Cracking Heterophily in Coarsening-Based Scalable GNN Training: a Coarsening Residual View

Guoming Li[1,2], Jian Yang[3], Xukun Wang[3], Yifan Chen[1§]
[1]Hong Kong Baptist University, [2]MBZUAI, [3]University of Chinese Academy of Sciences
paskardli@outlook.com, jianyang0227@gmail.com, wangxukun@amss.ac.cn, yifanc@hkbu.edu.hk

***Abstract*—[THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD] Scalability remains one of the major challenges for graph neural networks (GNNs) training, which sparks various methods to address this issue. Among these methods, a category of proposals utilizes graph coarsening and has gained notable attention; such methods train GNNs on the coarsened graphs rather than the original large graphs, which not only enhance training efficiency but also preserve efficacy compared with regular GNN training. However, existing works have focused on and evaluated solely with homophilic graphs, leaving the more demanding heterophilic graphs underexplored. In this paper, we identify the "residual" in graph coarsening and study its impact on heterophilic graphs, revealing in the heterophilic case the residual can drastically degrades the efficacy of coarsening-based methods. To tackle this critical issue of heterophily, we propose CTH, a novel coarsening-based GNN training framework. CTH clusters nodes with high similarity to coarsen graphs, then trains GNNs thereon, incorporating a post-compensation module that integrates the residual to improve training efficacy. Extensive experiments confirm CTH as a superior and practical solution for scalable GNN training, with particular advantages for heterophilic graphs. The code, as well as a complete manuscript with appendix, are provided with this link: https://anonymous.4open.science/r/Cracking-Heterophily-FD97/ [1], and scheduled to be open-sourced upon publication.**

I. Introduction

Graph neural networks (GNNs) [2] have emerged as powerful tools for capturing structural information from graph signals, achieving prominent performance across a broad spectrum of structured and graph signal processing tasks [3]–[7]. In real-world scenarios, graphs often contain hundreds of millions of nodes and billions of edges and provide rich information; however, complexity of mainstream graph propagation algorithms scales as well with the number of nodes and edges, resulting in intensive computational costs for GNN training.

Numerous methods have thus been introduced to reduce complexity in the model and / or the data aspect, yielding notable results. Among these methods, a category based on *graph coarsening* demonstrates prominent efficacy and has gained great attention. Such coarsening-based methods start by extracting a smaller coarsened graph from the original large graph using mature graph coarsening algorithms [8]–[14]; then they train the GNNs on this coarsened graph instead of the original one, with the objective of producing accurate label predictions for the original graphs. Recent studies [15]–[17] confirm the practicality of these methods, which not only bolster training efficiency but also attain performance comparable to that of GNNs trained on the full graph.

Despite the encouraging successes, existing proposals have concentrated and evaluated solely on homophilic graphs, leaving the more demanding *heterophilic* graphs relatively underexplored [18], [19]. Unlike conventional homophilic graphs, which operate under a strong assumption that neighboring nodes share similar node embeddings or labels, heterophilic graphs challenge this premise by featuring diverse labels among nodes in the same local neighborhood. This distinctive property presents critical challenges for graph learning and further complicates coarsening-based GNN training.

In this paper, we navigate coarsening-based GNN training on heterophilic graphs through a novel lens of *coarsening residual*. Under an information-theoretic framework that models GNN training as mutual information maximization, we reveal that, on heterophilic graphs, the coarsening-induced residuals severely degrade the training efficacy, presenting a pressing challenge for this paradigm. To tackle this challenge, we propose CTH (**C**oarsening-based **T**raining on **H**eterophilic graphs), an advanced two-stage coarsening-based GNN training framework. CTH initially generates a coarsened graph via clustering nodes based on the similarity of an augmented feature, which combines position embeddings and propagated features of the original graph. GNNs are then trained on this coarsened graph, along with a post-compensation module that integrates the residual to boost training performance. By evaluating CTH on heterophilic graph benchmarks, we demonstrate its superior performance compared to existing methods, underscoring its advantage.

Our contributions. *First*, we examine coarsening-based GNN training with in-depth analysis, uncovering how coarsening residuals induce a performance gap on heterophilic graphs, emphasizing the pressing challenge of graph heterophily. *Second*, we propose CTH, an advanced coarsening-based GNN training paradigm crafted to overcome the heterophily challenge. *Third*, we conduct extensive experiments to validate CTH as a superior and practical solution for scalable GNN training, with distinct benefits for heterophilic graphs, marking a key advancement in coarsening-based GNN training.

II. Background and Preliminaries

We denote $\mathcal{G} = (A, X)$ as a graph with adjacency matrix $A \in \{0,1\}^{n\times n}$ and node features / signals $X \in \mathbb{R}^{n\times d}$. Let $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ be the normalized graph Laplacian [20],

§Correspondence to Yifan Chen.

with I and D being the identity matrix and the degree matrix, respectively. $Y \in \{0,1\}^{n\times c}$ is the one-hot node label matrix.

A. Graph coarsening

The graph coarsening technique merges nodes to coarsen the original graph [8]–[14]. The essence of graph coarsening is to partition nodes into $n' < n$ clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{n'}$, where each of these clusters correspond to a *supernode*. Such partition is represented by a partition matrix $P \in \{0,1\}^{n'\times n}$, where entry $P_{i,j} = 1$ with node $v_j \in \mathcal{C}_i$. Specifically, the coarsened graph $\mathcal{G}' = (A', X')$ is given as $A' = PAP^T$, $X' = C^{-1}PX$, $Y' = C^{-1}PY$, where $C = \text{diag}(|\mathcal{C}_1|, |\mathcal{C}_2|, ..., |\mathcal{C}_{n'}|)$.

