applied sciences

MDPI

Article

GCARe: Mitigating Subgroup Unfairness in Graph Condensation through Adversarial Regularization

Runze Mao [1], Wenqi Fan [2] and Qing Li [2,*]

[1] Department of Computer Science, City University of Hong Kong, Hong Kong, China; runzemao2-c@my.cityu.edu.hk
[2] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China; wenqifan03@gmail.com
* Correspondence: csqli@comp.polyu.edu.hk

Abstract: Training Graph Neural Networks (GNNs) on large-scale graphs in the deep learning era can be expensive. While graph condensation has recently emerged as a promising approach through which to reduce training cost by compressing large graphs into smaller ones and for preserving most knowledge, its capability in treating different node subgroups fairly during compression remains unexplored. In this paper, we investigate current graph condensation techniques from a perspective of fairness, and show that they bear severe disparate impact toward node subgroups. Specifically, GNNs trained on condensed graphs become more biased than those trained on original graphs. Since the condensed graphs comprise synthetic nodes, which are absent of explicit group IDs, the current algorithms used to train fair GNNs fail in this case. To address this issue, we propose **G**raph **C**ondensation with **A**dversarial **Re**gularization (**GCARe**) , which is a method that directly regularizes the condensation process to distill the knowledge of different subgroups fairly into resulting graphs. A comprehensive series of experiments substantiated that our method enhances the fairness in condensed graphs without compromising accuracy, thus yielding more equitable GNN models. Additionally, our discoveries underscore the significance of incorporating fairness considerations in data condensation, and offer invaluable guidance for future inquiries in this domain.

Keywords: data distillation; graph neural networks; fairness; adversarial learning

Citation: Mao, R.; Fan, W.; Li, Q. GCARe: Mitigating Subgroup Unfairness in Graph Condensation through Adversarial Regularization. *Appl. Sci.* **2023**, *13*, 9166. https://doi.org/10.3390/app13169166

Academic Editor: Mohamed Benbouzid

Received: 30 June 2023
Revised: 2 August 2023
Accepted: 3 August 2023
Published: 11 August 2023

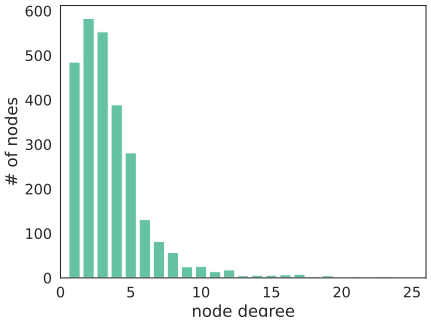
1. Introduction

Graph neural networks (GNNs) have emerged as popular models for handling graph data in various fields, such as social networks, recommender systems, molecular topology, and chemistry, owing to their capacity to leverage graph structures and aggregate neighborhood information [1–5]. However, modern deep GNNs are data-hungry and demand extensive training data to realize their full potential. Training GNNs on large graphs introduces challenges, leading to increased computational complexity, memory requirements, training time, and carbon emissions [6]. Furthermore, hyperparameter tuning and architecture searches [7] on large graphs exponentially exacerbate these complexities. As a result, scalable GNN training has become increasingly important [8,9].

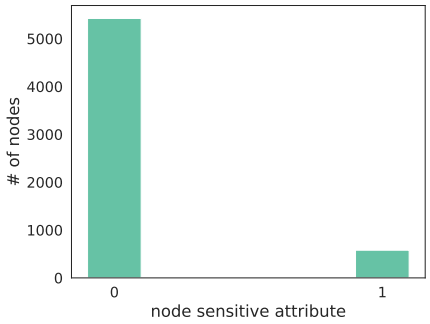
Graph condensation [10,11] is newly developed as a promising approach for compressing large graphs into smaller ones, significantly reducing both training time and memory consumption without sacrificing the performance of the GNNs trained on condensed graphs. It achieves this by feeding both the original and condensed graphs into the GNN, and by aligning the model gradients that correspond to each input at every iteration. The condensed graph is treated as if it were learnable parameters upon which convergence would induce GNNs, and this is such as if they were trained on the original graph. Once the condensation is complete, training GNNs on the condensed graph can be efficient. For instance, if the Ogbn-arxiv [12] dataset is condensed to a size of 0.05% for its nodes, training on a condensed graph leads to a four-time greater acceleration compared to training on

the original graph (measured on one NVIDIA RTX 2080ti GPU). Thus, the condensed graph can serve as a substitute for the original graph and can be utilized repeatedly. This is particularly helpful in applications like continual learning [13] and hyperparameter search, where one needs to train GNNs multiple times.

