Virtual Adversarial Training on Graph Convolutional Networks in Node Classification

Ke Sun[1], Zhouchen Lin[2], Hantao Guo[1], and Zhanxing Zhu[1,3,4(✉)]

[1] Center for Data Science, Peking University, Beijing, China
{ajksunke,guohantao,zhanxing.zhu}@pku.edu.cn
[2] Key Laboratory of Machine Perception (MOE), School of EECS,
Peking University, Beijing, China
zlin@pku.edu.cn
[3] School of Mathematical Science, Peking University, Beijing, China
[4] Beijing Institute of Big Data Research (BIBDR), Beijing, China

Abstract. The effectiveness of Graph Convolutional Networks (GCNs) has been demonstrated in a wide range of graph-based machine learning tasks. However, the update of parameters in GCNs is only from labeled nodes, lacking the utilization of unlabeled data. In this paper, we apply Virtual Adversarial Training (VAT), an adversarial regularization method based on both labeled and unlabeled data, on the supervised loss of GCN to enhance its generalization performance. By imposing virtually adversarial smoothness on the posterior distribution in semi-supervised learning, VAT yields an improvement on the performance of GCNs. In addition, due to the difference of property in features, we perturb virtual adversarial perturbations on sparse and dense features, resulting in GCN Sparse VAT (GCNSVAT) and GCN Dense VAT (GCNDVAT) algorithms, respectively. Extensive experiments verify the effectiveness of our two methods across different training sizes. Our work paves the way towards better understanding the direction of improvement on GCNs in the future.

Keywords: Graph Convolutional Networks · Virtual Adversarial Training

1 Introduction

Recently, research of analyzing graphs with machine learning has received more and more attention, mainly focusing on node classification [8], link prediction [17] and clustering tasks [5]. Graph convolutions, as the transformation of traditional

---

The first author is a student.

convolutions from Euclidean domain to non-Euclidean domain, have been leveraged to design Graph Neural Networks to deal with a wide range of graph-based machine learning tasks.

Graph Convolutional Networks (GCNs) [8] generalize convolutional neural networks (CNNs) to graph structured data from the perspective of spectral theory based on prior works [1,4]. It has been demonstrated that GCN and its variants [2,3,7,15] significantly outperform traditional multi-layer perceptron (MLP) models and prior graph embedding approaches [6,12,14].

However, there are still many deficits on GCNs, thus in this paper we propose to apply VAT on GCNs to tackle these drawbacks of GCNs. Particularly, we firstly highlight the importance of VAT on GCNs from the following aspects, which construct the motivation of our approaches.

Lacking the Leverage of Unlabeled Data for GCNs. The optimization of GCNs is solely based on the labeled nodes. Concretely speaking, GCNs directly distribute gradient information over the entire labeled set of nodes from the supervised loss. Due to the lack of loss on unlabeled data, the parameters that are not involved in the receptive field may not be updated [2], resulting in the inefficiency of information propagation of GCNs.

Effect of Regularization in Semi-supervised Learning. Regularization plays a crucial role in semi-supervised learning including graph-based learning tasks. On the one hand, by introducing regularization, a model can make full use of unlabeled data, thus enhancing the performance in semi-supervised learning. On the other hand, regularization can also be regarded as prior knowledge that can smooth the posterior output. For GCN model, a good regularization can not only leverage the unlabeled data to refine its optimization, but only benefit the performance of GCNs, resulting in a improved generalization performance.

Virtual Adversarial Regularization on GCNs. Virtual Adversarial Training (VAT) [11] smartly performs adversarial training without label information to impose a local smoothness on the classifier, which is especially beneficial to semi-supervised learning. In particular, VAT endeavors to smooth the model anisotropically in the direction in which the model is the most sensitive, i.e., the adversarial direction, to improve the generalization performance of a model. In addition, the existence of robustness issue in GCNs has been explored in recent works [18,19], allowing VAT on graph-based learning task.