B. Graphs with heterophily

Recently, graphs exhibiting the *heterophily* property have recently emerged as a topic of considerable interest. In contrast to conventional graphs that exhibit homophily, where connected nodes typically share the same label and embeddings, graphs with heterophily tend to link nodes with different labels and embeddings. An illustrative example is provided in Figure 1, where panel (b) represents a homophilic graph and panel (c) illustrates a heterophilic graph. The heterophilic graphs have been shown in prior studies to present notable challenges in the graph learning domain, particularly in the context of GNNs [18], [19].

C. Scalable GNN training via coarsening

Recent research has explored the intuition that coarsening can serve as a solution for scalable training of GNN. They propose to train GNN on a coarsened graph rather than the original large graph, which improves training efficiency while maintaining efficacy of the trained model [15]–[17]. For instance, [15] derived graph convolution on coarsened graphs and proved the efficacy for GNN training; [17] proposed ConvMatch which aligns graph convolution from a spectral perspective. However, existing works have centered on homophilic graphs, leaving the heterophilic graphs unexplored.

III. Navigating Heterophily Challenges in Coarsening-based GNN Training

This section explores the challenges of heterophily in graphs for coarsening-based GNN training through a novel lens of *coarsening residual*. We first formally define the problem setting of coarsening-based GNN training and then provide an information-theoretic analysis to uncover how the coarsening-induced residual relates to heterophily challenges.

A. Problem formulation

Building upon the structured analysis in prior research, such as [15], the paradigm of coarsening-based GNN training can be formally defined as follows:

Definition 1. Let $\mathcal{G} = (A, X)$ be a graph with node label matrix Y, and its coarsened version be $\mathcal{G}' = (A', X')$ with coarsened label Y'. Let $f(;\Theta)$ denote a GNN parameterized by Θ. The predictions of the GNN for the original and coarsened graphs are $f(A, X; \Theta)$ and $f(A', X'; \Theta)$, respectively. Using the *cross-entropy loss* as the criterion, coarsening-based GNN training minimizes the loss on the original graph, $\mathcal{L}(f(A, X; \Theta), Y)$, by instead minimizing the loss on the coarsened graph, $\mathcal{L}(f(A', X'; \Theta), Y')$.

To simplify the analysis, we further assume that the node labels are balanced and follow a uniform distribution (this assumption can be generalized as the condition that all the category probabilities are bounded from below), stated as:

Assumption 1. The node label Y follows a uniform distribution, i.e., $\mathbb{E}_Y[y_{ij}] = 1/c$, for $1 \le i \le n$, $1 \le j \le c$.

B. An Coarsening residual view on heterophily challenges

1) Coarsening residual in graph coarsening: We first introduce a novel perspective on graph coarsening through the lens of the *coarsening residual*, formally defined below:

Definition 2 (coarsening residual)**.** Let $\mathcal{G}$ and $\mathcal{G}'$ be the original graph and its coarsened version, respectively. The coarsening residual introduced by the coarsening operation from graph $\mathcal{G}$ to $\mathcal{G}'$ is expressed as $\mathcal{G} \setminus \mathcal{G}' \triangleq (A - \Pi A \Pi, X - \Pi X)$. Here, Π is an orthogonal matrix induced by $P^T P$.

The concept of coarsening residual is inspired by prior research on graph coarsening, where the quality of coarsening algorithms is evaluated by measuring the discrepancies $A - \Pi A \Pi$ and $X - \Pi X$, with Π derived as $P^T P$ incorporating certain normalizations [9], [10], [12]. This $\mathcal{G} \setminus \mathcal{G}'$ directly captures the reconstruction error of the original graph $\mathcal{G}$ on top of the coarsened graph $\mathcal{G}'$ and partition matrix P; specifically, with $\mathcal{G}', \mathcal{G} \setminus \mathcal{G}'$, and P, we can reconstruct the original graph $\mathcal{G}$. Conceptually the spectral norm of $A - \Pi A \Pi$ and $X - \Pi X$ can quantify the loss of graph information during the coarsening: if $\mathcal{G}$ exhibits a prone-to-coarsening structure, $\|A - \Pi A \Pi\|$ and $\|(I - \Pi)X\|$ are supposed to be negligible.

For better clarity, an illustrative example of this representation is provided in Figure 1(a). The coarsening residual $\mathcal{G} \setminus \mathcal{G}'$ refers to the graph details contained within the supernodes that are excluded from the coarsened graph $\mathcal{G}'$. As such, the original graph $\mathcal{G}$ can be reconstructed using both $\mathcal{G}'$ and $\mathcal{G} \setminus \mathcal{G}'$.

2) Demystifying heterophily challenges through the lens of coarsening residual: Based on the identified coarsening residual, we provide an information-theoretic perspective on the heterophily challenges inherent in coarsening-based GNN training. We first introduce a proposition that formalizes the equivalence of cross-entropy loss and mutual information, enabling us to analyze how mutual information evolves during GNN training for deeper insights.