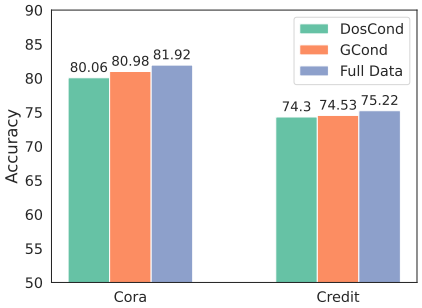
Although current graph condensation methods are capable of distilling most of the knowledge from the original graphs, we find that this capability varies with respect to different node subgroups, leading to severe fairness issues. Specifically, the GNNs trained on condensed graphs exhibit much larger performance gaps between advantaged and disadvantaged node subgroups when compared to those trained on original graphs. Figure 1 demonstrates such phenomenon on two real-world datasets, namely Cora and Credit-defaulter. In the Cora dataset, the node degrees follow a long-tail distribution; in it, we split the nodes into 3 subgroups according to their degrees. In the Credit-defaulter dataset, nodes are annotated with the Age attribute; in it, we used this attribute to split all of the nodes into two subgroups. We evaluated two existing graph condensation methods, GCond [11] and DosCond [10], and compared them against their use with the original full graph. We report both the prediction accuracy and fairness metrics. For Cora, the subgroup fairness was measured by Δ_{acc}, which is the difference between the highest and lowest subgroup accuracy; for Credit, the fairness metric was Δ_{EO}, which is the difference in true positive rates between two subgroups. As shown in the figure, while GNNs trained on condensed graphs by GCond or DosCond can achieve similar overall accuracies to training on full data, they are more biased toward certain advantaged groups, and the performance on disadvantaged groups is sacrificed.



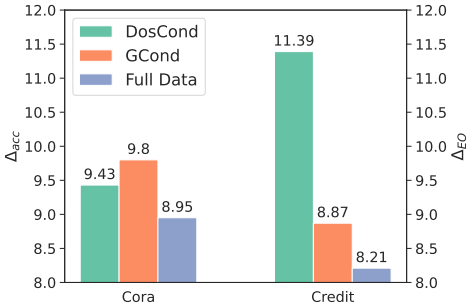
(**a**) Cora node degree distribution.



(**b**) Credit-defaulter imbalanced sensitive attribute distribution.



(**c**) Overall accuracy.



(**d**) Fairness evaluation.

Figure 1. Empirical evaluation on the Cora and Credit-defaulter datasets. For Cora, we used node degree to split the subgroups. For Credit, we used the node attribute Age. (**a**): The node degrees of Cora follow a long-tail distribution. (**b**): The Credit dataset was highly imbalanced regarding a specific attribute. (**c**): GCond and DosCond yielded comparable overall accuracy on both datasets. (**d**): GCond and DosCond resulted in a larger disparate impact for GNNs when compared to training on the original full data.

In this paper, we investigate the issue of the exacerbated subgroup unfairness that is present in current graph condensation techniques. Although a substantial body of literature exists on the topic of GNN fairness, existing algorithms of fair GNN training cannot be trivially applied in this case because they require that nodes in the training data are explicitly annotated with group IDs. However, current graph condensation methods do not preserve node attributes or group IDs during the compression process. Furthermore, current graph condensation methods are embedded within a complex bi-level optimization framework [10,11,14], making it even harder to integrate with modules like data augmentation [15], graph contrastive learning [16], or pseudo label prediction [17]. To address this problem, we propose **G**raph **C**ondensation with **A**dversarial **Re**gularization (**GCARe**), which is a light-weight, efficient, and effective regularization term. GCARe renders condensed graphs that preserve the knowledge of different subgroups fairly to alleviate the unfairness of GNNs that are trained afterward. It achieves this by directly regularizing the condensation process with adversarial training. Specifically, we treat the condensation model as a generator, and append a discriminator to the output layer of the condensation model. During condensation, both the node features and group IDs are fed to the generator in order to produce hidden embeddings, from which the discriminator tries to predict the group IDs of the input nodes. Meanwhile, the generator is trained to fool the discriminator and to prevent it from making correct predictions. In this way, the group ID information is eliminated from the node representations, thus making the condensation model unbiased. Intuitively, GNNs trained on condensed graphs would inherit the unfairness of the GNNs that are utilized during the condensation procedure (i.e., the condensation model); this is because the latter one acts as the information bottleneck between the original graph and the former one. Therefore, it is necessary to impose a fair GNN to condense the graphs appropriately.

Our contributions can be summarized as follows:

1. We provide broad empirical evaluations, which reveal that performance disparities between advantaged and disadvantaged subgroups become more pronounced for GNNs that are trained on condensed graphs.
2. We introduce GCARe, an innovative paradigm that leverages adversarial learning to debias the condensation model, thereby achieving fair graph condensation. Notably, our proposed method maintains the same level of complexity as existing condensation techniques, and it can be implemented as a regularization term that is easy to plug and play.
3. Our comprehensive experiments demonstrate that GCARe not only mitigates fairness issues, but also enhances the overall performance of condensed graphs. For example, when applied to the Recidivism dataset, GCARe reduces the Δ_{SP} value from 4.47% to 3.98%, as well as the Δ_{EO} value from 3.40% to 2.52%, while simultaneously increasing the accuracy from 80.04% to 81.97%.

