Surrogate as Teacher: Distillation-Guided Graph Poisoning Attack

Xingyu Peng[1,2], **Ke Xu**[1,2*]

[1]State Key Laboratory of Complex & Critical Software Environment (CCSE), Beihang University
[2]Zhongguancun Laboratory
{xypeng, kexu}@buaa.edu.cn

Abstract

While leveraging pseudo-labels has become a common paradigm in untargeted gray-box graph poisoning attacks, it suffers from two critical limitations: the use of brittle hard pseudo-labels that overlook uncertainty and can amplify surrogate model errors, and static guidance that progressively becomes stale as the graph is perturbed. To resolve these issues, we propose MetaDist, a novel framework that reframes the attack as an adversarial self-knowledge distillation process. Here, a "teacher" model provides continuously refined soft pseudo-labels to a "student" model, with the attack objective being to maximize the divergence between them. MetaDist makes two synergistic innovations. It employs the Reverse KL (RKL) divergence as a more strategic attack loss that efficiently converts uncertain nodes into robust, high-confidence errors. Concurrently, it introduces the Online Adaptive Teacher (OAT) mechanism, which adapts the teacher via student feedback to ensure the guidance signal remains relevant. Extensive experiments demonstrate that MetaDist consistently and significantly outperforms strong baselines across multiple datasets, proving its effectiveness and transferability even against advanced graph defenses.

Code — https://github.com/Crysta1ovo/MetaDist

Introduction

Graph Neural Networks (GNNs) have become a dominant tool for learning on graph-structured data (Wu et al. 2020; Zhou et al. 2020), achieving remarkable success in domains such as social network analysis (Fan et al. 2019), recommendation systems (Wu et al. 2019), drug discovery (Bongini, Bianchini, and Scarselli 2021), and financial fraud detection (Wang et al. 2019). However, the increasing deployment of GNNs in these security-critical applications has drawn significant attention to their vulnerabilities (Sun et al. 2022). In particular, GNNs have been shown to be highly susceptible to poisoning attacks (Zügner, Akbarnejad, and Günnemann 2018), where an attacker injects malicious but inconspicuous perturbations into the graph before a model is trained, aiming to degrade its performance at inference time.

Among various threat models, the untargeted gray-box setting stands out as particularly realistic and challenging. In this scenario, an attacker with partial model knowledge seeks to degrade the model's overall accuracy. The dominant paradigm for this task is the meta-gradient-based attack (Zügner and Günnemann 2019), which employs a surrogate model to compute meta-gradients that iteratively guide the search for effective perturbations. To achieve a more global impact, these attacks typically formulate their objective using pseudo-labels on unlabeled nodes, extending adversarial influence from the sparse labeled set to the entire graph.

Despite their effectiveness, these prevailing approaches face two fundamental limitations. First, they rely on brittle hard pseudo-labels. By assigning a single, definitive class to each node, this rigid guidance overlooks the surrogate's predictive uncertainty and can amplify initial labeling errors. Second, and more subtly, this guidance is static. Pseudo-labels are generated only once from a surrogate trained on the original, clean graph. As the attack proceeds and the graph evolves, this supervisory signal becomes progressively stale and ill-suited to the dynamic adversarial landscape, creating a significant bottleneck for attack efficacy.

To address these compounding challenges, we propose MetaDist, a novel framework that recasts graph poisoning as an adversarial self-knowledge distillation process. This paradigm shift immediately resolves the first limitation by replacing brittle hard pseudo-labels with more expressive soft pseudo-labels. In our framework, these soft labels are provided by a "teacher" model (a surrogate trained on the clean graph) to guide a "student" model (another surrogate trained on the progressively perturbed graph). The attack aims to identify perturbations that maximize the divergence between their predictions. The success of this new framework, however, hinges on two further innovations.

First, to maximize soft-label distillation's potential, we employ the Reverse KL (RKL) divergence as a more strategic attack loss. Our theoretical and empirical analyses reveal that conventional losses are suboptimal, as they inefficiently target already misclassified nodes ("lost causes"). In contrast, RKL's unique gradient profile concentrates attack pressure on the most uncertain nodes, generating a significantly stronger gradient signal. This allows the attack to efficiently create robust, high-confidence misclassifications. Second, to solve the problem of stale guidance, we introduce the Online Adaptive Teacher (OAT) mechanism. OAT creates a novel feedback loop where the teacher's knowledge is

*Corresponding author.

continuously refined using gradients from the student. This enables the teacher to adapt to the evolving perturbed graph, ensuring its soft-label guidance remains potent and relevant throughout the attack.

Extensive experiments on four benchmark datasets confirm that MetaDist decisively outperforms a comprehensive suite of advanced baselines, establishing a new state-of-the-art in all settings, even against robust GNN defense models. Our main contributions are summarized as follows:

- We propose MetaDist, a novel and powerful adversarial self-knowledge distillation framework for graph poisoning, leveraging continuously refined soft pseudo-labels.
- We are the first to identify and theoretically/empirically analyze the strategic advantages of RKL divergence for generating high-quality, robust poisoning attacks.
- We design the OAT mechanism, the first approach to successfully address the stale guidance problem in iterative, meta-gradient-based graph attacks.

Related Work

The vulnerability of Graph Neural Networks (GNNs) to adversarial attacks is a well-established field. These attacks are broadly categorized by their timing into evasion (Wang et al. 2024; Li et al. 2024) and poisoning attacks (Bojchevski and Günnemann 2019; Zügner, Akbarnejad, and Günnemann 2018), and by their goal into targeted (Dai et al. 2018; Chen et al. 2018) and untargeted attacks (Waniek et al. 2018; Ma, Ding, and Mei 2020). Our work specifically focuses on the challenging untargeted, gray-box poisoning setting.

