$\mathbf{89.60_{\pm5.68}}$	$\mathbf{71.80_{\pm3.49}}$	$\mathbf{85.20_{\pm4.82}}$

Table 2: Results of attack stealthiness. ACC reports the accuracy of the backdoored model on clean, non-triggered input graphs. *Purified* refers to applying edge-based purification to the triggered graphs under Scenario 1.

5 times on a single NVIDIA V100 GPU. Full hyperparameter configurations and more details are in Appendix C.

Evaluation Metrics. Following (Lyu et al. 2024), we adopt two evaluation metrics: (1) Attack Success Rate (ASR) measures the proportion of triggered samples misclassified into the target class. (2) Accuracy (ACC) measures performance on clean test data.

5.2 Attack Effectiveness Evaluation

To evaluate the effectiveness of GFM-BA, we consider two attack scenarios based on whether the attacker can specify the target label. **Scenario 1: Target-Uncontrolled Attack**: The model is expected to misclassify triggered samples consistently into an arbitrary but fixed target class. **Scenario 2: Target-Controlled Attack**. The model is expected to misclassify triggered samples into the attacker-specified target class, aligned with the attacker's intent.

As shown in Table 1, GFM-BA consistently achieves the highest ASR across all datasets in both attack scenarios,

demonstrating its strong attack capability. In Scenario 2, the improvements are particularly notable, with gains ranging from 66.67% to 90.80% over the baselines across five datasets. This is because baseline methods associate the trigger with an unknown fixed target label and cannot align it with the attacker-specified label. Consequently, they degenerate into targeted adversarial evasion attacks (Sun et al. 2022a; Kwon and Kim 2025) and fail to satisfy the requirements of a targeted backdoor attack in Scenario 2.

5.3 Attack Stealthiness Evaluation

To ensure the practical utility of backdoor attacks, stealthiness is a critical requirement that involves two key aspects: (1) the backdoored model should maintain normal behavior on clean input graphs; and (2) the inserted trigger should be inconspicuous and resistant to detection. To assess the first aspect, we measure the ACC of the backdoored GFMs on the clean test set. To assess the second aspect, we apply a simple purification strategy that removes edges between node pairs

Dataset	Cora		Photo		Computers	
Victim | **Threaten**	**ASR**	*Drop*	**ASR**	*Drop*	**ASR**	*Drop*
SAMGPT | GCBA_R	52.22±15.68	↓9.44	38.80±14.99	↓13.20	28.80±3.96	↓6.60
| GCBA_M	54.58±9.34	↓6.86	44.40±21.11	↓13.00	40.60±10.45	↓5.60
| CrossBA	90.50±13.68	↓4.74	87.40±12.64	↓9.40	91.40±10.50	↓0.60
| **GFM-BA**	**98.66±1.85**	↓*1.34*	**96.00±2.55**	↓*4.00*	**98.60±1.95**	↓*1.40*
MDGPT | GCBA_R	-	-	-	-	-	-
| GCBA_M	-	-	-	-	-	-
| CrossBA	93.78±8.05	↓*1.36*	78.00±13.38	↓*4.60*	79.60±16.95	↓*2.40*
| **GFM-BA**	**99.32±1.31**	↓*0.68*	**97.80±1.64**	↓*0.60*	**98.40±2.19**	↓*0.80*

Table 3: Results of attack persistence under Scenario 1. *Drop* indicates the decrease in ASR compared to the setting without fine-tuning the pre-trained encoder.

with feature cosine similarity below 0.1 and report the ASR of the purified triggered graphs under Scenario 1.

As shown in Table 2, GFM-BA maintains high clean accuracy for its ability to leverage latent backdoor logic in the pre-trained GNN without the need to modify model parameters. Moreover, GFM-BA consistently achieves the highest ASR across all target GFMs and datasets after graph purification, outperforming baselines by an average of 36.81%, 19.98%, and 36.73% on GCOPE, MDGPT, and SAMGPT, respectively, demonstrating its strong stealthiness.