2. Related Work

2.1. Dataset Condensation

Dataset distillation and condensation are techniques aimed at reducing the size of large datasets while maintaining their core information and utility for training machine learning models. Wang et al. [18] distilled knowledge from massive image datasets into a few synthetic images, while still keeping the comparable performances of the models trained on them. Nonetheless, this method involves a nested loop optimization and is computationally expensive. Zhao et al. [14] proposed to align the gradients of the model parameters to correspond with real and synthetic training data. GCond [11] extends this idea to graph data, whereby learning synthetic graphs with only hundreds of nodes have training GNNs applied to them. DosCond [10] further accelerates the condensation process by performing one-step updates for nested optimizations, i.e., it only updates the synthetic graph without training the upstream GNNs that are used for condensation. Although these methods can preserve the core utilities of the original graph, we observed degradation in

the subgroup fairness for GNNs that were trained on synthetic graphs. In this work, our primary focus is on how to effectively apply condensed graph representations that treat node subgroups fairly in order to prevent possible harmful applications.

2.2. GNN Fairness

Fairness has always been an important topic in machine learning [19–22]. The bias issue in GNNs arises due to several factors: (i) imbalanced node degree distribution [16,23]; (ii) spurious correlation between labels and specific node attributes, e.g., gender and age [15,24]; and (iii) the homophily nature of real-world graph data where nodes with similar characteristics tend to group together [25]. Training GNNs that are unbiased and fair with respect to different subgroups of nodes is significant in the context of trustworthy AI applications [26,27]. Graph contrastive learning [16,28], pseudo labels [17,23], and sampling-based methods [29] have been proposed to tackle the structural bias of GNNs, i.e., fairness with respect to nodes with different degrees. On the other hand, when nodes are annotated with certain sensitive attributes, e.g., age, gender, and skin color, it has been observed that GNNs discriminate against nodes with certain attribute values, as well as make inferior predictions on these subgroups (which are called attribute bias). Adversarial learning [24], data augmentation [15], and causal inference [30] have been utilized to address these problems. In this paper, we simultaneously consider both structural and attribute bias, and apply our findings to learning condensed graphs that can fairly treat subgroups that have been divided according to node degrees and node attributes.

3. Background

In this section, we first introduce the notations to be used throughout this paper. Then, we brief the readers on the idea and algorithm of graph condensation.

In this paper, we consider the task of transductive node classification. A graph consisting of N nodes can be denoted as a triple (A, X, Y), where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, $X \in \mathbb{R}^{N\times D}$ is the feature matrix of the nodes, and $Y \in \{0, \ldots, C-1\}^N$ is the node labels over the C classes. Given a GNN model, we use θ to denote its model parameters, $\mathrm{GNN}_\theta(A, X)$ denotes the message-passing procedure, and $L(\mathrm{GNN}_\theta(A, X), Y)$ represents the cross-entropy classification loss.

Graph Condensation via Gradient Matching

Given a target graph dataset $\mathcal{T} = (A_\mathcal{T}, X_\mathcal{T}, Y_\mathcal{T})$ with $N_\mathcal{T}$ nodes, graph condensation aims at learning on such a synthetic graph dataset $\mathcal{S} = (A_\mathcal{S}, X_\mathcal{S}, Y_\mathcal{S})$ with $N_\mathcal{S}$ nodes ($N_\mathcal{S} \ll N_\mathcal{T}$), whereby a GNN trained on $\mathcal{S}$ performs comparably to one trained on $\mathcal{T}$. To achieve this goal, Zhao et al. [14] proposed to feed a model with batches from both the original and condensed datasets, and match the model gradients at each iteration. The intuition behind this is that, by forcing the gradients to be similar at any step, the model is expected to converge to the same point when trained on either dataset. The gradient matching loss is used to update the condensed data.

GCond [11] extends this idea to the graph field. Specifically, the node features of the synthetic graph, i.e., $X_\mathcal{S}$, are treated as learnable parameters, and are initialized first. The adjacency matrix is modeled as a function of the node features via the parameters Φ: $A_\mathcal{S} = \Phi(X_\mathcal{S})$. A GNN model is then employed as the agent for compressing the graph, which we refer to as the upstream GNN. Then, at the t-th iteration, both the original graph and condensed graph are fed to the GNN, and the corresponding gradients are computed through backpropagation:

$$g_t^\mathcal{S} = \nabla_\theta L\Big(\mathrm{GNN}_{\theta_t}(\Phi(X_\mathcal{S}), X_\mathcal{S}), Y_\mathcal{S}\Big) \tag{1}$$

$$g_t^\mathcal{T} = \nabla_\theta L\Big(\mathrm{GNN}_{\theta_t}(A_\mathcal{T}, X_\mathcal{T}), Y_\mathcal{T}\Big) \tag{2}$$