The dominant paradigm for this task is the meta-gradient-based attack, pioneered by MetaAttack (Zügner and Günnemann 2019). This approach iteratively perturbs the graph by computing meta-gradients from a surrogate model, often guided by pseudo-labels. Subsequent works have enhanced this framework from various angles. Some methods focus on improving the surrogate model's fidelity or stabilizing the gradients, such as SRLIM (Liu et al. 2022b), which preserves topology using isometric mappings, or AtkSE (Liu et al. 2022a), which employs momentum ensembling. Other research has focused on perturbation selection strategies, for instance, EpoAtk (Lin et al. 2020) uses an exploratory approach to sidestep misleading gradients. A third line of work has refined the loss function: GraD (Liu et al. 2023) introduced a loss to better target uncertain nodes, and Metacon (Yoon et al. 2024) added a contrastive loss to correct sampling biases.

Preliminaries

Notations

We formally represent an attributed graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, and $X \in \mathbb{R}^{N \times d}$ is the node feature matrix. Each row $X_i \in \mathbb{R}^d$ corresponds to the feature vector of node v_i. The graph's topology can also be described by its adjacency matrix $A \in \{0, 1\}^{N \times N}$, where $A_{ij} = 1$ if an edge exists between nodes v_i and v_j, and $A_{ij} = 0$ otherwise. Thus, the attributed graph is concisely represented as $\mathcal{G} = (A, X)$.

In the semi-supervised learning setting, the node set $\mathcal{V}$ is partitioned into a labeled subset $\mathcal{V}_L$ and an unlabeled subset $\mathcal{V}_U$, such that $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_U$ and $\mathcal{V}_L \cap \mathcal{V}_U = \emptyset$. Typically, only a small fraction of nodes are labeled, i.e., $|\mathcal{V}_L| \ll |\mathcal{V}_U|$.

Attack Setting

This work investigates an **untargeted poisoning attack** under a **gray-box** setting, a widely studied and practically relevant scenario in graph adversarial learning. The attacker aims to degrade the overall performance of a graph learning model by perturbing the graph before it is trained. The gray-box assumption implies that the attacker possesses partial knowledge of the victim model, typically its architecture and training data, but lacks access to its trained parameters. Consistent with prior work, our study focuses on attacking semi-supervised node classification by manipulating only the graph structure (i.e., adding or removing edges), leaving node features unchanged.

Attack Formulation

To facilitate the attack, a surrogate model f_θ is constructed to approximate the victim's behavior. Typically, f_θ is instantiated as a Graph Convolutional Network (GCN) (Kipf 2016), with each layer's forward propagation defined as:

$$H^{(0)} = X, \quad H^{(k+1)} = \sigma\left(\hat{A} H^{(k)} W^{(k)}\right), \tag{1}$$

where $\hat{A}$ denotes the normalized adjacency matrix with added self-loops, $W^{(k)}$ is the trainable weight matrix for layer k, and $\sigma(\cdot)$ is a non-linear activation function. After K layers, the final output logits are $Z = H^{(K)}$. Thus, we define the model's output as $f_\theta(A, X) = Z$.

The attacker's goal is to find a perturbed adjacency matrix A', subject to a perturbation budget Δ, that maximizes an attack loss $\mathcal{L}_{\text{atk}}$. This objective is formulated as a bilevel optimization problem:

$$\begin{aligned} \max_{A'} \quad & \mathcal{L}_{\text{atk}}\left(f_{\theta^*}(A', X)\right) \\ \text{s.t.} \quad & \theta^* = \arg\min_{\theta} \mathcal{L}_{\text{train}}\left(f_\theta(A', X), Y_L\right), \\ & \|A - A'\|_0 \leq 2\Delta. \end{aligned} \tag{2}$$

The inner optimization finds the optimal parameters θ^* by training the surrogate model on the perturbed graph $\mathcal{G}' = (A', X)$ using a standard training loss $\mathcal{L}_{\text{train}}$ (e.g., cross-entropy over labeled nodes $\mathcal{V}_L$). The outer optimization then seeks the perturbed adjacency matrix A' that maximizes $\mathcal{L}_{\text{atk}}$ for the resulting trained model f_{θ^*}. Since the graph is undirected, the perturbation budget is expressed as 2Δ to account for the symmetry of the adjacency matrix.

Directly solving this bilevel problem is computationally intractable. To make it feasible, methods like MetaAttack approximate the solution by treating the adjacency matrix as a trainable hyperparameter. This allows computing a **meta-gradient**: the gradient of the attack loss with respect to the graph structure, indicating how infinitesimal edge changes affect the final attack loss after model retraining.

Since graph structure is inherently discrete, this meta-gradient information is used to greedily select the most damaging edge perturbations. The attack proceeds as an iterative, two-step process:

1. **Inner-loop update:** Retrain the surrogate model on the current perturbed graph, $\mathcal{G}^{(t)} = (A^{(t)}, X)$, at iteration t.
2. **Outer-loop update:** Compute the meta-gradient of the attack loss with respect to $A^{(t)}$. Then, identify and apply the single edge modification (adding a non-existent edge with the most positive gradient or removing an existing one with the most negative) that yields the largest estimated increase in loss, creating $A^{(t+1)}$.

This alternating process of model retraining and edge perturbation repeats until the perturbation budget is exhausted.

Methodology

In this section, we introduce **MetaDist**, our proposed framework that reformulates the untargeted gray-box poisoning attack as an iterative adversarial self-knowledge distillation process. The overall framework consists of four key phases per iteration, as illustrated in Figure 1.