5.4 Attack Persistence Evaluation

The effectiveness of a backdoor attack typically depends on the parameters of the pre-trained model. However, in practical scenarios, downstream users may fine-tune the pre-trained GNN encoder, which can break the association between the trigger and the target embedding. To evaluate the persistence of GFM-BA, we conduct experiments on the Cora, Photo, and Computers datasets, targeting SAMGPT and MDGPT as victim models. The pre-trained GNN encoder is fine-tuned using a learning rate of 0.001, and we report the ASR under Scenario 1 described in Section 5.2, keeping all other experimental settings unchanged. As shown in Table 2, baseline methods exhibit a performance drop after fine-tuning the pre-trained encoder, whereas GFM-BA maintains a high ASR. This suggests that the backdoor logic of our GFM-BA is deeply embedded and resistant to forgetting during downstream adaptation.

5.5 Ablation Study

To evaluate the contribution of our designed label-free trigger association module, node-adaptive trigger generator, and persistent trigger anchoring module in GFM-BA, we conduct ablation studies using three variants: (1) GFM-BA(w/o E): We randomly set a target embedding under the target-controlled scenario to assess the impact of removing the label-free trigger association module. (2) GFM-BA(w/o S): A static trigger is used for all nodes under the graph purification defense in the target-uncontrolled scenario to evaluate the role of the node-adaptive trigger generator. (3) GFM-BA(w/o P): The persistence loss is removed, and the pre-trained encoder is fine-tuned to examine persistence in the absence of the persistent trigger anchoring module. Results on the Photo and Computers datasets with SAMGPT serv-
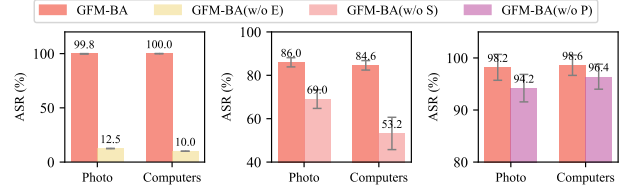

Figure 4: Results of ablation studies on Photo and Computers. GFM-BA(w/o E) is evaluated in the target-controlled scenario to assess effectiveness without the label-free trigger association module. GFM-BA(w/o S) is tested in the target-uncontrolled scenario with graph purification to assess stealthiness. GFM-BA(w/o P) is tested in the target-uncontrolled scenario with a fine-tuned backbone to assess the backdoor persistence.
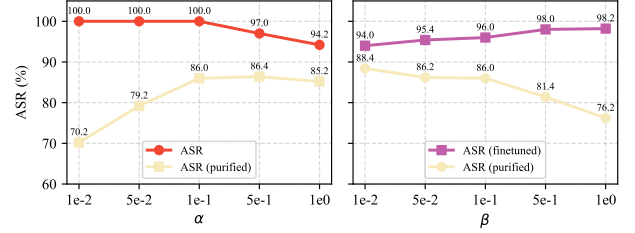

Figure 5: Hyperparameter study on the Photo dataset.

ing as the victim GFM are shown in Figure 4, while other datasets and victim models show similar patterns. As observed, the three components contribute to the effectiveness, stealthiness, and persistence of attacks, respectively.

5.6 Hyperparameter Study

In this section, we analyze the sensitivity of GFM-BA to hyperparameters α and β. For α, we examine its impact on the ASR of both the original triggered graphs and the purified triggered graphs, reflecting the trade-off between attack effectiveness and stealthiness. For β, we evaluate the ASR after fine-tuning the pre-trained encoder as well as under purification, assessing its influence on the persistence of the backdoor. Results on the Photo dataset under the target-uncontrolled scenario with SAMGPT as the victim are shown in Figure 5.

6 Conclusion

We propose GFM-BA, a novel backdoor attack model against Graph Foundation Models during the pre-training stage, designed to ensure effectiveness, stealthiness, and persistence. To perform effective backdoor when downstream tasks are unknown during pre-training, we introduce a label-free trigger association module that associates the trigger with a set of prototype embeddings selected via FPS. To enhance stealthiness, we design a node-adaptive trigger generator that produces trigger features close to the target node. For persistence, we anchor the trigger to parameters that are less sensitive to fine-tuning. Extensive experiments against representative victim GFMs demonstrate that GFM-BA consistently outperforms existing methods across all objectives.