Due to the fact that VAT has been successfully applied on semi-supervised image classification [11,16] and text classification [10], a natural question could be asked: *Can we utilize the efficacy of VAT to improve the performance of GCNs in semi-supervised node classification?*

Following this motivation, in our paper, we formally introduce VAT regularization on the original supervised loss of GCNs in semi-supervised node classification task. Concretely speaking, firstly, a detailed analysis of GCNs focusing on the first-order approximation of local spectral convolutions is provided to demystify how GCNs work in semi-supervised learning. Moreover, based on the motivation described above, we elaborate the process of applying VAT on GCNs

in a theoretical way by additionally imposing virtual adversarial loss on the basic loss of GCNs, resulting in GCNVAT algorithm framework. Next, due to the sparse property of node features, in the realization of our method, we actually add virtual adversarial perturbations on sparse and dense features, respectively, and attain the GCNSVAT and GCNDVAT algorithms. Finally, in the experimental part, we demonstrate the effectiveness of the two approaches under different training sizes and refine a theoretical analysis on the sensitivity to the hyper-parameters on VAT, facilitating us to apply our approaches in real applications involving graph-based machine learning tasks. In summary, the contributions of the paper are listed below:

– To the best of our knowledge, we are the first to focus on applying better regularization on original GCN to refine its generalization performance.
– We are the first to successfully transfer the efficacy of Virtual Adversarial Training (VAT) to the semi-supervised node classification on graphs and point out the difference compared with image and text classification setting.
– We refine the sensitivity analysis of hyper-parameters in GCNSVAT and GCNDVAT algorithms, facilitating the deployment of our methods in real scenarios.

2 GCNs with Virtual Adversarial Training

In this section, we will elaborate how GCNs work in semi-supervised learning and how to utilize the virtual adversarial training to smooth the posterior distribution of GCNs.

2.1 Semi-supervised Classification with GCNs

Firstly, we denote a graph by $G = (V, E)$, where V is the vertex set and E is the edge set. X and A are the features and adjacent matrix of the graph, respectively and $D = \text{diag}(d_1, d_2, \cdots, d_n)$ denotes the degree matrix of A, where $d_i = \sum_j a_{ij}$ is the degree of vertex i.

First-Order Approximation. GCNs are based on the graph spectral theory. For efficient computation, [4] approximate the spectral filter g_θ with Chebyshev polynomials up to K^{th} order:

$$g_{\theta'}(\Lambda) = \sum_{k=0}^{K-1} \theta'_k T_k(\Lambda), \tag{1}$$

where Λ is the eigenvalues matrix of normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. T_k is the Chebyshev polynomials and θ'_k is a vector of Chebyshev coefficients. Further, [8] simplified this model by limiting $K = 1$ and then the first-order approximation of spectral graph convolution is defined as:

$$g_\theta \star x = \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x, \tag{2}$$

where θ is the only Chebyshev coefficients left. Through the normalization trick, the final form of graph convolutional networks with two layers in GCNs [8] is:

$$Z = f(X, A) = \text{softmax}(\hat{A} \text{ ReLU}(\hat{A}XW^{(0)})W^{(1)}), \tag{3}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tilde{A} = A+I$. $\tilde{D}$ is the degree matrix of $\tilde{A}$. Z is the obtained embedding matrix from nodes, $W^{(0)}$ is the input-to-hidden weight matrix and $W^{(1)}$ is the hidden-to-output weight matrix.

Optimization. Finally, the loss function is defined as the cross entropy error over all labeled nodes:

$$\mathcal{L}_0 = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} Z_{lf}, \tag{4}$$

where $\mathcal{Y}_L$ is the set of node indices that have labels. In fact, the performance of GCNs heavily depends on the efficiency of this Laplacian Smoothing Convolutions, which has been demonstrated in [8,9]. Therefore, how to design a good regularization to smooth the posterior distribution of GCNs plays a crucial role for the improvement of performance for GCNs.

2.2 Virtual Adversarial Training in GCNs

Virtual Adversarial Training (VAT) [11] is a regularization method that trains the output distribution to be isotropically smooth around each input data point by selectively smoothing the model in its most anisotropic direction, namely adversarial direction. In this section, we apply VAT on GCNs to smooth the posterior distribution of GCNs.