Revisiting Attack Loss Design

In meta-gradient-based attacks, the attack loss function's design is pivotal, as it directly guides the perturbation process. The most straightforward strategy is to align the attack loss with the training objective:

$$\mathcal{L}_{\text{atk}} = -\frac{1}{|\mathcal{V}_L|} \sum_{i \in \mathcal{V}_L} \sum_{c=1}^{C} y_{i,c} \log\left(q_{i,c}\right), \tag{3}$$

where $y_{i,c}$ is the one-hot ground-truth label, and $q_{i,c}$ denotes the predicted probability for node v_i belonging to class c.

To enhance attack effectiveness, MetaAttack (Zügner and Günnemann 2019) incorporates a self-training mechanism. It first trains a surrogate model on the clean graph to generate **pseudo-labels** $\hat{Y}_U$ for the unlabeled nodes $\mathcal{V}_U$. The attack objective is then redefined over this broader set:

$$\mathcal{L}_{\text{atk}} = -\frac{1}{|\mathcal{V}_U|} \sum_{i \in \mathcal{V}_U} \sum_{c=1}^{C} \hat{y}_{i,c} \log\left(q_{i,c}\right). \tag{4}$$

This enables the attack to leverage additional supervision and extend its influence beyond the limited labeled nodes.

However, this formulation exhibits a critical limitation, as identified by GraD (Liu et al. 2023). Cross-entropy-based losses tend to assign large gradients to nodes that are already confidently misclassified. Consequently, their surrounding edges receive disproportionately large meta-gradient values, causing the attack to waste its limited budget on perturbations yielding little additional impact. To mitigate this issue, GraD introduces the Negative Probability (NP) loss:

$$\mathcal{L}_{\text{atk}} = -\frac{1}{|\mathcal{V}_U|} \sum_{i \in \mathcal{V}_U} \sum_{c=1}^{C} \hat{y}_{i,c}\, q_{i,c}, \tag{5}$$

which replaces the logarithmic term with the raw probability $q_{i,c}$. This adjustment reduces the emphasis on confidently misclassified nodes and promotes a more efficient perturbation allocation.

Adversarial Self-Knowledge Distillation

While pseudo-labeling broadens the attack's scope, existing methods typically use hard pseudo-labels, assigning each unlabeled node to a single class. This rigid assignment can amplify errors from the surrogate model and overlook predictive uncertainty. Instead, we argue that **soft pseudo-labels**, complete probability distributions over classes, provide a more expressive and robust supervision signal, especially for ambiguous nodes.

Building on this insight, we formulate the attack as an **adversarial self-knowledge distillation** process. A surrogate model trained on the clean graph acts as the teacher, producing soft pseudo-labels P, while another surrogate trained on the perturbed graph serves as the student, generating predictions Q. The attack objective is to find graph perturbations that maximize the divergence between the student's output Q and the teacher's guidance P over unlabeled nodes. Since the teacher's predictions represent the desired model behavior on the clean graph, maximizing this divergence inherently guides perturbations that force the student model to learn incorrect patterns, thereby degrading its performance.

For the divergence measure, we innovatively adopt the **Reverse Kullback-Leibler (RKL)** divergence over the conventional KL divergence:

$$\mathcal{L}_{\text{atk}} = D_{\text{KL}}(Q \| P) = \frac{1}{|\mathcal{V}_U|} \sum_{i \in \mathcal{V}_U} \sum_{c=1}^{C} q_{i,c} \log\left(\frac{q_{i,c}}{p_{i,c}}\right). \tag{6}$$

This choice is motivated by our observation that standard KL loss suffers from similar inefficiencies in budget allocation as cross-entropy. To formalize this, we decompose the meta-gradient with respect to an edge A_{uv} using the chain rule:

$$\frac{\partial \mathcal{L}_{\text{atk}}}{\partial A_{uv}} = \sum_{i=1}^{N} \sum_{c=1}^{C} \frac{\partial \mathcal{L}_{\text{atk}}}{\partial z_{i,c}} \cdot \frac{\partial z_{i,c}}{\partial A_{uv}}. \tag{7}$$

The term $\frac{\partial z_{i,c}}{\partial A_{uv}}$ depends solely on the GNN architecture and is nonzero only when edge A_{uv} lies within node v_i's receptive field. In contrast, $\frac{\partial \mathcal{L}_{\text{atk}}}{\partial z_{i,c}}$ captures how the chosen loss affects the gradient signal. Since the attack prioritizes modifications with the largest gradient magnitudes, our analysis centers on the ℓ_1 norm of this loss-dependent component:

$$\left\|\nabla_{z_i} \mathcal{L}_{\text{atk}}\right\|_1 = \sum_{c=1}^{C} \left|\frac{\partial \mathcal{L}_{\text{atk}}}{\partial z_{i,c}}\right|, \tag{8}$$

which quantifies the total absolute gradient contribution of node v_i, serving as an effective proxy for its influence in a magnitude-driven attack.

We now compare the gradient behaviors induced by different loss functions. For the cross-entropy loss, the logit gradient norm takes the form:

$$\left\|\nabla_{z_i} \mathcal{L}_{\text{atk-CE}}\right\|_1 = 2\left(1 - q_{i,t^*}\right), \tag{9}$$

where q_{i,t^*} is the predicted probability for the correct class t^*. This norm is maximized as $q_{i,t^*} \to 0$, which causes the attack to waste its budget on nodes that are already confidently misclassified.