Acknowledgements

References

Bongini, P.; Bianchini, M.; and Scarselli, F. 2021. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450: 242–252.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PmLR.

Cheng, D.; Wang, X.; Zhang, Y.; and Zhang, L. 2020. Graph neural network for fraud detection via spatial-temporal attention. *TKDE*, 34(8): 3800–3813.

Cheng, P.; Wu, Z.; Du, W.; Zhao, H.; Lu, W.; and Liu, G. 2025. Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. *TNNLS*.

Dai, E.; Lin, M.; Zhang, X.; and Wang, S. 2023. Unnoticeable backdoor attacks on graph neural networks. In *WWW*, 2263–2273.

Eldar, Y.; Lindenbaum, M.; Porat, M.; and Zeevi, Y. Y. 1997. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9): 1305–1315.

Fang, T.; Zhang, Y.; Yang, Y.; Wang, C.; and Chen, L. 2023. Universal prompt tuning for graph neural networks. *NeurIPS*, 36: 52464–52489.

Feng, B.; Jin, D.; Wang, X.; Cheng, F.; and Guo, S. 2024. Backdoor attacks on unsupervised graph representation learning. *Neural Networks*, 180: 106668.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Gu, N.; Fu, P.; Liu, X.; Liu, Z.; Lin, Z.; and Wang, W. 2023. A gradient control method for backdoor attacks on parameter-efficient tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 3508–3520.

Hassani, K. 2022. Cross-domain few-shot graph classification. In *AAAI*, volume 36, 6856–6864.

He, Y.; and Hooi, B. 2024. Unigraph: Learning a cross-domain graph foundation model from natural language. *arXiv e-prints*, arXiv–2402.

Idrissi, A.; Rehioui, H.; Laghrissi, A.; and Retal, S. 2015. An improvement of DENCLUE algorithm for the data clustering. In *ICTA*, 1–6. IEEE.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *KDD*, 66–74.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Kwon, H.; and Kim, D.-J. 2025. Dual-Targeted adversarial example in evasion attack on graph neural networks. *Scientific Reports*, 15(1): 3912.

Lachi, D.; Azabou, M.; Arora, V.; and Dyer, E. 2024. Graphfm: A scalable framework for multi-graph pretraining. *arXiv preprint arXiv:2407.11907*.

Lang, I.; Manor, A.; and Avidan, S. 2020. Samplenet: Differentiable point cloud sampling. In *CVPR*, 7578–7588.

Ling, H.; Jiang, Z.; Liu, M.; Ji, S.; and Zou, N. 2023. Graph mixup with soft alignments. In *ICML*, 21335–21349. PMLR.

Liu, J.; Yang, C.; Lu, Z.; Chen, J.; Li, Y.; Zhang, M.; Bai, T.; Fang, Y.; Sun, L.; Yu, P. S.; et al. 2025. Towards graph foundation models: A survey and beyond. *TPAMI*.

Luo, J.; Sun, Q.; Yuan, H.; Fu, X.; and Li, J. 2025. Robust Graph Learning Against Adversarial Evasion Attacks via Prior-Free Diffusion-Based Structure Purification. In *Proceedings of the ACM on Web Conference 2025*, 2098–2110.

Lyu, X.; Han, Y.; Wang, W.; Qian, H.; Tsang, I.; and Zhang, X. 2024. Cross-context backdoor attacks against graph prompt learning. In *KDD*, 2094–2105.

Mao, H.; Chen, Z.; Tang, W.; Zhao, J.; Ma, Y.; Zhao, T.; Shah, N.; Galkin, M.; and Tang, J. 2024. Position: Graph foundation models are already here. In *ICML*.

Molchanov, P.; Mallya, A.; Tyree, S.; Frosio, I.; and Kautz, J. 2019. Importance estimation for neural network pruning. In *CVPR*, 11264–11272.

Morcos, A. 2022. Understanding Contrastive versus Reconstructive Self-supervised Learning of Vision Transformers. In *NeurIPS Workshop on Self-Supervised Learning*.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *NeurIPS*.