The distance between the two gradients are computed layer by layer via cosine similarity to obtain the gradient matching loss $D(g_t^{\mathcal{S}}, g_t^{\mathcal{T}})$, which is then backwarded to update both Φ and X_S. Note that this involves computing the second-order derivatives $\frac{\partial^2 L}{\partial \theta \partial X_S}$, which is time-consuming. Then, the upstream GNN is trained for several steps on the updated synthetic graph. This is achieved by aligning the gradients at each iteration, whereby the synthetic graph imitates the whole trajectory of the model updates on the original graph. GCond gained remarkable success on graph condensation. For instance, it was able to approximate the accuracy by up to 95.3% on reddit, and 99.0% on Citeseer while reducing the graph sizes by 99.9%.

While GCond suffers from the time complexity that occurs when interchangeably updating the synthetic graph and the upstream GNN, DosCond [10], on the other hand, skips the latter and only updates the synthetic graph. This significantly accelerates the condensation process without sacrificing the performance.

Our method is highly flexible and can be easily integrated with either condensation method. In this paper, we utilize both methods as the backbone.

4. Graph Condensation with Adversarial Regularization

In this section, we introduce our proposed method, namely GCARe, which is used to regularize the graph condensation process for a more fair distillation of all node subgroups.

A rich body of literature focuses on GNN fairness from different perspectives, including data augmentation [15], graph contrastive learning [16], pseudo label prediction [17], adversarial training [24], and fairness constraints [26]. While one cannot directly apply these methods to GNN training on condensed graphs owing to the absence of explicit group IDs after condensation, we chose to regularize the condensation GNNs during condensation. The intuition behind this choice is that, the condensation GNN plays the role of information bottleneck between the original graph and the GNNs trained on condensed graphs, and the latter will inherit the bias and unfairness from the former.

The framework of our proposed method GCARe is depicted in Figure 2. At its core, GCARe adopts a Generative Adversarial Network (GAN) [24,31,32] to debias condensation GNNs. We treat condensation GNNs as the generator, and it generates hidden embeddings for the node v from the original graph:

$$h_v = \text{GNN}_\theta(\mathcal{G}_v, x_v)$$

where $\mathcal{G}_v$ is the adjacency associated with v, and x_v is the input node feature. On the other hand, a linear classification layer f was adopted as the discriminator to predict which subgroup v comes from. The generator plays a minimax game against the discriminator to prevent correct group prediction. The game can be formulated as follows:

$$\max_\theta \min_f L_{adv} := -\sum_{v \in X} \sum_{k=1}^{K} \mathbb{1}_{\{s_v = k\}} \log f(\text{GNN}_\theta(\mathcal{G}_v, x_v))_k$$

where $\mathbb{1}$ is the indicator function, $f(\cdot)$ is a vector containing the predictive probabilities for each subgroup, $f(\cdot)_k$ is the probability specifically for the k-th subgroup, and K is the number of all of the subgroups. We add this loss term to the original condensation loss, weighted by a hyperparameter λ, as a regularization of the condensation GNN. Thus, the total loss at the t-th iteration is as follows:

$$L_t := D(g_t^{\mathcal{S}}, g_t^{\mathcal{T}}) + \lambda L_{adv}$$

In practice, we adopt an alternative optimization scheme. Specifically, in each iteration, we first freeze the condensation GNN, and minimize $D(g_t^{\mathcal{S}}, g_t^{\mathcal{T}})$ with respect to the condensed graph. Then, we minimize the adversarial loss with respect to f. Lastly, we unfreeze the GNN and maximize the adversarial loss with respect to it.