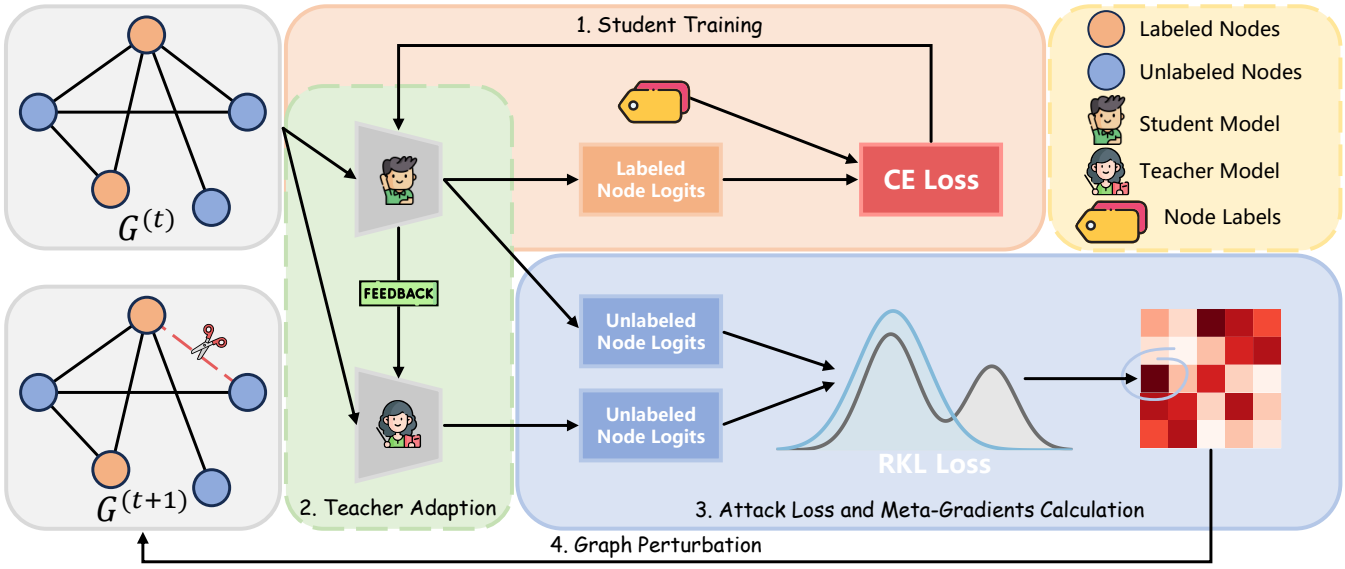

Figure 1: An overview of our proposed MetaDist framework, illustrating one full attack iteration. The process begins with a teacher model pre-trained on the clean graph. Each iteration then consists of four phases: (1) The student model is trained on the current perturbed graph $\mathcal{G}^{(t)}$ using a standard cross-entropy (CE) loss on labeled nodes. (2) The teacher model is adapted via our Online Adaptive Teacher (OAT) mechanism, using feedback (gradients) from the trained student. (3) The adapted teacher and student predict on unlabeled nodes; their Reverse KL (RKL) divergence is computed to generate meta-gradients. (4) These meta-gradients guide the selection of the most damaging edge perturbation, updating the graph to $\mathcal{G}^{(t+1)}$ for the next iteration.

A similar inefficiency arises with the KL loss. Its gradient norm is proportional to the distributional difference:

$$\|\nabla_{z_i}\mathcal{L}_{\text{atk-KL}}\|_1 \propto \sum_{c=1}^{C} |q_{i,c} - p_{i,c}| \,. \tag{10}$$

To illustrate its behavior, let $t^* = \arg\max_c p_{i,c}$ denote the teacher's top prediction. When the student strongly disagrees with the teacher ($q_{i,t^*} \to 0$), the student must be confident in some other class $j \neq t^*$. Since the teacher's distribution P is typically sharp, its confidence in this incorrect class, $p_{i,j}$, will be near zero. This scenario maximizes the gradient norm, causing it to approach 2. Conversely, when the student's prediction aligns with the teacher's ($q_{i,t^*} \to 1$), distributions match, and the gradient norm vanishes. Therefore, like cross-entropy, standard KL divergence directs the attack's focus toward nodes that are already confidently misclassified, an inefficient strategy for a limited budget.

In contrast, the NP loss exhibits a more desirable gradient profile for adversarial purposes. Its gradient norm is:

$$\|\nabla_{z_i}\mathcal{L}_{\text{atk-NP}}\|_1 = 2q_{i,t^*}\left(1 - q_{i,t^*}\right), \tag{11}$$

which forms a parabolic curve that peaks when the student is most uncertain ($q_{i,t^*} = 0.5$) and vanishes when confident (either $q_{i,t^*} \to 0$ or 1). This non-monotonic property allows the attack to focus its budget on ambiguous nodes that are most susceptible to perturbation, avoiding wasteful allocations on confidently classified or misclassified nodes.

Our proposal used RKL loss retains this strategic focus on uncertain nodes but with superior properties. Its gradient norm is given by:

$$\|\nabla_{z_i}\mathcal{L}_{\text{atk-RKL}}\|_1 \propto \sum_{c=1}^{C} \left| q_{i,c} \left[\log\left(\frac{q_{i,c}}{p_{i,c}}\right) - D_{\text{KL}}(Q\|P) \right] \right| . \tag{12}$$

When the student agrees with the teacher ($q_{i,t^*} \to 1$), the log-ratio and the overall divergence $D_{\text{KL}}(Q\|P)$ both approach zero, nullifying the gradient. More importantly, when the student confidently disagrees ($q_{i,t^*} \to 0$ and $q_{i,j} \to 1$ for some other class j), the gradient also vanishes. This is because for any class c with low student confidence ($q_{i,c}$), its gradient contribution is suppressed by the $q_{i,c}$ pre-factor. For the single class j where the student is confident, the large log term is almost perfectly offset by the similarly large overall divergence term $D_{\text{KL}}(Q\|P)$. By suppressing the gradient at both ends of the confidence spectrum, RKL naturally concentrates its power on the most ambiguous nodes where the model is most impressionable.

