IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 26, 2022, accepted May 16, 2022, date of publication May 20, 2022, date of current version June 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176634

Schatten Graph Neural Networks

YOUFA LIU[1], YONGYONG CHEN[2], (Member, IEEE), GUO CHEN[3], AND JIAWEI ZHANG[1]
[1]College of Informatics, Huazhong Agricultural University, Wuhan 430070, China
[2]School of Computer Science, Harbin Institute of Technology, Shenzhen 518055, China
[3]School of Business, Hubei University, Wuhan 430062, China

Corresponding authors: Yongyong Chen (yongyongchen.cn@gmail.com) and Guo Chen (guochen@hubu.edu.cn)

This work was supported in part by the Natural Science Foundation of Hubei Province under Grant 2021CFB139, in part by the Fundamental Research Funds for the Central Universities under Project 2662020XXQD002, and in part by the Shenzhen College Stability Support Plan under Grant GXWD20201230155427003-20200824113231001.

ABSTRACT Graph Neural Networks (GNNs) have been intensively studied in recent years because of their promising performance over graph-structural data and have provided assistance in many fields. Recalling recent works on graph neural networks, we found that imposing graph smoothing via Frobenius norm was proven to be effective in the architecture of graph neural networks from the standpoint of the graph signal processing. In this paper, we aim to model the graph smoothness of graph neural networks using a Schatten p-norm with p in the interval $[1, 2)$ to characterize smoothness and propose a novel architecture called Schatten graph neural networks. This architecture stems from a primal-dual solution scheme for a graph signal denoising problem. There is difficulty in solving subproblems with respect to the Schatten p-norm. We propose a fixed point iteration scheme and prove that it tracks with the linear convergence rate with solid mathematical analysis. Extensive experiments demonstrate the effectiveness of the proposed architecture of graph neural networks and their robustness to the graph adversarial attacks.

INDEX TERMS Low-rank constraint, Schatten p-norm, graph signal processing, primal-dual optimization, graph adversarial attack.

I. INTRODUCTION

Although deep neural networks have seen great development in recent years, they have no ability to treat irregular data, such as that in social networks [1], protein-protein networks [2] and traffic networks [3]. Graph neural networks (GNNs) [4] have been one of the most popular tools to deal with this kind of data and can learn powerful representation from graph-structural data. GNNs can be used in many tasks including node classification [5], link prediction [6], graph classification [7] and recommendation systems [8] and many others [43]–[46].

Based on the mode of local computation, GNNs can be roughly divided into two classes: graph convolution networks [5] and message passing networks [9]. The graph convolution stems from the convolution in deep neural networks which operate on regular graphs, such as image, text and videos. Graph convolution can deal with an irregular graph and capture local information to generate better representations. Mathematically, the k-th graph convolutional layer is

$$X^{(k)} = \varphi_k(\widetilde{L}X^{(k-1)}W^{(k)}),$$

where $X^{(k-1)}$ is the $(k-1)$-th layer representation, $W^{(k)}$ is the feature transformation matrix and φ_k is the activation function. Graph convolutional network (GCN) [5] and graph attention network (GAT) [10] are two classical graph convolution networks. GCN adopts the spectral convolution methods and simple approximation of a Chebyshev polynomial. GAT extends the attention mechanism from deep neural networks to graph neural networks. Message-passing-based GNNs follow from the old message passing algorithms and represents the shared functions by means of graph neural networks. Mathematically, popular massage-passing networks [9] can be unified by

$$\begin{cases} Y^{(k)} = g(C_{out}^{T}X^{(k-1)}), \\ \overline{Y}^{(k)} = C_{in}Y^{(k)}, \\ X^{(k)} = \phi(XW_1 + \theta(\overline{Y}^{(k)})W_2), \end{cases}$$

where the first equation is k-step message computation, the second equation is the k-step message aggregation and the third equation is the k-step node state update.

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Mei Zhang.

We examine message-passing networks in this paper. Recently, it was shown in [11] that many such networks have an affinity with graph signal denoising problems with l_2 graph smoothness. This includes a second-order approximation term and Laplacian regularization term. As a matter of fact, these networks can be deduced from the graph signal denoising problem by using different formats of optimization schemes, for example, a gradient descent algorithm with different step sizes. ElasticGNN [12] attempts to improve the smoothness by both l_1- and l_2-based graph smoothing. Further, it considers the l_{21}- and l_2-based graph smoothing schemes.