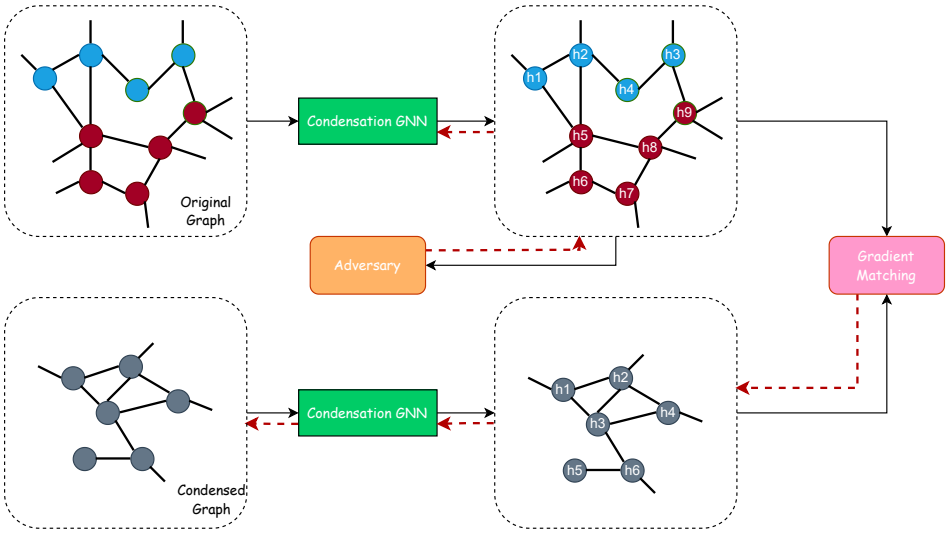


Figure 2. Framework of our method. At each iteration, both the original graph and condensed graph were fed to the condensation GNN, which is regularized by the proposed adversary term. Red dotted lines indicate a backward gradient.

5. Experiments

In this section, we empirically validate the ability of GCARe in fair graph condensation. We first introduce the experiment settings, then compare GCARe with baselines; finally, we analyze the other characteristics of our method.

5.1. Experiment Settings

5.1.1. Datasets

We adopt four transductive node classification graph datasets for our experiments: Cora [33], Ogbn-arxiv [12], Credit-defaulter [34], and Recidivism [34]. Statistics of the three datasets can be found in Table 1. For Cora, we condensed the graph to 35 nodes, resulting in a compression rate of 1.29%. For Ogbn-arxiv, we reduced the graph size to 90 nodes, resulting in a compression rate of 0.053%. For Credit-defaulter, the node number and compression were 120 and 0.4% respectively, while for Recidivism they were 200 and 1.06%.

Table 1. Dataset Statistics. The first two datasets were divided into subgroups according to node degrees, while the last two were divided according to their sensitive attributes.

Dataset	#Nodes	#Edges	#Classes	#Features	Train/Valid/Test
Cora	2708	5429	7	1433	140/500/1000
Ogbn-arxiv	169,343	1,166,243	40	128	90,941/29,799/48,603
Credit-defaulter	30,000	2,873,716	2	13	6000/7500/7500
Recidivism	18,876	642,616	2	18	1000/4719/4719

5.1.2. Subgroup Division Evaluation Metrics

In this paper, we consider fairness with respect to both node degrees and attributes. For Cora, the nodes were divided, according to their degrees, into three subgroups. The thresholds of the degrees used to make divisions were chosen to generate subgroups of comparable sizes. The subgroup sizes were (71, 48, 21) for the training nodes, and (400, 348, 252) for the testing nodes. For Ogbn-arxiv, the nodes were divided into five subgroups in a similar manner. The subgroup sizes were (25,134, 16,597, 18,607, 17,183, 13,420) for the training nodes, and (11,576, 10,557, 10,111, 9330, 7029) for the testing nodes. For Credit-

defaulter, we followed [34] in using the binary sensitive attribute Age to split the subgroups, whereby NoDefaultNextMonth was used as the label. The subgroups sizes were $(5442, 578)$ for the training nodes, and $(6811, 689)$ for the testing nodes. For Recidivism, we also followed the preprocessing method of [34]. The attribute *Race* was used to split the nodes into two subgroups. The group sizes were $(477, 523)$ for the training nodes, and $(2303, 2416)$ for the testing nodes.

For the degree bias, we evaluated the GNN's fairness with respect to two metrics: Δ_{acc} was set as the accuracy gap between the most and least advantaged subgroups, and σ_{acc} was set as the standard deviation of the accuracies across all subgroups. By denoting the accuracy of group i as $a_i, \forall i \in \mathcal{G}$, the two metrics were computed as follows:

$$\Delta_{acc} := \max_{i \in \mathcal{G}} a_i - \min_{j \in \mathcal{G}} a_j \tag{3}$$

$$\sigma_{acc} := \sqrt{\mathbb{E}[(a_i - \bar{a})^2]} \tag{4}$$

For attribute bias, we focused on the fairness metrics commonly adopted, namely statistical parity and equal opportunity [26], which evaluate the statistical dependencies between predictions and attributes. Suppose a binary attribute is denoted as s, and prediction as $\hat{y}$, then these two metrics are defined as follows:

$$\Delta_{SP} := \left| p(\hat{y} = 1 | s = 0) - p(\hat{y} = 1 | s = 1) \right| \tag{5}$$

$$\Delta_{EO} := \left| p(\hat{y} = 1 | y = 1, s = 0) - p(\hat{y} = 1 | y = 1, s = 1) \right| \tag{6}$$

5.1.3. Baselines

We present a comparative analysis of our algorithm, GCARe, with two state-of-the-art approaches: GCond [11] and DosCond [10]. GCond learns condensed graph representations by aligning the model gradients with respect to the inputs from both the original and the condensed graph. Furthermore, it trains condensation GNNs at each iteration during the condensation process. In contrast, DosCond enhances this approach by eliminating the training of condensation GNNs, thus leading to a remarkable acceleration. Our proposed method, GCARe, offers a general approach to regularize the condensation process and to achieve a fair graph condensation. To demonstrate the effectiveness of GCARe, we have implemented it on both GCond and DosCond. The results were then compared with those that were obtained from the vanilla methods.