RKL provides two critical advantages over NP loss. First, its logarithmic structure **produces stronger gradient signals in uncertain regions** compared to the NP loss, whose parabolic gradient is capped at a maximum of 0.5. This higher gradient magnitude yields a better signal-to-noise ratio, enabling the attack to more reliably and accurately select the optimal edge perturbation in each iteration.

Second, RKL's identity as a **mode-seeking** divergence (Bishop and Nasrabadi 2006) leads to higher-quality, more robust attacks. The NP loss's weaker signal may create fragile misclassifications (e.g., 60% confidence in a wrong class), which lie close to the decision boundary and are easily reversed. In contrast, RKL's objective is more aggres-

sive: it forces the student's probability distribution to form a new, sharp peak (a mode) at an incorrect class. This creates robust, high-confidence errors deeply embedded within the wrong class's decision region, which are far more difficult for any defense to correct.

In summary, RKL enables more aggressive and efficient perturbation in regions of uncertainty, perfectly balancing the depth of the attack (pursuing high-quality, robust outcomes) with its breadth (efficiently allocating the budget).

Online Adaptive Teacher

Pseudo-label-based graph poisoning attacks face a critical challenge: a growing gap between the knowledge of the "teacher" model, derived from the original, clean graph $\mathcal{G}$, and the "student" model, trained on the progressively perturbed graph $\mathcal{G}^{(t)}$. As the attack proceeds, the teacher's guidance (i.e., fixed pseudo-labels) becomes increasingly stale and ill-suited to the evolving adversarial landscape, creating a bottleneck for attack efficacy.

Naive solutions to this problem are ineffective. On one hand, fully retraining the teacher on the perturbed graph in each iteration would make it identical to the student, causing the adversarial signal to vanish. On the other hand, simply having the clean-trained teacher evaluate the heavily perturbed graph leads to a severe Out-of-Distribution (OOD) collapse, rendering its guidance noisy and unreliable.

To resolve this dilemma, we propose the **Online Adaptive Teacher (OAT)** framework, which formulates the attack as an online learning process over a stream of perturbed graphs. At each iteration t, after the student $f_{\theta_S^{(t)}}$ is trained on the current perturbed graph $\mathcal{G}^{(t)}$, the teacher adapts based on feedback from the student. Specifically, the teacher takes a single update step using the student's final training gradient:

$$\theta_T^{(t)} = \theta_T^{(t-1)} - \eta_T \nabla_{\theta_S} \mathcal{L}_{\text{train}} \left(f_{\theta_S^{(t)}} \left(A^{(t)}, X \right), Y_L \right). \quad (13)$$

This incremental update ensures the teacher adapts just enough to guide relevantly without mirroring the student.

The adapted teacher $f_{\theta_T^{(t)}}$ then generates a new set of target logits by making predictions on the same graph:

$$Z_T^{(t)} = f_{\theta_T^{(t)}} \left(A^{(t)}, X \right). \quad (14)$$

These logits are passed through a softmax function to yield the teacher's updated guidance $P^{(t)}$, which is subsequently used to compute the attack loss for selecting the next graph perturbation, thereby continuing the online process.

The OAT framework provides two significant advantages. First, the online adaptation process elegantly circumvents the OOD problem by having the teacher adjust to the current poisoned environment, ensuring its guidance remains potent. Simultaneously, because the update is minimal, the teacher retains knowledge from its previous states and remains distinct from the student, preserving a strong adversarial signal. Second, OAT inherently improves the transferability of the attack by mitigating the risk of overfitting to a single surrogate. The final poisoned graph is not optimized against a static teacher but is instead the result of optimization against an evolutionary sequence of models $\left(\left\{ (f_{\theta_T^{(0)}}, f_{\theta_T^{(1)}}, \dots \right\} \right)$. This process, analogous to temporal ensembling, compels the attack to discover fundamental vulnerabilities in the GNN's message-passing mechanism, rather than superficial quirks of a single model instance, thereby enhancing its effectiveness against unseen victim models.

Experiments

Experimental Setup

Datasets. We conduct experiments on four widely used benchmark datasets: *Cora* (McCallum et al. 2000), *CiteSeer* (Sen et al. 2008), *PolBlogs* (Adamic and Glance 2005), and *Cora-ML* (McCallum et al. 2000). Following standard practice (Zügner and Günnemann 2019), we use a semi-supervised split where 10% of nodes constitute the labeled training set, with the remaining 90% being unlabeled.

Baselines. We benchmark our method against a comprehensive suite of graph poisoning attacks, grouped into three categories: (i) simple heuristic baselines: **Random** and **DICE** (Waniek et al. 2018); (ii) PGD-based attacks (Xu et al. 2019): **PGD-CE** and **PGD-CW**, which utilize cross-entropy and Carlini-Wagner losses, respectively; and (iii) meta-gradient-based approaches, including **Meta-Train** (Zügner and Günnemann 2019), **Meta-Self** (Zügner and Günnemann 2019), **AtkSE** (Liu et al. 2022a), **GraD** (Liu et al. 2023), **Metacon-S** (Yoon et al. 2024), and **Metacon-D** (Yoon et al. 2024). All baseline methods are implemented using their official codebases for faithful reproduction.