Proposition 1 ([21, Theorem 1])**.** *In training a neural network under ordinary regularity conditions (see [21]), the infimum of the expected cross-entropy loss with softmax output is equivalent to the mutual information between input and output variables up to constant* $\log c$ *under Assumption 1.*

The proposition reformulate the GNN training problem defined in Definition 1 as an optimization task based on mutual

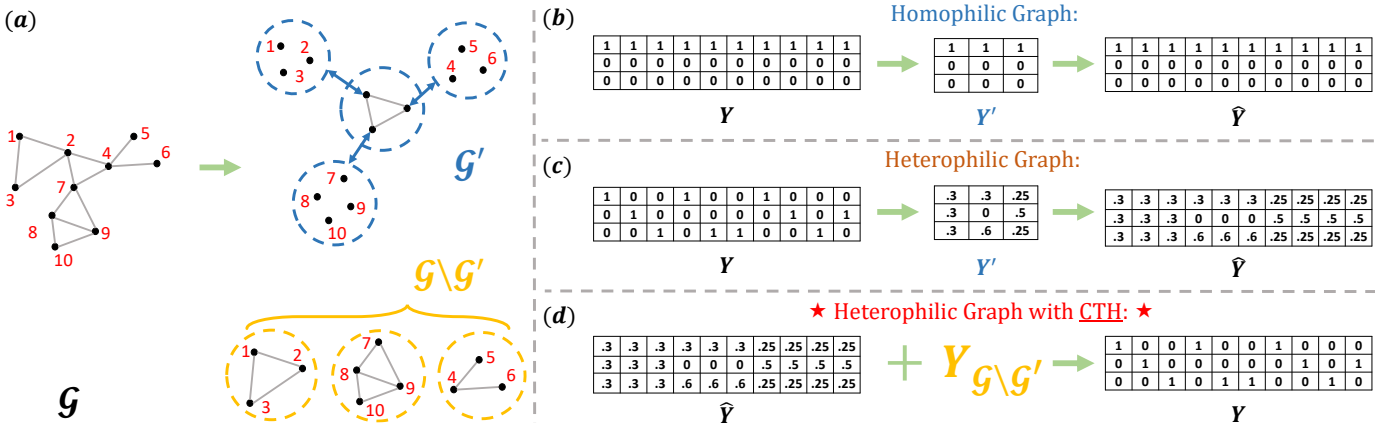

Fig. 1. Illustration of coarsening residual in graph coarsening and our CTH method. (a) shows the decomposition of the original graph $\mathcal{G}$ into its coarsened graph $\mathcal{G}'$ and the coarsening residual $\mathcal{G} \setminus \mathcal{G}'$. (b) and (c) contrast effects of the residual: in (b), homophilic graphs exhibit minimal residual, allowing the original label Y to be retrieved from the coarsened label Y', whereas in (c), heterophilic graphs lead to indispensable residual and thus impossible restoration of Y from Y'. (d) shows that CTH introduces a term $Y_{\mathcal{G}\setminus\mathcal{G}'}$ (c.f. Section IV-B) to compensate for the coarsening residual and recovers Y more accurately.

information. Specifically, the coarsening-based GNN training in Definition 1 seeks to maximize the mutual information $I(f(A, X; \Theta); Y)$ on the original graph by instead optimizing $I(f(A', X'; \Theta); Y')$ on the coarsened graph.

Thus, we respectively define Θ^* and Θ^{**} as the GNN parameters obtained by maximizing $I(f(A, X; \Theta); Y)$ and $I(f(A', X'; \Theta); Y')$, which can be formulated as:

$$\Theta^* \triangleq \arg\max_{\Theta} f(A, X; \Theta); Y) \ , \tag{1}$$

$$\Theta^{**} \triangleq \arg\max_{\Theta} f(A', X'; \Theta); Y') \ . \tag{2}$$

The effectiveness of coarsening-based GNN training is evaluated by examining the *discrepancy* between the GNN trained on the coarsened graph, $f(;\Theta^{**})$, and the GNN trained on the original full graph, $f(;\Theta^*)$. This discrepancy, analyzed through the lens of mutual information, can be represented by the positive term

$$\Delta \triangleq I(f(A, X; \Theta^*); Y) - I(f(A, X; \Theta^{**}); Y).$$

A larger Δ signifies greater performance degradation for the GNN trained on the coarsened graph than on the original one.

In light of this, we present a proposition to characterize Δ with $\mathcal{G} \setminus \mathcal{G}'$ (the proof is deferred to Section VIII in the complete manuscript [1]), stated as follows:

Proposition 2. *Let $g : \mathcal{G} \setminus \mathcal{G}' \mapsto \mathcal{Y}$ be a neural network specified in [21, Theorem 1] that maps $\mathcal{G} \setminus \mathcal{G}'$ to the node label space $\mathcal{Y}$ and maximizes the mutual information $I(g(\mathcal{G} \setminus \mathcal{G}'); Y | f(A, X; \Theta^{**}))$. Then, the following inequality holds:*

$$\Delta \geq I(g(\mathcal{G} \setminus \mathcal{G}'); Y | f(A, X; \Theta^{**})) \ . \tag{3}$$

Remark 1. This proposition highlights the pivotal role of coarsening residual $\mathcal{G} \setminus \mathcal{G}'$ on the effectiveness of coarsening-based GNN training, particularly in heterophilic graphs. Specifically, the lower bound $I(g(\mathcal{G} \setminus \mathcal{G}'); Y | f(A, X; \Theta^{**}))$ quantifies the amount of information regarding the label Y possessed in $\mathcal{G} \setminus \mathcal{G}'$, which reflects the information discarded during coarsening. This term holds limited significance for *homophilic graphs*: as shown in Figure 1(b), the original graph information Y can be fully recovered using only the coarsened graph information Y' and the partition P. On *heterophilic graphs*, however, as depicted in Figure 1(c), recovering Y from Y' is compromised, underscoring the critical role of the missing information in $\mathcal{G} \setminus \mathcal{G}'$. Hence, on heterophilic graphs, the discrepancy Δ grows significantly as the lower bound rises sharply, causing severe performance degradation in GNNs trained on coarsened graphs. This exposes a critical issue with graph heterophily in coarsening-based GNN training.