Assumptions. Firstly, both VAT and GCNs mainly focus on semi-supervised setting, in which two assumptions should be implicitly met [16]:

– **Manifold Assumption.** The observed data x presented in high dimensional space is with high probability concentrated in the vicinity of some underlying manifold with much lower dimensional space.
– **Smoothness Assumption.** If two points $x_1, x_2 \in \mathcal{M}$ are close in manifold distance, then the conditional probability $p(y|x_1)$ and $p(y|x_2)$ should be similar. In other words, the true classifier, or the true condition distribution $p(y|x)$ varies smoothly along the underlying manifold $\mathcal{M}$.

In the node classification task, GCNs, which involve the graph embedding process, also implicitly conform to these assumptions. There is underlying manifold in the process of graph embedding and the conditional distribution of embedding vectors are expected to vary smoothly along the underlying manifold. In this way, we are capable of utilizing VAT to smooth the embedding of nodes in the adversarial direction to improve the generalization of GCNs.

Difference of VAT on Graph and Image, Text. Traditional VAT [11] is proposed on image classification while VAT on text classification [10] is applied

on word embedding vectors of each word. For VAT on graphs, we simply apply VAT on the features of nodes for easy implementation. Additionally, another obvious difference lies in that the relation between each node is not independent for the node classification task compared with image and text classification. The classification result of each node not only depends on the feature itself but also the features of its neighbors, resulting in the *Propagation Effect* of perturbations on feature of each node. We use $\mathcal{D}_l$ and $\mathcal{D}_{ul}$ to denote dataset with labeled nodes and unlabeled nodes respectively. $\overline{x}$ represents features excluding feature x of current node.

Adversarial Training in GCNs. Here we formally define the adversarial training in GCNs, where adversarial perturbations are solely added on features of labeled nodes:

$$\min_{\theta} \max_{r, \|r\| \leq \epsilon} D\left[q(y|x_l, \overline{x}, A), p(y|x_l + r, \overline{x}, A; \theta) \right], \tag{5}$$

where $D[q, p]$ measures the divergence between two distributions q and p. $q(y|x_l, \overline{X}, A)$ is the true distribution of output labels, usually one hot vector $h(y; y_l)$ and $p(y|x_l + r, \overline{x}, A) = f(X, A)$ denotes the predicted distribution by GCNs. x_l represents the feature of current labeled node and r represents the adversarial perturbation on the feature x_l. When the true distribution is denoted by one hot vector $h(y; y_l)$, the perturbation r_{adv} in L_2 norm can be linearly approximated:

$$r_{\mathrm{adv}} \approx \epsilon \frac{g}{\|g\|_2}, \text{ where } g = \nabla_{x_l} D\left[h(y; y_l), p(y|x_l, \overline{x}, A; \theta) \right]. \tag{6}$$

Virtual Adversarial Loss. In order to utilize the unlabeled data, we are expected to evaluate the true conditional probability $q(y|x_l, \overline{x}, A)$. Therefore, we use the current estimate $p(y|x, \overline{x}, A; \hat{\theta})$ in place of $q(y|x, \overline{x}, A)$.

$$\min_{\theta} \max_{r, \|r\| \leq \epsilon} D\left[p(y|x, \overline{x}, A; \hat{\theta}), p(y|x + r, \overline{x}, A; \theta) \right] \tag{7}$$

Then virtual adversarial regularization is constructed from inner max loss:

$$\mathcal{R}_{\mathrm{vadv}}(x, \mathcal{D}_l, \mathcal{D}_{ul}, \theta) = \max_{r, \|r\| \leq \epsilon} D\left[p(y|x, \overline{x}, A; \hat{\theta}), p(y|x + r, \overline{x}, A; \theta) \right] \tag{8}$$

The final regularization term we propose in this study is the average of $\mathcal{R}_{\mathrm{vadv}}(x, \mathcal{D}_l, \mathcal{D}_{ul}, \theta)$ over all input nodes:

$$\mathbb{E}_{x \sim \mathcal{D}} \mathcal{R}_{\mathrm{vadv}} = \frac{1}{N_l + N_{ul}} \sum_{x \in \mathcal{D}_l, \mathcal{D}_{ul}} \mathcal{R}_{\mathrm{vadv}}(x, \mathcal{D}_l, \mathcal{D}_{ul}, \theta) \tag{9}$$