In this paper, we propose to establish a graph signal denoising problem with the Schatten p-norm for p in the interval $(0, 2)$. It is well-known that the rank function is the p-order power of the Schatten p-norm at $p = 0$. When p approaches zero, the low-rank property emerges. In practice, the nuclear norm (i.e. $p = 1$) is often used as a convex surrogate of rank function for the convenience of optimization. When p lies in the interval $[1, 2)$, the p-order power of the Schatten p-norm is finite and convex. In this case, the sparsity occurs when we take an appropriate regularization coefficient. Note that the Schatten p-norm with p in the interval $[1, 2)$ dominates the l_2 norm. As a result, the proposed graph signal denoising model incudes the smoothness between l_2 and the Schatten p-norm with $p \in [1, 2)$. When p is greater than 2, the graph signal denoising model just characterizes the l_2 smoothness because the l_2 norm is the upper bound of the Schatten p-norm for $p \geq 2$. This is why we restrict the value of p in the interval $[1, 2)$.

We also discuss the optimization of the graph signal denoising problem with a Schatten p-norm for p in the interval $[1, 2)$. Note that when $p \geq 1$, the p-order power of the Schatten p-norm is convex. With this good property, the objective function in the graph signal denoising problem is the composition of two convex functions. We choose the modified Proximal Alternating Predictor-Corrector (PAPC) optimization scheme [13] to find a solution, which actually belongs to a primal-dual solution scheme. With a special choice of step sizes in PAPC, we obtain a novel architecture of graph neural networks.

In the PAPC schemes, there is difficulty in solving the subproblem of the proximal operator of the Fenchel conjuagte with respect to a convex function (i.e., the scaling p-order power of the Schatten p-norm). By Moreau decomposition, we just need to solve the proximal operator of the scaling p-order power of the Schatten p-norm. We propose an efficient fixed-point iteration scheme. Theoretical analysis shows that this scheme has a linear convergence rate $O(\rho^k)$ with some $\rho \in (0, 1)$.

The robustness under a graph adversarial attack is also examined in our experiments. By setting different attack ratios, the performance is recorded and reported. This reveals the effective robustness of the proposed graph neural networks.

As summarization, the contribution of this paper is as follows.

(1) We propose a novel architecture of graph neural networks called Schatten graph neural networks from the standpoint of graph signal processing, in which, Schatten p-norm is employed to characterize the smoothness. When $p \in (1, 2)$, it gives rise to the mixture of low-rank and l_2 smooth property.

(2) The convergence of the proposed message passing schemes is theoretically proved. In particular, this scheme contains a subproblem of solving proximal operator with respect to the Schatten p-norm. We develop a fixed point iteration algorithm and prove that it bears with linear convergence rate.

The remainder of this paper is organized as follows. In Section II, related works are briefly reviewed. In Section III, we give the problem formulation and notations that are used in the latter sections. In Section IV, we set up the methodology. In Section V, we propose the graph neural network architecture. In Section VI, convergence analysis is performed. Complexity analysis is provided in Section VII. Extensive experiments are conducted in Section VIII. Finally, we conclude this paper in Section IX.

II. RELATED WORKS

A. GRAPH SIGNAL PROCESSING

The popular architectures of graph neural networks including GCN [5] and GAT [10] can be implicitly obtained by using a gradient descent algorithm to solve the following graph signal denoising problem with a particular step:

$$\min_{X \in \mathbb{R}^{n\times d}} \frac{1}{2}\|X - X_{input}\|_F^2 + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\|x[i:] - x[j:]\|_2^2, \tag{1}$$

where X_{input} is the input signal, $A \in \mathbb{R}^{n\times n}$ is the symmetric adjacency matrix whose entries are A_{ij}, $x[i, :]$ is the i-th row instances and λ is a positive trade-off parameter. The last term indicates the global l_2 smoothness. Recently, Elastic GNNs [12] were proposed to improve the smoothness. GNNs include l_1- and l_{21}-level smoothness, which induces a better sparsity of graph signals. The l_1 smoothness is characterized by

$$\min_{X\in\mathbb{R}^{n\times d}} \frac{1}{2}\|X - X_{input}\|_F^2 + \lambda_1\|\Delta X\|_1 + \frac{\lambda_2}{2} tr(X^T\widetilde{L}X), \tag{2}$$

where $\Delta \in \{-1, 0, 1\}^{m\times n}$ is the oriented incident matrix, where m is equal to $|E|$ and each row is like

$$(0, 0, -1, 0, \cdots, 0, 1, 0, 0), \tag{3}$$

where the nonzero terms denote two nodes with directed edges. The l_{21} can induce the row sparsity of X. This provides more precise sparsity than the Frobenius norm. The l_{21} smoothness is characterized by

$$\min_{X\in\mathbb{R}^{n\times d}} \frac{1}{2}\|X - X_{input}\|_F^2 + \lambda_1\|\Delta X\|_{21} + \frac{\lambda_2}{2} tr(X^T\widetilde{L}X), \tag{4}$$