IV. CTH: Cracking Heterophily on Coarsening-based GNN Training

The preceding analysis of coarsening residual not only motivates fresh insights but also inspires a pathway for advancing coarsening-based GNN training tailored to heterophilic graphs. We anchor our solution in two pivotal concepts:

- A graph coarsening method that divides $\mathcal{G}$ into $\mathcal{G}'$ and $\mathcal{G} \setminus \mathcal{G}'$ while minimizing coarsening residual.
- A training paradigm that incorporates $\mathcal{G} \setminus \mathcal{G}'$ for further bolstering GNN training effectiveness.

These principles form the foundation of our CTH (**C**oarsening-based **T**raining on **H**eterophilic graphs). CTH operates in two sequential stages: ❶ *Similarity-guided Coarsening* and ❷ *Residual-reintegrated Training*. Each stage offers a systematic solution to one of the above principles. The following sections explore these stages in detail.

A. Stage ❶: similarity-guided coarsening

Graph coarsening has traditionally relied solely on the structural information of the original graph, i.e., the adjacency A, neglecting the influence of node feature X [9], [10], [12], [14], [22]. Recent innovations, such as ConvMatch [17], have demonstrated that integrating X into the coarsening process can significantly enhance GNN training efficacy by creating feature-informed coarsened graphs. Informed by these advances, our CTH proposes to build the coarsened graph

using an *augmented feature* that integrates information from a wider perspective. Specifically, we construct the augmented feature Ω by combining low-pass and high-pass components of X [23]–[25], along with a positional embedding $X_{\text{PE}} \in \mathbb{R}^{n\times d}$ pre-computed with Deepwalk algorithm [26], as shown below:

$$\Omega = [L^M X, (I - L^M)X, X_{\text{PE}}] \in \mathbb{R}^{n\times 3d} \ , \tag{4}$$

where the M is set to adjust the trade-off between low-pass component $L^M X$ and high-pass component $(I - L^M)X$.

With the augmented feature Ω, we then construct the coarsened graph through iteratively merging the supernode pairs with the highest similarity (we simply adopt the minimum *Euclidean distance* as the criteria) between the augmented features. We summarize the proposed procedure of the proposed graph coarsening in Algorithm 1.

Algorithm 1 Coarsened graph construction

Input: Original graph $\mathcal{G} = (A, X)$, augmented feature Ω, coarsening ratio r

Output: Coarsened graph $\mathcal{G}' = (A', X')$, partition $\mathcal{P}$

```
Initialize clusters P = {C_1, C_2, ..., C_n} with C_j = {v_j}
while |P|/n > r do
    (p*, q*) = argmin_{1≤p<q≤|P|} ‖ Avg(Ω_{C_p}) − Avg(Ω_{C_q}) ‖
    P ← P + (C_{p*} ∪ C_{q*}) − C_{p*} − C_{q*}
end while
A' = PAP^T, X' = C^{-1}PX, with (P, C) from P
return G' = (A', X'), P
```

By performing coarsening based on the augmented feature Ω, nodes clustered into the same supernode exhibit strong similarity in both graph structure and node features, achieving "local homophily" within supernodes. This property ensures that information within each supernode is retained with minimal residual, as it can be represented by the supernode. Moreover, the information of $\mathcal{G} \setminus \mathcal{G}'$ is effectively captured by Ω. Hence, the proposed Similarity-guided Coarsening satisfies the first principle of the solution.

B. Stage ❷: residual-reintegrated training

Proposition 2 shows that excluding the information from $\mathcal{G}\setminus\mathcal{G}'$ significantly undermines training efficacy on heterophilic graphs. This inspires a novel perspective that reintegrates the coarsening residual into GNN training. To achieve this, we propose an MLP-based post-compensation module to efficiently integrate the coarsening residual $\mathcal{G} \setminus \mathcal{G}'$ into the GNN output on the coarsened graph, $f(A', X'; \Theta)$. This enhancement produces an output $\tilde{Z}$, which serves as the input for loss computation during training. The procedure can be formally expressed as follows:

$$\tilde{Z} = P^T f(A', X'; \Theta) + (P^T f(A', X'; \Theta)) \odot \text{MLP}(\Omega) \ . \tag{5}$$

Here, $\text{MLP}(\cdot)$ serves as the implementation of the transformation $g(\cdot)$ described in Proposition 2, which is dataset-specific and would be jointly optimized with the GNN during training. Since MLPs operate on unified feature matrix instead of graph signals formulated as a tuple of graph structure and node feature, we propose using the augmented feature Ω as input. This choice ensures that the information from $\mathcal{G} \setminus \mathcal{G}'$ is preserved, bypassing the need to utilize its formulation shown in Definition 2. Furthermore, we enhance the model's learning capacity by applying a Hadamard product between $\text{MLP}(\Omega)$ and the GNN output on the coarsened graph, $P^T f(A', X'; \Theta)$. This aligns with prior studies demonstrating that the Hadamard product can boost capacities of neural networks over traditional operations [27]. Thus, the term $Y_{\mathcal{G}\setminus\mathcal{G}'} = (P^T f(A', X'; \Theta)) \odot \text{MLP}(\Omega)$ models the "residual" between the GNN's output on the full graph, $f(A, X; \Theta)$, and the output on the coarsened graph, compensating for the GNN's output, as depicted in Figure 1(d).