Virtual Adversarial Training. The full objective function is thus given by:

$$\min_{\theta} \mathcal{L}_0 + \alpha \mathbb{E}_{x \sim \mathcal{D}} \mathcal{R}_{\mathrm{vadv}}, \tag{10}$$

where $\mathcal{L}_0$ is constructed from labeled nodes in GCNs, α denotes the regularization coefficient and VAT regularization is crafted from both labeled and unlabeled nodes.

2.3   Fast Approximation of VAT in GCNs

The key of VAT in GCNs is the approximation of r_{vadv} where

$$r_{\text{vadv}} = \underset{r, \|r\| \le \epsilon}{\arg\max} D\left[p(y|x, \overline{x}, A; \hat{\theta}), p(y|x + r, \overline{x}, A; \hat{\theta})\right]. \tag{11}$$

Second-Order Approximation. Just like the situation in traditional VAT, the evaluation of GCNs with VAT cannot be performed with the linear approximation since first-order approximation equals zero [10]. Therefore, a second-order approximation is needed:

$$D(r, x, \overline{x}, A; \hat{\theta}) \approx \frac{1}{2} r^T H(x, \overline{x}, A; \hat{\theta}) r, \tag{12}$$

where $H(x, \overline{x}, A; \hat{\theta}) := \nabla_r^2 D(r, x, \overline{x}, A; \hat{\theta})|_{r=0}$. Then the evaluation of r_{vadv} can be approximated by:

$$r_{\text{vadv}} \approx \underset{r}{\arg\max}\{r^T H(x, \hat{\theta}) r; \ \|r\|_2 \le \epsilon\} = \overline{\epsilon u(x, \overline{x}, A; \hat{\theta})}, \tag{13}$$

where $u(x, \overline{x}, A; \hat{\theta})$ is the first dominant eigenvector of $H(x, \overline{x}, A; \hat{\theta})$ with magnitude 1.

Power Iteration and Finite Difference Approximation. After power interation and finite difference approximation mentioned in [10], the final approximation of r_{vadv} is:

$$r_{\text{vadv}} \approx \epsilon \frac{g}{\|g\|_2}, \text{where } g = \nabla_r D\left[p(y|x, \overline{x}, A; \hat{\theta}), p(y|x + r, \overline{x}, A; \hat{\theta})\right]\Big|_{r=\xi d}. \tag{14}$$

3   Algorithm

In this section, we will elaborate our Graph Convolutional Networks with Virtual Adversarial Training (GCNVAT) Algorithm. Algorithm 1 summarizes the procedures of the computation of mini-batch SGD for GCNs with VAT algorithm.

Our GCNVAT Algorithm Framework is economical in computation since the derivative of the full objective function can be computed with at most three sets of propagation in total. Specifically speaking, firstly, by initializing the random unit vector $d^{(i)}$ in mini-batch and computing the gradient of divergence between predicted distribution of GCNs and that with the initial perturbation, we can evaluate the fast approximated r_{vadv}, which is involved in the first set of back propagation. Secondly, after the computation of r_{vadv}, we are able to compute the average virtual adversarial loss in the mini-batch and optimize this loss under fixed r_{vadv}, which incorporates the second set of back propagation. Finally, the third back propagation is related to the original supervised loss based on labeled nodes in GCNs. All in all, by this GCNs with VAT algorithm including three sets of back propagation, we are capable of imposing the local adversarial