Evaluation Protocol. We evaluate attack effectiveness by measuring the performance degradation of a victim model trained from scratch on the perturbed graph. Our primary victim model is a two-layer GCN. To assess transferability, we also employ a two-layer Graph Attention Network (GAT) (Veličković et al. 2018) as a second victim. To ensure robust results, each victim model is trained 10 times with different random seeds. We report the mean classification accuracy and standard deviation on the test set, where lower accuracy signifies a more effective attack.

Implementation Details. All attacks are performed with a perturbation budget fixed at 5% of the total number of edges. For methods that require a surrogate model, we employ a standard two-layer GCN architecture. To ensure fair comparison, hyperparameters for each attack method are configured based on the optimal settings reported in their respective papers. All experiments were implemented in PyTorch and ran on a single NVIDIA GeForce RTX 4090D GPU.

Main Results

The main experimental results, presented in Table 1, confirm MetaDist's superiority. It establishes a new state-of-the-art by achieving the best performance in all 8 experimental settings, consistently outperforming all baselines across every dataset and victim model.

Our analysis of baselines reveals a clear progression in attack sophistication. Simple heuristic and PGD-based methods show limited effectiveness, while meta-gradient-based

Method		Cora		CiteSeer		PolBlogs		Cora-ML	
		GCN	GAT	GCN	GAT	GCN	GAT	GCN	GAT
	Clean	82.75±0.26	84.35±0.58	72.14±0.45	73.25±0.73	95.24±0.48	95.40±0.32	85.62±0.28	85.55±0.42
Rand	Random	81.79±0.39	83.06±0.59	71.69±0.42	71.96±0.85	91.73±0.38	93.25±0.34	84.88±0.27	84.15±0.46
	DICE	82.36±0.35	83.27±0.48	71.58±0.39	72.33±0.67	88.99±0.52	90.58±0.27	84.50±0.35	84.07±0.35
PGD	PGD-CE	83.71±0.22	83.98±0.59	72.38±0.36	72.75±0.62	93.16±0.06	93.28±0.15	85.67±0.07	85.97±0.25
	PGD-CW	80.49±0.35	82.11±0.46	69.97±0.48	72.54±0.60	85.70±0.61	86.11±0.78	83.41±0.31	83.21±0.30
Meta	Meta-Train	78.54±0.45	82.61±0.42	69.41±0.39	72.68±0.60	87.69±0.58	93.43±0.98	82.40±0.30	83.41±0.45
	Meta-Self	75.22±0.48	80.62±0.58	68.34±0.34	73.09±0.76	77.59±0.54	88.24±1.67	80.60±0.30	82.25±0.49
	AtkSE	77.15±0.36	83.62±0.55	65.15±0.63	72.58±0.57	75.98±0.20	85.97±1.69	75.40±0.65	76.87±0.53
	GraD	69.50±0.67	79.67±0.55	60.46±0.83	71.61±0.81	74.40±0.18	85.44±1.57	79.88±0.34	82.46±0.99
	Metacon-S	76.77±0.46	79.68±0.73	67.26±0.35	69.73±0.82	74.02±0.57	82.47±1.30	78.58±0.25	80.46±0.68
	Metacon-D	76.88±0.35	79.62±0.38	65.38±0.58	67.88±0.95	76.42±0.52	84.83±1.55	78.83±0.18	80.95±0.62
	MetaDist (Ours)	**67.23±1.21**	**78.59±0.57**	**57.55±0.67**	**66.60±1.06**	**68.06±0.17**	**75.63±1.45**	**73.77±0.44**	**76.83±0.51**

Table 1: Main results for untargeted gray-box poisoning attacks on various datasets. We report the mean classification accuracy (%) and standard deviation over 10 runs for GCN and GAT victim models. Lower values indicate more effective attacks. The **best** and second-best results are highlighted in bold and underlined, respectively.

Method	Cora			PolBlogs		
	RGCN	ProGNN	SimPGCN	RGCN	ProGNN	SimPGCN
Clean	83.56±0.26	84.96±0.08	81.97±0.83	95.37±0.16	95.62±0.37	95.18±0.47
Meta-Train	79.16±0.45	83.14±0.02	78.82±1.12	89.67±0.39	94.89±0.81	85.48±1.74
Meta-Self	76.93±0.37	81.86±0.33	78.57±0.94	78.03±0.49	94.33±0.85	75.18±2.01
AtkSE	79.69±0.44	82.31±0.49	79.52±1.11	76.03±0.34	93.43±0.11	74.04±1.48
GraD	73.71±1.62	79.94±0.32	78.16±0.75	74.74±0.47	92.18±0.71	73.85±1.23
Metacon-S	77.69±0.54	83.87±0.13	79.46±0.38	74.49±0.29	90.53±0.77	72.22±2.16
Metacon-D	77.23±0.41	83.09±0.32	79.32±0.57	76.41±0.29	93.21±0.12	74.43±1.62
MetaDist	**71.78±0.74**	**78.67±0.30**	**76.95±2.34**	**68.37±0.08**	**85.61±0.40**	**67.86±0.53**

Table 2: Attack performance against representative GNN defense models on the Cora and PolBlogs datasets.

Method	Cora	CiteSeer	PolBlogs	Cora-ML
MetaDist	**67.23±1.21**	**57.55±0.67**	**68.06±0.17**	**73.77±0.44**
w/ KL Loss	74.95±0.29	67.35±0.34	74.99±0.09	78.58±0.35
w/o OAT	68.37±1.35	61.48±0.39	72.42±0.07	76.32±0.27
w/o T-Adapt	69.84±1.62	58.98±0.62	71.24±0.05	76.10±0.28

Table 3: Ablation study results on various datasets.

approaches yield substantial performance boosts. Within this paradigm, Meta-Self improves over Meta-Train by incorporating pseudo-labels. Further gains are achieved by GraD, which refines the loss function, and Metacon, which adds a contrastive objective.