B. GRAPH ADVERSARIAL ATTACK

An adversarial attack on graph structured data [14] is an active field of graph learning. For a given graph structured dataset $\mathcal{D} = (c_j, G_j, y_j)$, if we change G_j as $\widetilde{G}_j$ a little such that the adversarial samples $\widetilde{G}_j$ and G_j become similar under some specified metrics, the performance of the graph task is worse than before. This is the general adversarial attack problem over graph data. There exist many works on graph adversarial attacks including [14] and [15]. To effectively generate adversarial samples, nodes or edges can be slightly changed and the similarity can be achieved by some perturbation evaluation metrics. As in [14], the imperceptible perturbation can be roughly categorized into four classes: node-level perturbation, edge-level perturbation, structure preserving perturbation and attribute preserving perturbation.

C. LOW-RANK MATRIX MINIMIZATION

The low-rank approximation model is formulated as

$$\min_{X \in \mathbb{R}^{n\times d}} L(X, Y) + \lambda rank(X), \tag{5}$$

However, this optimization problem is difficult to solve. Then nuclear norm is considered an approximate scheme because the nuclear norm is is the convex envelope of rank on the unit ball of matrix operator norm [16]. Hence, one can get a relaxed version as

$$\min_{X \in \mathbb{R}^{n\times d}} L(X, Y) + \lambda \|X\|_*, \tag{6}$$

where the nuclear norm is

$$\|X\|_* = \sum_{i=1}^{\min\{n,d\}} \sigma_i(X). \tag{7}$$

The low-rank matrix approximation of a given matrix $Y \in \mathbb{R}^{n\times d}$ with the Schatten p-norm is described as

$$\min_{X \in \mathbb{R}^{n\times d}} L(X, Y) + \lambda \|X\|_{\mathcal{S}_p}^p, \tag{8}$$

where L is the loss function, $p > 0$ and

$$\|X\|_{\mathcal{S}_p} = \left(\sum_{i=1}^{\min\{n,d\}} \sigma_i^p(X) \right)^{\frac{1}{p}}. \tag{9}$$

It includes nuclear norm ($p = 1$) and Frobenius norm ($p = 2$) as two popular examples.

Apart from the power form of eigenvalues, a more general nonconvex and nonsmoothness low-rank minimization is summarized in [17].

$$\min_{X \in \mathbb{R}^{n\times d}} L(X, Y) + \sum_{i=1}^{n} g_\lambda(\sigma_i(X)), \tag{10}$$

where $\lambda \geq 0$ is a nonnegative controlling parameter. The usual examples of penalty include L_p [18], SCAD [19] and Laplace [20]. L_p penalty is

$$g_\lambda(\theta) = \lambda\theta^p. \tag{11}$$

The SCAD penalty is described as

$$g_\lambda(\theta) = \begin{cases} \lambda\theta, & \text{if } \theta \leq \lambda; \\ \dfrac{-\theta^2 + 2\gamma\lambda\theta - \lambda^2}{2(\gamma - 1)}, & \text{if } \lambda < \theta \leq \gamma\lambda; \\ \dfrac{\lambda^2(\gamma + 1)}{2}, & \text{if } \theta > \gamma\lambda. \end{cases} \tag{12}$$

The Laplace penalty is

$$g_\lambda(\theta) = \lambda \left(1 - e^{-\frac{\theta}{\gamma}} \right). \tag{13}$$

D. OPTIMIZATION

Smooth optimization [22] and non-smooth optimization [21] have been widely studied in machine learning fields. Some optimization problems can be decomposed as the sum of convex smooth and nonsmooth components. Mathematically,

$$\min_{X \in \mathbb{R}^{n\times d}} f(X) + g(X), \tag{14}$$

where f and g are convex functions but f is smooth function.