Algorithm 1. Mini-batch SGD for GCNVAT Framework

Input: Features Matrix X, Adjacent Matrix A. Graph Convolution Network f_θ

Output: Graph Embedding $Z = f_\theta(X, A)$

1: Choose M samples of $x^{(i)}(i = 1, \ldots, M)$ from dataset $\mathcal{D}$ at random.
2: Compute the predicted distribution of current GCNs:

$$p(y|x_l, \overline{x}, A; \hat{\theta}) \leftarrow f_{\hat{\theta}}(X, A)$$

3: **% Step 1: Fast Approximation of r_{vadv}**
4: Generate a random unit vector $d^{(i)} \in R^I$ using an iid Gaussian distribution.
5: Calculate r_{vadv} via taking the gradient of D with respect to r on $r = \xi d^{(i)}$ on each input data point $x^{(i)}$:

$$g^{(i)} \leftarrow \nabla_r D\left[p(y|x^{(i)}, \overline{x}, A; \hat{\theta}), p(y|x^{(i)} + r, \overline{x}, A; \hat{\theta})\right]\Big|_{r=\xi d^{(i)}}$$

6: Evaluation of r_{vadv}:

$$r_{\text{vadv}}^{(i)} \leftarrow \epsilon g^{(i)}/\|g^{(i)}\|_2$$

7: **% Step 2: Evaluation of Virtual Adversarial Loss**

$$\mathbb{E}_{x \sim \mathcal{D}} \mathcal{R}_{\text{vadv}} = \nabla_\theta \left(\frac{1}{M} \sum_{i=1}^{M} D\left[p(y|x^{(i)}; \hat{\theta}), p(y|x^{(i)} + r_{\text{vadv}}^{(i)}; \theta)\right]\right)\Big|_{\theta=\hat{\theta}}$$

8: **% Step 3: Virtual Adversarial Training**
9: Compute the supervised loss $\mathcal{L}_0$ of GCNs:

$$\mathcal{L}_0 = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} Z_{lf}$$

10: Update θ by optimizing the full objective function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_0 + \alpha \mathbb{E}_{x \sim \mathcal{D}} \mathcal{R}_{\text{vadv}}$$

regularization on the original supervised loss of GCNs through smoothing the posterior distribution of the model in the most adversarial direction, thereby improving the generalization of original GCNs.

GCNSVAT and GCNDVAT. In the real scenarios, there are usually sparse features for each node especially for a large graph, which are involved in the computation of sparse tensor. In this case, in the implementation of our GCNVAT algorithm framework, we customize two similar GCNVAT methods for different properties of node features. For GCN Sparse VAT (GCNSVAT), we only apply virtual adversarial perturbations on the specific sparse elements in feature of each node, which may save much computation time especially for high-dimensional feature vectors. For GCN Dense VAT (GCNDVAT), we actually perturb each element in feature by transforming the sparse feature matrix to a dense one.

4   Experiments

In the experimental part, we conduct extensive experiments to demonstrate the effectiveness of our GCNSVAT and GCNDVAT algorithms. Firstly, we test the performance of both algorithms under different label rates compared with the original GCN. Then we make another comparison under the standard semi-supervised setting with other state-of-the-art approaches. Finally, a sensitivity analysis of hyper-parameters is provided for broad deployment of our method in real applications.

Experimental Setup. For the graph dataset, we select the three commonly used citation networks: CiteSeer, Cora and PubMed [13]. Dataset statistics are summarized in Table 1. For all methods involved in GCNs, we use the same hyper-parameters as in [8]: learning rate of 0.01, 0.5 dropout rate, 2 convolutional layers, and 16 hidden units without validation set for fair comparison. As for the hyper-parameters, we fix regularization coefficient $\alpha = 1.0$ and only change the perturbation magnitude ϵ to control the regularization effect under different training sizes, which is further discussed later in the sensitivity analysis part. All the results are the mean accuracy of 10 runs to avoid stochastic effect.

Table 1. Dataset statistics

Dataset	Nodes	Edges	Classes	Features	Label rate
CiteSeer	3327	4732	6	3703	3.6%
Cora	2708	5429	7	1433	5.2%
PubMed	19717	44338	3	500	0.3%

4.1   Effect Under Different Training Sizes

To verify the consistent effectiveness of our two methods on the improvement of generalization performance, we compare GCNSVAT and GCNDVAT algorithms with original GCN method [8] under different training sizes across the three datasets and the results can be observed in Fig. 1.