We also compared against fairness constraint regularization [26]. Concretely, for a task with k subgroups and c classes, the generalized statistical parity regularization was defined as follows:

$$\text{sp\_reg} = \frac{1}{k} \sum_{i=1}^{k} \max_{y_j \in [c]} |P(\hat{y} = y_j) - P(\hat{y} = y_j | s = s_i)|$$

where $\hat{y}$ is the predicted label. When $k = c = 2$, this formulation reduces to Equation (5). We replaced the adversarial regularization term in GCARe with this statistical parity regularization term for this variant.

5.1.4. Hyperparameters

We followed the hyperparameter settings in [10,11] for the baseline results on Cora and Ogbn-arxiv, while on Credit-defaulter and Recidivism we searched for lr_adj and lr_feat in the range $[0.0001, 0.001, 0.01]$. For GCARe, we fixed all other hyperparameters to be the same as the baselines, and only tuned one new hyperparameter to be introduced by our method, i.e., the regularization weight λ. For all datasets, we adopted SGC [9] as the upstream GNN (used for condensation), and GCN [33] as the downstream GNN (used for training and evaluation).

5.2. Main Results and Analysis

The comparison between GCARe and the baselines is shown in Table 2. All experiments were run with five different random seeds, and the average values are reported. The results of the training GNNs on full data are also reported as an upper bound of our method. We implemented both our method GCARe and the statistical parity regularization based on GCond or DosCond, and they are referred to as +Ours and +SP, respectively.

Table 2. Results from the Cora, Ogbn-arxiv, and Credit-defaulter datasets. *acc* stands for overall accuracy; Δ_{acc} for the accuracy gap between the highest and lowest subgroups; σ_{acc} for the std. dev. of the accuracies among all subgroups; Δ_{SP} for statistical parity; and Δ_{EO} for equal opportunity. GCARe achieved superior performances regarding both accuracy and fairness metrics. All the numbers were averaged over five random seeds. The best results are marked in bold.

Dataset	Metric	GCond	+SP	+Ours	DosCond	+SP	+Ours	Whole Dataset
Cora	$acc \uparrow$	80.98	80.80	**81.03**	80.06	80.25	**80.32**	81.92
	$\Delta_{acc} \downarrow$	9.80	9.83	**9.56**	9.43	9.31	**9.07**	8.95
	$\sigma_{acc} \downarrow$	4.50	4.41	**4.28**	4.18	**4.14**	4.15	3.96
Ogbn-arxiv	$acc \uparrow$	59.23	58.89	**59.37**	**58.61**	58.08	57.81	71.21
	$\Delta_{acc} \downarrow$	31.36	30.75	**30.22**	31.79	30.53	**29.85**	22.04
	$\sigma_{acc} \downarrow$	10.98	10.75	**10.71**	11.15	10.63	**10.35**	7.57
Credit-defaulter	$acc \uparrow$	74.53	**74.60**	74.37	74.30	74.66	**74.96**	75.22
	$\Delta_{SP} \downarrow$	11.21	11.20	**10.98**	13.99	**10.10**	10.90	10.23
	$\Delta_{EO} \downarrow$	8.87	8.91	**8.74**	11.39	**8.15**	8.82	8.21
Recidivism	$acc \uparrow$	80.04	**82.43**	81.97	80.08	80.88	**81.16**	81.01
	$\Delta_{SP} \downarrow$	4.47	4.46	**3.98**	4.76	4.90	**4.68**	3.59
	$\Delta_{EO} \downarrow$	3.40	2.76	**2.52**	3.45	3.61	**3.43**	1.21

5.2.1. Fairness

The results from all four datasets demonstrated that GCARe can generate condensed graphs that accommodate fair GNN training. Note that all the fairness metrics reported here are better the lower they are.

Impressively, our technique outperforms both GCond and DosCond across all datasets. When evaluating degree bias, the data from the Cora and Ogbn-arxiv datasets revealed that GCARe achieves lower Δ_{acc} and σ_{acc} compared to both GCond and DosCond. Similarly, in terms of attribute bias, GCARe improves the Δ_{SP} and Δ_{EO} from the Credit-defaulter and Recidivism datasets. It is worth noting that when our method is combined with GCond (GCond + Ours), the Δ_{SP} value was reduced from 4.47 to 3.98, and the Δ_{EO} decreased from 3.40 to 2.52 on the Credit-defaulter dataset. This equates to improvements of 10.96% and 25.88%, respectively. Variants utilizing statistical parity regularization also exhibit significant improvements over GCond and DosCond, although they are generally not as effective as GCARe. These findings underscore the ability of our method to effectively enhance the fairness of GNNs that are trained on condensed graphs.