Shi, C.; Chen, J.; Liu, J.; and Yang, C. 2024a. Graph foundation model. *Frontiers of Computer Science*, 18(6): 186355.

Shi, C.; Yang, C.; Fang, Y.; Sun, L.; and Yu, P. S. 2024b. Lecture-style Tutorial: Towards Graph Foundation Models. In *WWW 2024*, 1264–1267.

Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Yu, P. S.; He, L.; and Li, B. 2022a. Adversarial attack and defense on graph data: A survey. *TKDE*, 35(8): 7693–7711.

Sun, M.; Zhou, K.; He, X.; Wang, Y.; and Wang, X. 2022b. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *KDD*, 1717–1727.

Sun, X.; Cheng, H.; Li, J.; Liu, B.; and Guan, J. 2023. All in one: Multi-task prompting for graph neural networks. In *KDD*, 2120–2131.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024a. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*, 491–500.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Xia, L.; Yin, D.; and Huang, C. 2024b. Higpt: Heterogeneous graph language model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2842–2853.

Wang, K.; Deng, H.; Xu, Y.; Liu, Z.; and Fang, Y. 2024a. Multi-target label backdoor attacks on graph neural networks. *Pattern Recognition*, 152: 110449.

Wang, Z.; Zhang, Z.; Chawla, N.; Zhang, C.; and Ye, Y. 2024b. Gft: Graph foundation model with transferable tree vocabulary. *NeurIPS*, 37: 107403–107443.

Xi, Z.; Pang, R.; Ji, S.; and Wang, T. 2021. Graph backdoor. In *30th USENIX security symposium (USENIX Security 21)*, 1523–1540.

Xia, L.; Kao, B.; and Huang, C. 2024. OpenGraph: Towards Open Graph Foundation Models. In *EMNLP*, 2365–2379.

Xu, J.; and Picek, S. 2022. Poster: Clean-label backdoor attack on graph neural networks. In *CCS*, 3491–3493.

Xu, J.; Xue, M.; and Picek, S. 2021. Explainability-based backdoor attacks against graph neural networks. In *Proceedings of the 3rd ACM workshop on wireless security and machine learning*, 31–36.

Yang, X.; Li, G.; and Li, J. 2024. Graph neural backdoor: fundamentals, methodologies, applications, and future directions. *arXiv preprint arXiv:2406.10573*.

Yang, Y.; Li, Q.; Jia, J.; Hong, Y.; and Wang, B. 2024. Distributed backdoor attacks on federated graph learning and certified defenses. In *CCS*, 2829–2843.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 40–48. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *NeurIPS*, 33: 5812–5823.

Yu, X.; Gong, Z.; Zhou, C.; Fang, Y.; and Zhang, H. 2025. SAMGPT: Text-free Graph Foundation Model for Multi-domain Pre-training and Cross-domain Adaptation. In *WWW*, 1142–1153.

Yu, X.; Zhou, C.; Fang, Y.; and Zhang, X. 2024. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*.

Zhang, H.; Chen, J.; Lin, L.; Jia, J.; and Wu, D. 2023. Graph contrastive backdoor attacks. In *ICML*, 40888–40910. PMLR.

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. *NeurIPS*, 31.

Zhang, Z.; Jia, J.; Wang, B.; and Gong, N. Z. 2021. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM symposium on access control models and technologies*, 15–26.

Zhang, Z.; Lin, M.; Dai, E.; and Wang, S. 2024. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *KDD*, 4386–4397.

Zhao, H.; Chen, A.; Sun, X.; Cheng, H.; and Li, J. 2024. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *KDD*, 4443–4454.

Zheng, H.; Xiong, H.; Chen, J.; Ma, H.; and Huang, G. 2023. Motif-backdoor: Rethinking the backdoor attack on graph neural networks via motifs. *IEEE Transactions on Computational Social Systems*, 11(2): 2479–2493.

Zi, C.; Zhao, H.; Sun, X.; Lin, Y.; Cheng, H.; and Li, J. 2024. Prog: A graph prompt learning benchmark. *NeurIPS*.

Zou, D.; and Gu, Q. 2019. An improved analysis of training over-parameterized deep neural networks. *NeurIPS*, 32.