   As illustrated in Fig. 1, GCNSVAT (the red line) and GCNDVAT (the blue line) outperform original GCN (the black line) consistently under all tested label rates. Actually, it is important to note that with the increasing of label rates, the regularization effect imposed by VAT on GCNs diminishes in both approaches since the improvement from regularization based on unlabeled data is decreasing. In other words, the superior performance of GCN with Virtual Adversarial Training are especially significant when there are few training sizes. Fortunately, in real scenarios, it is common to observe graphs with a small number of labeled nodes, thereby our algorithms are especially practical in these applications.
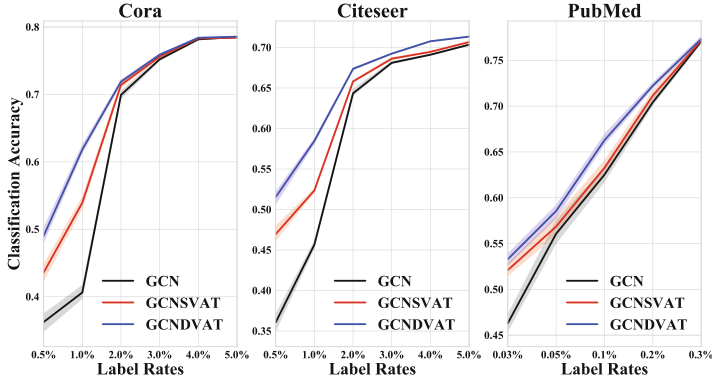
Fig. 1. Classification accuracies of GCNSVAT and GCNDVAT algorithms compared with GCN across three datasets. (Color figure online)

Choice of GCNSVAT and GCNDVAT. GCNDVAT performs consistently better in comparison with GCNSVAT even though GCNDVAT requires extra computation cost related to perturbations in the entire feature space. As for the reason, we argue that continuous perturbations in features facilitate the effect of VAT than discrete perturbations in sparse features. However, in the scenarios where the graph are large-scaled and their features are sparse, it is more appropriate to utilize GCNSVAT from the perspective of economical computation.

Table 2. Classification accuracies on Cora with different label rates. Numbers in bracket are the standard deviation of accuracies.

	Rates	0.5%	1%	2%	3%	4%	5%
Cora	GCN	36.2	40.6	69.0	75.2	78.2	78.4
		(0.11)	(0.08)	(0.05)	(0.03)	(0.1)	(0.01)
	GCN SVAT	43.6	53.9	71.4	75.6	78.3	78.5
		(0.10)	(0.08)	(0.05)	(0.02)	(0.01)	(0.01)
	GCN DVAT	**49.0**	**61.8**	**71.9**	**75.9**	**78.4**	**78.6**
		(0.10)	**(0.06)**	**(0.03)**	**(0.02)**	**(0.01)**	**(0.01)**

More specifically, we list the detailed performances of GCNSVAT and GCND-VAT compared with original GCN under different label rates, which are exhibited in Tables 2, 3 and 4, respectively. We report the mean accuracy of 10 runs. The results in tables provide a more sufficient evidence for the effectiveness of our two methods.

4.2   Effect on Standard Semi-supervised Learning

Apart from the experiments under different training sizes, we also test the performance of GCNSVAT and GCNDVAT algorithms in standard semi-supervised

Table 3. Classification accuracies on CiteSeer with different label rates. Numbers in bracket are the standard deviation of accuracies.

	Rates	0.5%	1%	2%	3%	4%	5%
CiteSeer	GCN	36.1 (0.09)	45.7 (0.04)	64.3 (0.04)	68.1 (0.02)	69.1 (0.01)	70.3 (0.01)
	GCN SVAT	47.0 (0.08)	52.4 (0.02)	65.8 (0.02)	68.6 (0.01)	69.5 (0.01)	70.7 (0.01)
	GCN DVAT	**51.5 (0.07)**	**58.5 (0.03)**	**67.4 (0.01)**	**69.2 (0.01)**	**70.8 (0.01)**	**71.3 (0.01)**

Table 4. Classification accuracies on PubMed with different label rates. Numbers in bracket are the standard deviation of accuracies.