5.2.2. Node Classification

Previous works often observe a trade-off between model fairness and accuracy [35,36]. Our method not only benefits fair graph condensation, but also improves the overall performance in node classification tasks. As Table 2 shows, GCARe achieves a higher accuracy over GCond and DosCond in almost all cases, and it also takes a further step toward recovering the potentials of the original large graphs. In particular, on the Recidivism dataset, GCARe even achieved a higher accuracy than training on the full graph (81.97 vs. 81.01). In addition, the statistical parity regularization approach also surpassed the baselines. These results demonstrate that our method provides better representations for condensed graphs.

5.3. Cross Architecture Generalization

In all previous experiments, we fixed the condensation model to be SGC, and we evaluated the performances of GCN on the condensed graphs. However, it is important to assess whether the condensed graphs can accommodate the training of other types of GNNs. In this section, we fixed the condensed graphs to the same as those reported in Table 2, but we varied the GNNs used for evaluation among GCN [33], GraphSage [37], SGC [9], APPNP [38], and Cheby [39]. We also included a two-layer feedforward neural network, which uses only node features for prediction and is referred to as MLP. Specifically, once the condensation process was finished, we utilized the same condensed graph and trained various GNN architectures on it. We implemented GCARe with DosCond as the backbone on the Credit-defaulter dataset, and choose GCN, GraphSage, SGC, MLP, APPNP, and Cheby as the GNNs to evaluate. The results are shown in Table 3. We find that GCARe can improve fairness and overall accuracy for a wide range of GNN architectures. This demonstrates that our method generalizes well across different GNN models and can be applied in various scenarios.

Table 3. Cross-architecture evaluation on the Credit-defaulter dataset. Different downstream GNNs were evaluated on the graphs condensed by DosCond and GCARe. GCARe achieved superior performances in both utility and fairness, and it showed better cross-architecture generalization on all GNN variants. The best results are marked in bold.

Method	Metric	GCN	GraphSage	SGC	MLP	APPNP	Cheby
DosCond	$acc \uparrow$	74.30	**74.73**	74.63	74.64	75.11	77.77
	$\Delta_{SP} \downarrow$	13.99	10.35	**14.69**	11.65	15.11	11.60
	$\Delta_{EO} \downarrow$	11.39	8.34	12.00	9.61	12.65	8.51
GCARe	$acc \uparrow$	**74.96**	73.97	**75.82**	**75.01**	**75.56**	**78.12**
	$\Delta_{SP} \downarrow$	**10.90**	**7.47**	14.76	**8.70**	**13.11**	**6.54**
	$\Delta_{EO} \downarrow$	**8.82**	**5.99**	**11.83**	**6.79**	**10.48**	**4.37**

6. Conclusions

Graph condensation is an important technique for efficient GNN training and deployment. In this paper, we showed the limitations of previous graph condensation methods in exacerbating subgroup unfairness through empirical evaluations. Starting from the intuition that condensation GNNs act as an information bottleneck, we propose GCARe, which utilizes adversarial learning to regularize condensation GNNs during the condensation process. Extensive experiments demonstrated GCARe's ability in simultaneously improving accuracy and fairness for condensed graphs.

Limitations and Broader Impact

Adversarial learning suffers from unstable training and mode collapse, which can increase the difficulty of hyperparameter tuning in graph condensations. In addition, when there are many subgroups, class-based sampling may not sample all of the nodes from certain groups. This hinders the applicability of GCARe in settings of more fine-grained subgroup divisions. Future work should be conducted on analyzing the adversarial robustness and privacy of condensed graphs to take one step closer toward trustworthy AI.