Despite the strong performance of these advanced approaches, MetaDist surpasses them in every scenario, often by a substantial margin. For instance, on PolBlogs against a GCN victim, MetaDist achieves 68.06% accuracy, nearly a 6-point improvement over Metacon-S (74.02%). We attribute this consistent superiority to the synergy of our two core innovations: the RKL loss provides a more potent and strategic gradient signal, while the OAT mechanism ensures its relevant throughout the attack process.

Furthermore, MetaDist exhibits excellent transferability. As shown in Table 1, it maintains a large gap against baselines, even when the victim model (GAT) differs from the surrogate (GCN). This suggests the vulnerabilities discovered by MetaDist are intrinsic to the graph message-passing paradigm, making the attack more robust and generalizable.

Evaluation Against Robust Defenses

To assess our attack's potency, we evaluated it against three representative GNN defense models: **RGCN** (Zhu et al. 2019), **ProGNN** (Jin et al. 2020), and **SimPGCN** (Jin et al. 2021). As shown in Table 2, MetaDist consistently outperforms all baselines, demonstrating a superior ability to bypass these advanced defenses.

While all defenses mitigate baseline attacks to some degree, MetaDist still inflicts substantial performance degradation. For instance, against the ProGNN defense on PolBlogs, our method reduces accuracy to 85.61%, a drop of over 10 percentage points, whereas most baselines are rendered ineffective. This superior performance is attributed to MetaDist's core design: the RKL loss creates robust, high-confidence errors far from the decision boundary, while the OAT mechanism discovers fundamental, generalizable vulnerabilities. This combination generates perturbations that are more difficult for defenses to mitigate, confirming MetaDist poses a significant threat even to robust models.

Ablation Study

To assess the contributions of MetaDist's core components (the RKL loss and the OAT mechanism), we conduct an ablation study. The results, presented in Table 3, confirm that both are essential for achieving state-of-the-art performance.

First, we evaluate the impact of the loss function. Replacing the RKL loss with standard KL divergence (`w/ KL Loss`) leads to a substantial drop in attack effectiveness. For instance, victim accuracy on CiteSeer rises sharply from 57.55% to 67.35%, confirming RKL's strategic focus on uncertain nodes enables more effective gradient allocation.

Next, we examine the OAT mechanism. Removing it entirely (`w/o OAT`), reverting to a static teacher, consistently weakens the attack. More revealingly, disabling only the

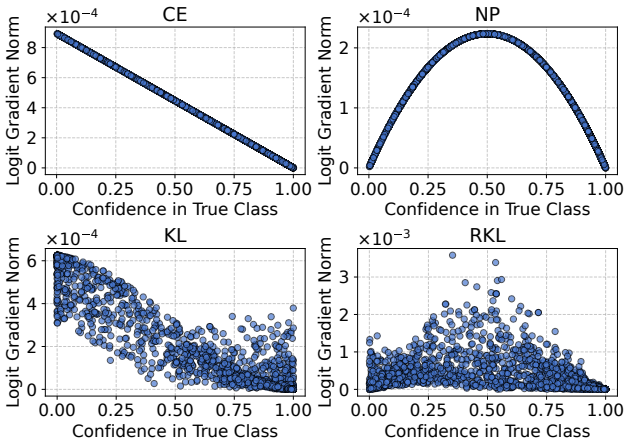

Figure 2: Empirical visualization of the ℓ_1 norm of the logit gradient as a function of model confidence for four losses on the Cora dataset (captured at 60% of the 5% budget).

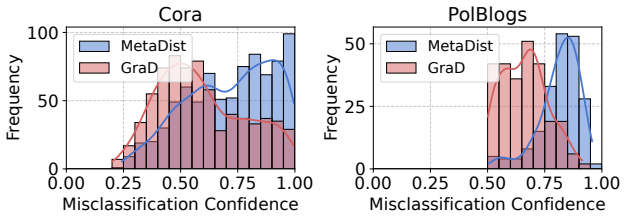

Figure 3: Confidence distributions of misclassified nodes for MetaDist and GraD on the Cora and PolBlogs datasets.

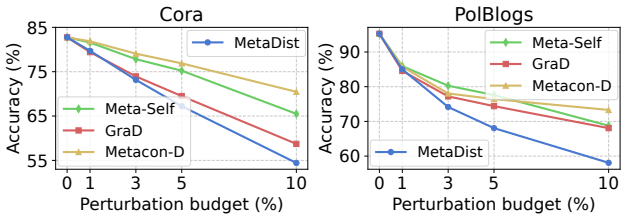

Figure 4: Performance comparison of MetaDist against competitive baselines under varying perturbation budgets on the Cora and PolBlogs datasets.

teacher's gradient-based adaptation (`w/o T-Adapt`) while retaining online re-evaluation yields inconsistent results, improving performance on some datasets while degrading it on Cora. This suggests that re-evaluation without adaptation can exacerbate the OOD problem. Therefore, the synergy of online re-evaluation with gradient-based adaptation is key to the OAT mechanism's robust effectiveness.

Visualizing the Gradient Dynamics

Figure 2 confirms our theoretical analysis by plotting the empirical gradient dynamics for each loss function during an attack on Cora. These visual profiles intuitively explain performance variations reported in Tables 1 and 3, showing each loss function's distinct gradient behavior closely aligns with its corresponding model's performance: CE to Meta-Self, NP to GraD, KL to our `w/ KL Loss` ablation, and RKL to our full MetaDist model.