	Rates	0.03%	0.05%	0.1%	0.2%	0.3%
PubMed	GCN	46.3 (0.08)	56.1 (0.10)	63.3 (0.06)	70.4 (0.04)	77.1 (0.02)
	GCN SVAT	52.1 (0.06)	56.9 (0.08)	63.5 (0.07)	71.2 (0.04)	77.2 (0.02)
	GCN DVAT	**53.3 (0.06)**	**58.6 (0.06)**	**66.3 (0.05)**	**72.2 (0.03)**	**77.3 (0.02)**

setting with standard label rates listed in Table 1. Particularly, we compare our methods with other state-of-the-art methods on the node classification task under standard label rate and the results of baselines are referred from [8].

From Table 5, it turns out that our GCNDVAT algorithm exhibits the state-of-the-art performance though the improvement are not apparent compared with that in few training sizes, while our GCNSVAT algorithm also shares a similar performance. Through the extensive experiments in semi-supervised learning, we demonstrate thoroughly that VAT suffices to improve the generalization performance of GCNs by additionally providing an adversarial regularization both in semi-supervised setting with few labeled nodes and standard semi-supervised setting.

Table 5. Accuracy under 20 labels per class across three datasets.

Method	CiteSeer	Cora	PubMed
ManiReg	60.1	59.5	70.7
SemiEmb	59.6	59.0	71.7
LP	45.3	68.0	63.0
DeepWalk	43.2	67.2	65.3
Planetoid	64.7	75.7	77.2
GCN	68.4	78.4	77.3
GCNSVAT	68.7	78.5	77.5
GCNDVAT	**69.3**	**78.6**	**77.6**

4.3   Sensitivity Analysis of Hyper-parameters

One of the notable advantage of VAT in GCNs is that there are just two scalar-valued hyper-parameters: (1) the perturbation magnitude ϵ that constraints the norm of adversarial perturbation and (2) the regularization coefficient α that controls the balance between supervised loss $\mathcal{L}_0$ and virtual adversarial loss $\mathbb{E}_{x\sim\mathcal{D}}\mathcal{R}_{\text{vadv}}$. We refine the analysis in original VAT [11] and theoretically demonstrate the total loss is more sensitive to ϵ rather than α in the regularization control of GCNs with VAT setting.

Consider the second approximation of virtual adversarial regularization:

$$
\begin{aligned}
\mathcal{R}_{\text{vadv}}(x, \mathcal{D}_l, \mathcal{D}_{ul}, \theta) &= \max_{r}\{D(r, x, \overline{x}, A; \theta); \|r\|_2 \leq \epsilon\} \\
&\approx \frac{1}{2}\epsilon^2 \lambda_1(x, \overline{x}, A; \theta),
\end{aligned}
\tag{15}
$$

where $\lambda_1(x, \overline{x}, A; \theta)$ is the dominant eigenvalue of Hessian matrix $H(x, \overline{x}, A; \theta)$ of D. Substituting this into the objective function, we obtain

$$
\begin{aligned}
\mathcal{L}_0 + \alpha\mathbb{E}_{x\sim\mathcal{D}}\mathcal{R}_{\text{vadv}} &= \mathcal{L}_0 + \alpha\frac{1}{N_l + N_{ul}}\sum_{x_*\in\mathcal{D}_l,\mathcal{D}_{ul}}\mathcal{R}_{\text{vadv}}(x, \mathcal{D}_l, \mathcal{D}_{ul}, \theta) \\
&\approx \mathcal{L}_0 + \frac{1}{2}\alpha\epsilon^2\frac{1}{N_l + N_{ul}}\sum_{x_*\in\mathcal{D}_l,\mathcal{D}_{ul}}\lambda_1(x_*, \overline{x}, A; \theta).
\end{aligned}
\tag{16}
$$

Thus, the strength of regularization is approximately proportional to α and ϵ^2. In consideration of the regularization term is more sensitive to the change of ϵ, in our experiments we just tune the perturbation ϵ to control the regularization by fixing $\alpha = 1$ for both methods.