Author Contributions: Conceptualization, R.M., W.F. and Q.L.; methodology, validation, writing—original draft preparation, R.M.; writing—review and editing, R.M., W.F. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duvenaud, D.K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In Proceedings of the 29th Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
2. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph Neural Networks for Social Recommendation. In Proceedings of the World Wide Web Conference, Las Vegas, NV, USA, 29 July–1 August 2019; pp. 417–426.
3. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
4. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, London, UK, 19–23 August 2018; pp. 974–983.
5. You, J.; Liu, B.; Ying, R.; Pande, V.; Leskovec, J. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 6412–6422.
6. Patterson, D.; Gonzalez, J.; Hölzle, U.; Le, Q.; Liang, C.; Munguia, L.M.; Rothchild, D.; So, D.R.; Texier, M.; Dean, J. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer* **2022**, *55*, 18–28. [CrossRef]
7. Zoph, B.; Le, Q. Neural Architecture Search with Reinforcement Learning. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
8. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 639–648.
9. Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying Graph Convolutional Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6861–6871.
10. Jin, W.; Tang, X.; Jiang, H.; Li, Z.; Zhang, D.; Tang, J.; Yin, B. Condensing graphs via one-step gradient matching. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 720–730.
11. Jin, W.; Zhao, L.; Zhang, S.; Liu, Y.; Tang, J.; Shah, N. Graph Condensation for Graph Neural Networks. In Proceedings of the 10th International Conference on Learning Representations, Virtual, 25–29 April 2022.
12. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020.
13. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental Classifier and Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5533–5542.
14. Zhao, B.; Mopuri, K.R.; Bilen, H. Dataset Condensation with Gradient Matching. In Proceedings of the 9th International Conference on Learning Representations, Virtual, 3–7 May 2021.
15. Ling, H.; Jiang, Z.; Luo, Y.; Ji, S.; Zou, N. Learning Fair Graph Representations via Automated Data Augmentations. In Proceedings of the 11th International Conference on Learning Representations, Kobe, Japan, 2–7 April 2023.
16. Wang, R.; Wang, X.; Shi, C.; Song, L. Uncovering the Structural Fairness in Graph Contrastive Learning. In Proceedings of the 36th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
17. Wang, X.; Wu, Z.; Lian, L.; Yu, S.X. Debiased Learning From Naturally Imbalanced Pseudo-Labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14647–14657.
18. Wang, T.; Zhu, J.; Torralba, A.; Efros, A.A. Dataset Distillation. *arXiv* **2018**, arXiv:1811.10959.
19. Esipova, M.S.; Ghomi, A.A.; Luo, Y.; Cresswell, J.C. Disparate Impact in Differential Privacy from Gradient Misalignment. In Proceedings of the 11th International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
20. Koh, P.W.; Sagawa, S.; Marklund, H.; Xie, S.M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R.L.; Gao, I.; et al. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Volume 139, pp. 5637–5664.
21. Zhu, Z.; Luo, T.; Liu, Y. The Rich Get Richer: Disparate Impact of Semi-Supervised Learning. In Proceedings of the 10th International Conference on Learning Representations, Virtually, 25–29 April 2022.
22. Piratla, V.; Netrapalli, P.; Sarawagi, S. Focus on the Common Good: Group Distributional Robustness Follows. In Proceedings of the 10th International Conference on Learning Representations, Virtually, 25–29 April 2022.

23. Tang, X.; Yao, H.; Sun, Y.; Wang, Y.; Tang, J.; Aggarwal, C.; Mitra, P.; Wang, S. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management, Online, 23 October 2020 ; pp. 1435–1444.
24. Dai, E.; Wang, S. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual, 8–12 March 2021; pp. 680–688.
25. Spinelli, I.; Scardapane, S.; Hussain, A.; Uncini, A. FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning. *IEEE Trans. Artif. Intell.* **2021**, *3*, 344–354. [CrossRef]
26. Dai, E.; Zhao, T.; Zhu, H.; Xu, J.; Guo, Z.; Liu, H.; Tang, J.; Wang, S. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. *arXiv* **2022**, arXiv:2204.08570.
27. Fan, W.; Zhao, X.; Chen, X.; Su, J.; Gao, J.; Wang, L.; Liu, Q.; Wang, Y.; Xu, H.; Chen, L.; et al. A Comprehensive Survey on Trustworthy Recommender Systems. *arXiv* **2022**, arXiv:2209.10117.
28. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual, 6–10 July 2020; pp. 1150–1160.
29. Kojaku, S.; Yoon, J.; Constantino, I.; Ahn, Y.Y. Residual2Vec: Debiasing graph embedding with random graphs. In Proceedings of the 35th Conference on Neural Information Processing Systems, Online, 6–14 December 2021.
30. Ma, J.; Guo, R.; Wan, M.; Yang, L.; Zhang, A.; Li, J. Learning Fair Node Representations with Graph Counterfactual Fairness. In Proceedings of the 15th ACM International Conference on Web Search and Data Mining, Virtual, 21–25 February 2022; pp. 695–703.
31. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
32. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
33. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
34. Agarwal, C.; Lakkaraju, H.; Zitnik, M. Towards a Unified Framework for Fair and Stable Graph Representation Learning. In Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence, Online, 27–30 July 2021.
35. Menon, A.K.; Williamson, R.C. The Cost of Fairness in Binary Classification. In Proceedings of the 1st Conference on Fairness, Accountability and Transparency, New York, NY, USA, 23–24 February 2018; pp. 107–118.
36. Dutta, S.; Wei, D.; Yueksel, H.; Chen, P.Y.; Liu, S.; Varshney, K. Is There a Trade-Off between Fairness and Accuracy? A Perspective Using Mismatched Hypothesis Testing. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020; Volume 119, pp. 2803–2813.
37. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
38. Gasteiger, J.; Bojchevski, A.; Günnemann, S. Combining Neural Networks with Personalized PageRank for Classification on Graphs. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
39. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.