Note that the module is only applied during the training process. Specifically, in training, we minimize the loss between the adjusted output $\tilde{Z}$ and the original label Y, leveraging the post-compensation module to enhance GNN training effectiveness. In contrast, during testing, the trained GNN operates directly on the original graph, excluding the post-compensation module entirely.

V. Empirical Studies

This section validates the efficacy of our CTH compared to prior methods with standard node classification experiments.

A. Experimental settings

Datasets. We employ three extremely large heterophilic graphs: Gamers, Pokec, and Wiki, from [28], along with one homophilic graph, ogbn-arxiv, from [29]. The heterophilic graphs are split randomly into 50% training, 25% validation, and 25% testing, following the protocol in [28]. For ogbn-arxiv, we use the fixed split recommended in [29], with a ratio of 54% for training, 18% for validation, and 28% for testing.

Baselines. Three state-of-the-art coarsening-based GNN training methods, SCAL [15], VNG [16], and ConvMatch [17], are used as baselines for comparison with CTH. As a reference, we also include a baseline trained on the original graph, referred to as the "Full graph" baseline. All experiments are conducted using the PyG library [30]. Besides, both VNG and ConvMatch are reimplemented; VNG lacks public code and ConvMatch is only available in a different code library.

Hyper-parameter setting of CTH. We set the exponent M in Eq. (4) as 5 for all experiments. The post-compensation module is configured as MLP with 3 layers and 256 hidden dimension, where learning-rate of 0.01, weight-decay of 0.0005, and dropout rate of 0.5 are adopted during training.

Backbone GNNs. We employ three widely used GNN models—GCN [31], APPNP [32], and GPRGNN [33]—as the backbone models. The hyper-parameters for each backbone GNN are fixed based on grid search results from full graph training specific to each dataset, and will be applied across all scalable training approaches for fairness.

Training details. We train the models for a maximum of 500 epochs, with early stopping applied if validation accuracy

TABLE I
NODE CLASSIFICATION RESULTS ON LARGE GRAPHS: MEAN ACCURACY (%) ± STANDARD DEVIATION. "OOM" DENOTES "OUT-OF-MEMORY".