Further, we present the tendency between the selected optimal ϵ and label rates. As for the different label rates, it is natural to expect that GCNs with VAT under lower label rate requires larger VAT regularization, yielding the urge for larger optimal ϵ. We empirically verify this conclusion in Fig. 2.

From Fig. 2, it is easy to observe that with the increasing of label rates, there is a descending trend of optimal ϵ for both GCNSVAT and GCNDVAT across three datasets. It meets our expectation since large VAT regularization are more expected for GCNs under lower label rates to obtain the optimal generalization of GCNs. In addition, the optimal ϵ parameter in GCNSVAT under the same label rate tends to be higher than that in GCNDVAT, especially when the label rate is lower. The reason is obvious because GCNSVAT only applies perturbations on specific elements of sparse feature for each node, thus requiring larger perturbations on those features to get similar regularization effect compared with GCNDVAT.
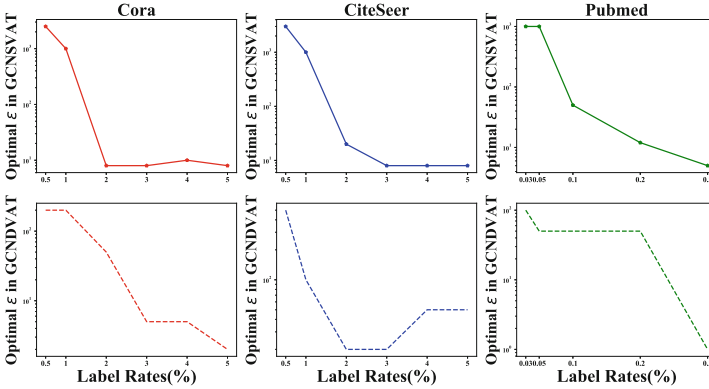
Fig. 2. Sensitivity analysis of epsilon ϵ on two methods.

5   Discussions and Conclusion

GCNs with Virtual Adversarial Training is established on the adversarial training on GCNs, which in our paper is simply constrained in the adversarial perturbations on the features of nodes. However, there may exists a better form of adversarial training in GCNs by additionally considering the change of sensitive edges with respects to the output performance. Therefore, incorporating a better form of Virtual Adversarial Training into graphs allows better improvement of generalization of GCNs. Besides, how to combine VAT with other form Graph Neural Networks especially in inductive setting, is also worthwhile to explore in the future.

In our paper, we impose VAT regularization on the original supervised loss of GCN to enhance its generalization in semi-supervised learning, resulting in GCNSVAT and GCNDVAT, whose perturbations are added in sparse and dense features, respectively. Particularly, we apply VAT on GCNs in a theoretical way by additionally imposing virtual adversarial loss on the basic supervised loss of GCNs. Then we empirically demonstrate the improvement caused by the VAT regularization under different training sizes across three datasets. Our endeavour validates that smoothing anisotropic direction on the posterior distribution of GCNs suffices to improve the performance of original GCN model.

References

1. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
2. Chen, J., Zhu, J.: Stochastic training of graph convolutional networks. arXiv preprint arXiv:1710.10568 (2017)
3. Dai, H., Kozareva, Z., Dai, B., Smola, A., Song, L.: Learning steady-states of iterative algorithms over graphs. In: International Conference on Machine Learning, pp. 1114–1122 (2018)

4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852 (2016)
5. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp. 1024–1034 (2017)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
9. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. arXiv preprint arXiv:1801.07606 (2018)
10. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725 (2016)
11. Miyato, T., Maeda, S.I., Ishii, S., Koyama, M.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1979–1993 (2018)
12. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
13. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI Mag. **29**(3), 93 (2008)
14. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
15. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 1(2) (2017)
16. Yu, B., Wu, J., Zhu, Z.: Tangent-normal adversarial regularization for semi-supervised learning. arXiv preprint arXiv:1808.06088 (2018)
17. Zhu, J., Song, J., Chen, B.: Max-margin nonparametric latent feature models for link prediction. arXiv preprint arXiv:1602.07428 (2016)
18. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2847–2856. ACM (2018)
19. Zügner, D., Günnemann, S.: Adversarial attacks on graph neural networks via meta learning (2018)