As anticipated, the CE and KL losses (left column) exhibit monotonically decreasing gradients concentrated on low-confidence nodes. In stark contrast, NP and RKL (right column) focus on the ambiguous region around 0.5 confidence, exhibiting the desired parabolic-like shape. Crucially, RKL demonstrates a decisive advantage over NP: its gradient magnitudes in this region are over an order of magnitude larger. This provides strong visual evidence that RKL produces a far more potent optimization signal, which drives MetaDist's superior performance over GraD.

Analysis of Misclassification Quality

Figure 3 plots the confidence distribution of misclassified nodes to analyze the quality of the induced errors, revealing a stark contrast between the methods. The misclassifications from GraD are concentrated in the mid-confidence range (0.4–0.7), indicating fragile errors near the decision boundary. Conversely, the distribution for MetaDist is heavily skewed towards 1.0, demonstrating that it consistently forces highly confident incorrect predictions. This provides strong empirical evidence for the mode-seeking nature of our RKL-based objective, which prioritizes creating robust errors that are more damaging and difficult to reverse.

Effectiveness Across Different Budgets

To evaluate our attack's robustness and efficiency, we conducted a sensitivity analysis on the perturbation budget, with results presented in Figure 4. The plots demonstrate that MetaDist consistently outperforms all baselines across the full range of tested budgets (1% to 10%) on both datasets.

Notably, the performance gap between MetaDist and its competitors widens as the budget increases. This trend suggests MetaDist utilizes additional perturbations more efficiently. We attribute this growing advantage to the OAT mechanism, which prevents the guidance signal from becoming stale. As baseline attacks falter in heavily perturbed graphs, MetaDist's adaptive teacher remains effective, ensuring sustained performance at higher attack intensities.

Conclusion

We proposed MetaDist, a novel untargeted gray-box graph poisoning attack that achieves state-of-the-art performance. By reframing the attack as adversarial self-knowledge distillation, MetaDist addresses key limitations of prior methods via two core components: an RKL loss that strategically targets uncertain nodes to induce confident misclassifications, and an OAT mechanism that refines supervision through student feedback, resolving the stale guidance issue.

Our extensive experiments across multiple benchmarks demonstrate that MetaDist consistently outperforms strong baselines and remains effective against advanced GNN defenses. These results highlight its generalizability and underscore the need for more robust graph defense strategies.

Acknowledgments

This work has been supported by CCSE project (CCSE-2024ZX-09) and Zhongguancun Laboratory.

References

Adamic, L. A.; and Glance, N. 2005. The political blogosphere and the 2004 US election: divided they blog. In *LinkKDD*, 36–43.

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Bojchevski, A.; and Günnemann, S. 2019. Adversarial attacks on node embeddings via graph poisoning. In *ICML*, 695–704. PMLR.

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

Chen, J.; Wu, Y.; Xu, X.; Chen, Y.; Zheng, H.; and Xuan, Q. 2018. Fast gradient attack on network embedding. *arXiv preprint arXiv:1809.02797*.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial attack on graph structured data. In *ICML*, 1115–1124. PMLR.

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *WWW*, 417–426.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node similarity preserving graph convolutional networks. In *WSDM*, 148–156.

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. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

Li, K.; Chen, Y.; Liu, Y.; Wang, J.; He, Q.; Cheng, M.; and Ao, X. 2024. Boosting the adversarial robustness of graph neural networks: An ood perspective. In *ICLR*.

Lin, X.; Zhou, C.; Yang, H.; Wu, J.; Wang, H.; Cao, Y.; and Wang, B. 2020. Exploratory adversarial attacks on graph neural networks. In *ICDM*, 1136–1141. IEEE.

Liu, Z.; Luo, Y.; Wu, L.; Li, S.; Liu, Z.; and Li, S. Z. 2022a. Are gradients on graph structure reliable in gray-box attacks? In *CIKM*, 1360–1368.

Liu, Z.; Luo, Y.; Wu, L.; Liu, Z.; and Li, S. Z. 2023. Towards reasonable budget allocation in untargeted graph structure attacks via gradient debias. *arXiv preprint arXiv:2304.00010*.

Liu, Z.; Luo, Y.; Zang, Z.; and Li, S. Z. 2022b. Surrogate representation learning with isometric mapping for gray-box graph adversarial attacks. In *WSDM*, 591–598.

Ma, J.; Ding, S.; and Mei, Q. 2020. Towards more practical adversarial attacks on graph neural networks. In *NeurIPS*, 4756–4766.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2): 127–163.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

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

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wang, B.; Lin, M.; Zhou, T.; Zhou, P.; Li, A.; Pang, M.; Li, H.; and Chen, Y. 2024. Efficient, direct, and restricted black-box graph evasion attacks to any-layer graph neural networks via influence function. In *WSDM*, 693–701.

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019. A semi-supervised graph attentive network for financial fraud detection. In *ICDM*, 598–607. IEEE.

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2): 139–147.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *AAAI*, volume 33, 346–353.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1): 4–24.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019. Topology attack and defense for graph neural networks: an optimization perspective. In *IJCAI*, 3961–3967.

Yoon, K.; In, Y.; Lee, N.; Kim, K.; and Park, C. 2024. Debiased Graph Poisoning Attack via Contrastive Surrogate Objective. In *CIKM*, 3012–3021.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1: 57–81.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *KDD*, 1399–1407.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *KDD*, 2847–2856.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *ICLR*.