Backbone	Method	Gamers		Pokec		Wiki		ogbn-arxiv (non-heterophilic)	
		1%	10%	1%	10%	1%	10%	1%	10%
GCN	Full graph	$63.55_{\pm 0.2}$		$73.29_{\pm 0.3}$		OOM		$71.82_{\pm 0.3}$	
	SCAL	$30.81_{\pm 0.2}$	$35.36_{\pm 0.2}$	$39.81_{\pm 0.5}$	$44.63_{\pm 0.3}$	$25.66_{\pm 0.3}$	$32.42_{\pm 0.3}$	$53.55_{\pm 0.2}$	$66.38_{\pm 0.2}$
	VNG	$40.51_{\pm 0.3}$	$43.28_{\pm 0.2}$	$44.66_{\pm 0.3}$	$51.91_{\pm 0.2}$	$36.36_{\pm 0.3}$	$41.45_{\pm 0.3}$	$62.72_{\pm 0.2}$	$65.91_{\pm 0.2}$
	ConvMatch	$42.33_{\pm 0.3}$	$47.18_{\pm 0.4}$	$41.59_{\pm 0.3}$	$54.27_{\pm 0.3}$	$38.52_{\pm 0.3}$	$40.18_{\pm 0.3}$	$\mathbf{63.63_{\pm 0.2}}$	$\mathbf{66.69_{\pm 0.3}}$
	CTH (**ours**)	$\mathbf{48.91_{\pm 0.4}}$	$\mathbf{53.61_{\pm 0.4}}$	$\mathbf{55.81_{\pm 0.5}}$	$\mathbf{63.15_{\pm 0.6}}$	$\mathbf{41.79_{\pm 0.6}}$	$\mathbf{47.53_{\pm 0.5}}$	$62.59_{\pm 0.3}$	$66.61_{\pm 0.3}$
APPNP	Full graph	$61.83_{\pm 0.2}$		$62.16_{\pm 0.2}$		$51.55_{\pm 0.2}$		$71.59_{\pm 0.1}$	
	SCAL	$30.42_{\pm 0.2}$	$36.18_{\pm 0.3}$	$35.06_{\pm 0.2}$	$42.18_{\pm 0.3}$	$24.71_{\pm 0.2}$	$33.86_{\pm 0.2}$	$54.38_{\pm 0.1}$	$65.11_{\pm 0.2}$
	VNG	$39.87_{\pm 0.3}$	$42.44_{\pm 0.3}$	$43.13_{\pm 0.3}$	$51.36_{\pm 0.3}$	$33.88_{\pm 0.3}$	$40.79_{\pm 0.3}$	$61.66_{\pm 0.3}$	$64.62_{\pm 0.2}$
	ConvMatch	$41.50_{\pm 0.2}$	$45.43_{\pm 0.3}$	$43.36_{\pm 0.4}$	$52.72_{\pm 0.3}$	$36.42_{\pm 0.3}$	$39.62_{\pm 0.4}$	$62.75_{\pm 0.2}$	$\mathbf{65.51_{\pm 0.2}}$
	CTH (**ours**)	$\mathbf{47.47_{\pm 0.5}}$	$\mathbf{52.77_{\pm 0.5}}$	$\mathbf{53.29_{\pm 0.5}}$	$\mathbf{60.64_{\pm 0.5}}$	$\mathbf{42.60_{\pm 0.5}}$	$\mathbf{45.92_{\pm 0.5}}$	$\mathbf{62.81_{\pm 0.3}}$	$65.28_{\pm 0.3}$
GPRGNN	Full graph	$62.59_{\pm 0.3}$		$80.74_{\pm 0.2}$		$58.73_{\pm 0.3}$		$71.88_{\pm 0.2}$	
	SCAL	$33.74_{\pm 0.3}$	$43.47_{\pm 0.3}$	$38.22_{\pm 0.3}$	$63.35_{\pm 0.3}$	$28.54_{\pm 0.2}$	$40.19_{\pm 0.3}$	$53.69_{\pm 0.2}$	$63.34_{\pm 0.3}$
	VNG	$41.30_{\pm 0.2}$	$49.15_{\pm 0.3}$	$55.82_{\pm 0.3}$	$68.73_{\pm 0.3}$	$38.32_{\pm 0.3}$	$44.57_{\pm 0.3}$	$62.81_{\pm 0.3}$	$65.88_{\pm 0.3}$
	ConvMatch	$41.88_{\pm 0.3}$	$48.79_{\pm 0.3}$	$57.60_{\pm 0.3}$	$66.53_{\pm 0.2}$	$40.16_{\pm 0.3}$	$47.38_{\pm 0.3}$	$\mathbf{64.15_{\pm 0.3}}$	$\mathbf{66.89_{\pm 0.3}}$
	CTH (**ours**)	$\mathbf{46.11_{\pm 0.4}}$	$\mathbf{56.38_{\pm 0.5}}$	$\mathbf{64.67_{\pm 0.4}}$	$\mathbf{71.39_{\pm 0.5}}$	$\mathbf{49.90_{\pm 0.4}}$	$\mathbf{52.56_{\pm 0.4}}$	$63.37_{\pm 0.4}$	$66.27_{\pm 0.4}$

TABLE II
ABLATION STUDY ON POST-COMPENSATION MODULE (PCM). ACC_RED INDICATES THE ACCURACY REDUCTION WHEN PCM IS EXCLUDED.

Dataset	w/ PCM	w/o PCM	#ACC_RED
Gamers	$\mathbf{48.91_{\pm 0.4}}$	$44.34_{\pm 0.3}$	**4.57**
Pokec	$\mathbf{55.81_{\pm 0.5}}$	$47.69_{\pm 0.4}$	**8.12**
Wiki	$\mathbf{41.79_{\pm 0.6}}$	$37.38_{\pm 0.4}$	**4.41**
Arxiv	$\mathbf{62.59_{\pm 0.3}}$	$62.12_{\pm 0.3}$	**0.47**

TABLE III
RUNTIME RESULTS WITH GCN BACKBONE AND 1% COARSENING RATIO.

Dataset	SCAL	VNG	ConvMatch	CTH (**ours**)
Gamers	24.4min	35.7min	38.3min	25.6min
Pokec	6.3h	9.7h	10.6h	6.9h
Wiki	15.8h	26.4h	25.9h	17.0h
Arxiv	13.9min	17.8min	19.5min	14.2min

does not improve over 50 epochs. The Adam optimizer [34] is used for optimization. Experiments are conducted with coarsening ratios 1% and 10% for each dataset, and the results for each method are averaged from 5 random initializations.

B. Results and analysis

Improvements on training efficacy. As shown in Table I, GNNs trained with our CTH method consistently outperform those trained with other approaches across various coarsening ratios. The improvements are especially pronounced on heterophilic graphs and comparable on the homophilic ogbn-arxiv. These results highlight the significant advantages of CTH in enhancing the efficacy of GNN training over the state-of-the-art coarsening-based approaches.

Efficacy of post-compensation module (PCM). In Table II, we present ablation studies on the post-compensation module (PCM), implemented with a GCN backbone and a 1% coarsening ratio. The results reveal that omitting the post-compensation module from CTH (labeled "w/o PCM") leads to a noticeable reduction in performance, with the performance drop being significantly larger on heterophilic graphs compared to homophilic ones. These findings emphasize the effectiveness of the post-compensation module in mitigating the coarsening residual caused by the coarsening process. They also corroborate our theoretical analysis in Section III-B, which suggests that the impact of coarsening residual is more severe in heterophilic graphs.

Comparable time complexity. Table III presents the total runtime, which includes graph coarsening time, for each method with a GCN backbone and a 1% coarsening ratio. The results indicate that CTH achieves the second-lowest runtime overall, notably outperforming VNG and ConvMatch, while maintaining a runtime that is only marginally higher than SCAL. These outcomes emphasize the superiority of our CTH, which delivers both reduced time complexity and improved GNN training performance.