For the convex function g, its Fenchel conjugate is

$$g^*(X) = \sup_{Z \in \mathbb{R}^{n\times d}} \langle X, Z\rangle - g(Z). \tag{15}$$

Then, we can obtain the equivalent saddle point problem

$$\min_{X \in \mathbb{R}^{n\times d}} \max_{Z \in \mathbb{R}^{n\times d}} f(X) + \langle X, Z\rangle - g^*(Z), \tag{16}$$

Candidate algorithms such as Alternating Direction Method of Multipliers (ADMM) [23] and Newton type [24] may work. These candidate algorithms may contain the task of finding the solution to some nontrivial sub-problem with a heavy computation burden. Intermediate optimization problem-solving may be incompatible with the standard back-propagation (BP) algorithms in general deep learning. The Proximal Alternating Predictor-Corrector (PAPC) [13] is a kind of primal-dual optimization algorithm that has been proven to be effective in the recent work ElasticGNN [12].

III. PROBLEM FORMULATION AND NOTATIONS

Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ be a graph. $\mathcal{V}$ is the vertex set, $\mathcal{F}$ is the collection of features of nodes and $\mathcal{E}$ is the edge set. $\mathcal{E} = \{e_1, \cdots, e_m\}$ can be represented by a matrix $A \in \mathbb{R}^{n\times n}$ called adjacency matrix for the graph G, where n is the number of vertices. If node v_i and v_j in the vertex set V are connected, then set $A_{ij} = 1$; otherwise, set $A_{ij} = 0$. $\mathcal{E}$ can also be considered in another way: the edge set is characterized by the incident matrix

$$\Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \cdots \\ \Delta_m \end{bmatrix},$$

where Δ_i is like

$$(0, 0, -1, 0, \cdots, 0, 1, 0, 0) \in \mathbb{R}^n,$$

in which -1 indicates the initial point of edge e_i, and 1 denotes its terminal point.

Let $\widetilde{A} = I + A$, where I is the identity matrix of order n. Actually, $\widetilde{A}$ is the self-looped version because every node in V is self-connected. Let $\widetilde{D}$ be the degree matrix w.r.t $\widetilde{A}$ and $\widetilde{L} = \widetilde{D} - \widetilde{A}$ is the normalized Laplacian matrix. Given any matrix B, $\sigma_i(B)$ denotes the its i-th largest singular value.

For each node $v \in V$, a feature $x_v \in \mathbb{R}^m(x_v \in \mathcal{F})$ is assigned, in which m is the feature dimension. We use l labeled nodes $\{(x_1, y_1), \cdots, (x_l, y_l)\}$ and $n - l$ unlabeled nodes $\{x_{l+1}, \cdots, x_n\}$. The task of node classification is to establish a GNN model $f : (X, Y_l, A) \rightarrow \mathcal{Y}$, where $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times m}$, $Y_l = \{y_1, \cdots, y_l\}$, and $\mathcal{Y}$ denotes the label space. With this learned model, the predicted label of each unlabeled node is produced. We also consider the robustness of the proposed approach under graph adversarial attacks.

IV. METHODOLOGY

In this paper, we propose modelling the smoothness by the Schatten p-norm which is different from the ElasticGNN [12]. For any $p > 0$ and $Z \in \mathbb{R}^{n\times d}$,

$$\|Z\|_{\mathcal{S}_p} = Tr((Z^TZ)^{\frac{p}{2}})^{\frac{1}{p}} \tag{17}$$

$$= \left(\sum_{i=1}^{\min\{n,d\}} \sigma_i^p(Z)\right)^{\frac{1}{p}}. \tag{18}$$

Mathematically, we consider the following graph signal denoising model.

$$\min_{X\in\mathbb{R}^{n\times d}} \underbrace{\frac{1}{2}\|X - X_{input}\|_F^2 + \frac{\lambda_1}{2}tr(X^T\widetilde{L}X)}_{f(X)} + \underbrace{\lambda_2\|\widehat{\Delta}X\|_{\mathcal{S}_p}^p}_{u(\widehat{\Delta}X)}, \tag{19}$$

where X_{input} is the initial graph signal, p is a positive number, $\lambda_1 > 0$ and $\lambda_2 > 0$ are two trade-off parameters, and

$$\widehat{\Delta} = \Delta\widetilde{D}^{-\frac{1}{2}}. \tag{20}$$

In this model, λ_1 forces the l_2-smoothness. λ_2 forces the property induced by the Schatten p-norm, i.e. low-rank property, l_2-smooth property or their mixture, which depends on the value of p.

We interpret this model in detail and devise a novel graph neural networks later. The first term in the objective makes X approximate to X_{input}. The $tr(X^T\widetilde{L}X)$ in the second term can be expanded as

$$tr(X^T\widetilde{L}X) = \|\widehat{\Delta}X\|_2^2$$
$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\|x[i:] - x[j:]\|_2^2, \tag{21}$$

where $x[i\ :]$ and $x[j\ :]$ denote the i-th and j-th row vectors of X, respectively. This reveals that the l_2 sparsity is directly implied by the second term in (19).

By choosing proper parameter λ_1, two connected nodes draw closer to each other in the search space, which is similar to the manifold regularization [25]. The last term $\|\widehat{\Delta}X\|_{\mathcal{S}_p}^p$ may have diverse meanings. It depends on the value of p. There exist some particular cases as follows. Recall that when p tends to zero, for any $Z \in \mathbb{R}^{m\times d}$, we have

$$\lim_{p\rightarrow 0^+} \|Z\|_{\mathcal{S}_p}^p = rank(Z). \tag{22}$$

If p is equal to 1, then the Schatten 1-norm is the standard nuclear norm, i.e.

$$\|Z\|_{\mathcal{S}_1} = \|Z\|_*. \tag{23}$$

For $p = 2$, the corresponding Schatten 2-norm is exactly the Frobenius norm, i.e.

$$\|Z\|_{\mathcal{S}_2} = \|Z\|_F. \tag{24}$$

If $p = \infty$, then the Schatten ∞-norm is operator norm, namely,

$$\|Z\|_{\mathcal{S}_\infty} = \max_{x\in\mathbb{R}^d-\{0\}} \frac{\|Zx\|_2}{\|x\|_2}. \tag{25}$$

According to [26], we have the following inequalities, namely,

$$\|Z\|_{\mathcal{S}_\infty} \leq \|Z\|_{\mathcal{S}_2} \leq \|Z\|_{\mathcal{S}_1}. \tag{26}$$

Set $Z = \widehat{\Delta}X$. It is readily seen from the meaning of $\widehat{\Delta}$ that every line of Z is of the form

$$x[j:] - x[i:]. \tag{27}$$

if there exists a directed edge $e_{ij} \in \mathcal{E}$ from node i to node j. For an undirected graph, each edge can be decomposed as two directed edges with opposite orientation. In other words, if nodes v_i and v_j are connected, then two directed edges $E_{i\rightarrow j}$ and $E_{j\rightarrow i}$ emerge.

When p is small, the last regularization term in (19) induces the low-rank property of $\widehat{\Delta}X$. When $p \geq 2$, $\|Z\|_{\mathcal{S}_p} \leq \|Z\|_{\mathcal{S}_2}$, which implies that the second term in (19) dominates the last term. In this case, the model (19) is just l_2-smooth. It is well-known that the Schatten p-norm is convex when $p \geq 1$, which is friendly to the usual optimization strategy. Therefore, we restrict the value of p in the interval $[1, 2]$. When $p \in [1, 2]$, we have $\|Z\|_{\mathcal{S}_2} \leq \|Z\|_{\mathcal{S}_p} \leq \|Z\|_{\mathcal{S}_1}$, which means that the last term in (19) induces a mixed property between the low rank and sparsity of $\widehat{\Delta}X$. In the next section, we propose a novel graph neural network architecture from the optimization of (19).

V. THE PROPOSED ARCHITECTURE OF THE GRAPH NEURAL NETWORKS

A. REFORMULATION AS SADDLE POINT PROBLEM

The procedure is displayed in Figure 1. The method of solving optimization (19) is the key to deducing the Schatten graph neural networks. To solve problem (19), we consider its saddle point formulation. Let

$$\mathcal{L}(X, Z) = G(X) + H(X, Z), \tag{28}$$

where

$$G(X) = \frac{1}{2}\|X - X_{input}\|_F^2 + \frac{\lambda_1}{2}tr(X^T\widetilde{L}X) \tag{29}$$