VI. CONCLUSIONS

In this paper, we delve into coarsening-based GNN training on heterophilic graphs, framed through the lens of coarsening residual. We first establish that the coarsening residual on heterophilic graphs leads to significant degradation of coarsening-based GNN training, pointing to the critical challenge of graph heterophily. To compensate for the information loss within the residual, we introduce CTH, an advanced coarsening-based GNN training framework. CTH starts by creating a coarsened graph through clustering nodes with high similarity in the augmented feature. It then trains GNNs on the coarsened graph, integrating a post-compensation module to boost training efficacy. Extensive experiments reveal that CTH significantly outperforms state-of-the-art methods, especially on heterophilic graphs of our interest, positioning it as an advanced solution and key advancement in coarsening-based scalable GNN training.

REFERENCES

[1] "Repo for complete manuscript and model implementation," https://anonymous.4open.science/r/Cracking-Heterophily-FD97/.

[2] Y. Zhou, H. Zheng, X. Huang, S. Hao, D. Li, and J. Zhao, "Graph neural networks: Taxonomy, advances, and trends," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 1, jan 2022. [Online]. Available: https://doi.org/10.1145/3495161

[3] Z. Shan, X. Yi, H. Yu, C.-S. Liao, and S. Jin, "Learning to code on graphs for topological interference management," in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 2386–2391.

[4] J. Clausius, M. Geiselhart, D. Tandler, and S. t. Brink, "Graph neural network-based joint equalization and decoding," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 1203–1208.

[5] A. Gong, S. Cammerer, and J. M. Renes, "Graph neural networks for enhanced decoding of quantum ldpc codes," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 2700–2705.

[6] Z. Shan, X. Yi, L. Liang, C.-S. Liao, and S. Jin, "Grlinq: A distributed link scheduling mechanism with graph reinforcement learning," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 2436–2441.

[7] A. Magner, M. Baranwal, and A. O. Hero, "Fundamental limits of deep graph convolutional networks for graph classification," *IEEE Transactions on Information Theory*, vol. 68, no. 5, pp. 3218–3233, 2022.

[8] J. Chen, Y. Saad, and Z. Zhang, "Graph coarsening: from scientific computing to machine learning," *SeMA Journal*, vol. 79, no. 1, pp. 187–223, 2022.

[9] A. Loukas and P. Vandergheynst, "Spectrally approximating large graphs with smaller graphs," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 3237–3246. [Online]. Available: https://proceedings.mlr.press/v80/loukas18a.html

[10] Y. Jin, A. Loukas, and J. JaJa, "Graph coarsening with preserved spectral properties," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 4452–4462. [Online]. Available: https://proceedings.mlr.press/v108/jin20a.html

[11] M. Kumar, A. Sharma, and S. Kumar, "A unified framework for optimization-based graph coarsening," *Journal of Machine Learning Research*, vol. 24, no. 118, pp. 1–50, 2023. [Online]. Available: http://jmlr.org/papers/v24/22-1085.html

[12] Y. Chen, R. Yao, Y. Yang, and J. Chen, "A gromov-Wasserstein geometric view of spectrum-preserving graph coarsening," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 5257–5281. [Online]. Available: https://proceedings.mlr.press/v202/chen23ak.html

[13] M. Kumar, A. Sharma, S. Saxena, and S. Kumar, "Featured graph coarsening with similarity guarantees," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 17 953–17 975. [Online]. Available: https://proceedings.mlr.press/v202/kumar23a.html

[14] A. Joly and N. Keriven, "Graph coarsening with message-passing guarantees," 2024. [Online]. Available: https://arxiv.org/abs/2405.18127

[15] Z. Huang, S. Zhang, C. Xi, T. Liu, and M. Zhou, "Scaling up graph neural networks via graph coarsening," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 675–684. [Online]. Available: https://doi.org/10.1145/3447548.3467256

[16] S. Si, F. Yu, A. S. Rawat, C.-J. Hsieh, and S. Kumar, "Serving graph compression for graph neural networks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=T-qVtA3pAxG

[17] C. Dickens, E. Huang, A. Reganti, J. Zhu, K. Subbian, and D. Koutra, "Graph coarsening via convolution matching for scalable graph neural network training," in *Companion Proceedings of the ACM on Web Conference 2024*, ser. WWW '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1502–1510. [Online]. Available: https://doi.org/10.1145/3589335.3651920

[18] X. Zheng, Y. Wang, Y. Liu, M. Li, M. Zhang, D. Jin, P. S. Yu, and S. Pan, "Graph neural networks for graphs with heterophily: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2202.07082

[19] C. Gong, Y. Cheng, X. Li, C. Shan, and S. Luo, "Learning from graphs with heterophily: Progress and future," 2024. [Online]. Available: https://arxiv.org/abs/2401.09769

[20] F. Chung, *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, 1997, vol. 92.

[21] Z. Qin, D. Kim, and T. Gedeon, "Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator," 2020. [Online]. Available: https://arxiv.org/abs/1911.10688

[22] M. Purohit, B. A. Prakash, C. Kang, Y. Zhang, and V. Subrahmanian, "Fast influence-based coarsening for large networks," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1296–1305. [Online]. Available: https://doi.org/10.1145/2623330.2623701

[23] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[24] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, 2020.

[25] M. Balcilar, G. Renton, P. Héroux, B. Gaüzère, S. Adam, and P. Honeine, "Analyzing the expressive power of graph neural networks in a spectral perspective," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=-qh0M9XWxnv

[26] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 701–710. [Online]. Available: https://doi.org/10.1145/2623330.2623732

[27] Y. Wu, Z. Zhu, F. Liu, G. Chrysos, and V. Cevher, "Extrapolation and spectral bias of neural nets with hadamard product: a polynomial net study," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 26 980–26 993. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/acb3565a58dea4c39c84af35d4225d97-Paper-Conference.pdf

[28] D. Lim, F. M. Hohne, X. Li, S. L. Huang, V. Gupta, O. P. Bhalerao, and S.-N. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=DfGu8WwT0d

[29] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 22 118–22 133.

[30] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," 2019. [Online]. Available: https://arxiv.org/abs/1903.02428

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[32] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Combining neural networks with personalized pagerank for classification on graphs," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=H1gL-2A9Ym

[33] E. Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=n6jl7fLxrP

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[35] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

Appendix

VII. Experimental Details

A. Dataset statistics

We present the Statistics for the large-scale graph datasets utilized in the experiments. Following the definition in [18], [19], "# Homo" refers to the homophily ratio of graphs, with lower values signifying greater heterophily.

TABLE IV
STATISTICS FOR THE GRAPH DATASETS.

	Gamers	Pokec	Wiki	Ogbn-arxiv
# Nodes	168,114	1,632,803	1,925,342	169,343
# Edges	6,797,557	30,622,564	303,434,860	1,157,799
# Features	7	65	600	128
# Classes	2	2	5	40
# Homo	0.55	0.45	0.39	0.65

B. Experimental environments

Experiments were carried out on a dual NVIDIA A100 GPU setup, each equipped with 80GB of memory. The testing environment runs on a Linux OS (Ubuntu 22.04), with CUDA 11.8 for parallel computation support. Python is used as the programming language, and all datasets and models are implemented using PyTorch Geometric (PyG). A list of the involved packages, along with their respective version numbers, is provided below:

TABLE V
DETAILS OF INVOLVED PACKAGES.

Packages	Versions
python	3.10
torch	2.4.0
torchvision	0.19.1
torchaudio	2.4.1
torch-geometric	2.6.0
pyg_lib	0.4.0
torch_cluster	1.6.3
torch_scatter	2.1.2
torch_sparse	0.6.18
torch_spline_conv	1.2.2

VIII. Proof of Proposition 2

Proof. Recall the chain-rule of mutual information [35]:

$$I(U,V;W) - I(U;W) = I(V;W|U) \tag{6}$$

Let $U = f(A, X; \Theta^{**})$, $V = g(\mathcal{G} \setminus \mathcal{G}')$, $W = Y$, we obtain:

$$\begin{aligned} &I\Big(f(A,X;\Theta^{**}), g(\mathcal{G}\setminus\mathcal{G}');Y\Big) - I\Big(f(A,X;\Theta^{**});Y\Big) \\ &= I\Big(g(\mathcal{G}\setminus\mathcal{G}');Y\Big|f(A,X;\Theta^{**})\Big)\ . \end{aligned} \tag{7}$$

We focus on the term $I\Big(f(A, X; \Theta^{**}), g(\mathcal{G} \setminus \mathcal{G}'); Y\Big)$, which quantifies the relationship between two models trained on different partitions of the graph $\mathcal{G}$—namely, $\mathcal{G}'$ and $\mathcal{G}\setminus\mathcal{G}'$—and the labels of the entire graph, Y. Specifically, we leverage the data processing inequality [35]:

$$I(U;V) \geq I(U;W), \text{ with Markov chain: } U \to V \to W\ . \tag{8}$$

Notice that $f(;\Theta^*)$, $f(;\Theta^{**})$, and g are all deterministic functions. Hence, by letting $U = Y$, $V = f(A, X; \Theta^*)$, and $W = \Big(f(A, X; \Theta^{**}), g(\mathcal{G} \setminus \mathcal{G}')\Big)$, we can derive the following inequality:

$$I\Big(f(A,X;\Theta^*);Y\Big) \geq I\Big(f(A,X;\Theta^{**}), g(\mathcal{G}\setminus\mathcal{G}');Y\Big)\ . \tag{9}$$

The inequality encapsulates the intuitive principle that the aggregate contribution from the model trained on the coarsened graph, $f(A, X; \Theta^{**})$, and the additional term representing the information excluded in the coarsened graph, $g(\mathcal{G} \setminus \mathcal{G}')$, is bounded above by the output of the model trained directly on the full graph, $f(A, X; \Theta^*)$.

Finally, by combining Equation 7 and Equation 9, and observing that $I\Big(f(A, X; \Theta^*); Y\Big) \geq I\Big(f(A, X; \Theta^{**}); Y\Big)$, the following inequality is derived:

$$\begin{aligned} \Delta &= \Big|I\Big(f(A,X;\Theta^*);Y\Big) - I\Big(f(A,X;\Theta^{**});Y\Big)\Big| \\ &\geq I\Big(g(\mathcal{G}\setminus\mathcal{G}');Y|f(A,X;\Theta^{**})\Big)\ . \end{aligned} \tag{10}$$

This completes the proof. □