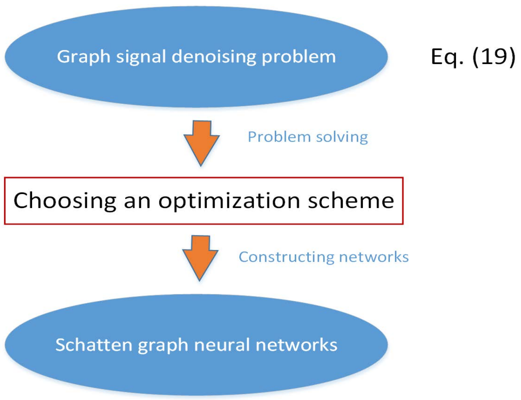
Graph signal denoising problem
Eq. (19)
Problem solving
Choosing an optimization scheme
Constructing networks
Schatten graph neural networks

FIGURE 1. The illustration of the procedure. The graph neural networks can be constructed by the inspiration of the optimization for solving graph signal denoising problems. For example, the GCN [5] can be regarded as a gradient descent scheme with a particular stepsize [12]. Analogously, our proposed Schatten graph neural networks can be derived by a optimization scheme (i.e. PAPC scheme).

and

$$H(X, Z) = \langle \widehat{\Delta} X, Z \rangle - g^*(Z), \tag{30}$$

in which $g^*(Z)$ is the Fenchel conjugate of $g(Z) = \lambda_2 \|Z\|_{\mathcal{S}_p}^p$, i.e.,

$$g^*(Z) = \sup_{Y \in \mathbb{R}^{m \times d}} \langle Z, Y \rangle - g(Y). \tag{31}$$

With these notations, the optimization problem (19) can be written as

$$\min_{X \in \mathbb{R}^{n \times d}} G(X) + g^*(\widehat{\Delta} X). \tag{32}$$

Then (19) can be further reformulated as a saddle point problem

$$\min_{X \in \mathbb{R}^{n \times d}} \max_{Z \in \mathbb{R}^{m \times d}} \mathcal{L}(X, Z). \tag{33}$$

B. OPTIMIZATION SCHEME

Following [12], we use the modified Proximal Alternating Predictor-Corrector (PAPC) [13] scheme as follows:

$$\overline{X}^{(k+1)} = X^{(k)} - \gamma \nabla G(X^{(k)}) - \gamma \widehat{\Delta}^T Z^{(k)}, \tag{34}$$

$$Z^{(k+1)} = \text{prox}_{\beta g^*}(Z^{(k)} + \beta \widehat{\Delta} \overline{X}^{(k+1)}), \tag{35}$$

$$X^{(k+1)} = X^{(k)} - \gamma \nabla G(X^{(k)}) - \gamma \widehat{\Delta}^T Z^{(k+1)}. \tag{36}$$

Equation (35) involves the proximal operator of Fenchel conjugate g^*. This is not easy to compute directly. By employing Moreau's decomposition [27], we have

$$Z = \text{prox}_{\beta g^*}(Z) + \lambda \text{prox}_{\beta^{-1} g}(\frac{1}{\beta} Z). \tag{37}$$

If we can solve the proximal operator of g, the problem with (35) is readily solved.

Recall that $g(Z) = \lambda_2 \|Z\|_{\mathcal{S}_p}^p$. Then the proximal operator with respect to g is

$$\text{prox}_{\beta^{-1} g}(Z) = \arg \min_{Y \in \mathbb{R}^{m \times d}} \frac{1}{2} \|Y - Z\|_F^2 + \frac{\lambda_2}{\beta} \|Z\|_{\mathcal{S}_p}^p. \tag{38}$$

By [39], it is closely related to

$$\min_{\delta \geq 0} h(\delta) = \frac{1}{2}(\delta - \sigma)^2 + \omega \delta^p. \tag{39}$$

In Section VI, we provide an efficient fixed-point iteration algorithm to solve this problem with strong convergence analysis. Once $\text{prox}_{\beta^{-1} g}(Z)$ is solved, by (37),

$$\text{prox}_{\beta g^*}(Z) = Z - \text{prox}_{\beta^{-1} g}(Z). \tag{40}$$

C. NETWORK ARCHITECTURE

Based on the selected optimization scheme, we deduce a novel message-passing-mechanism based graph neural network architecture.

Let

$$W^{(k)} = X^{(k)} - \gamma \nabla G(X^{(k)}), \tag{41}$$

where the function $G(\cdot)$ comes from Eq. (29). The first-order derivative of G is

$$\nabla G(X) = X - X_{input} + \lambda_1 \widetilde{L} X. \tag{42}$$

Inserting (40) into (39), we have

$$W^{(k)} = [(1 - \gamma) I - \lambda_1 \gamma \widetilde{L}] X^{(k)} + \gamma X_{input}. \tag{43}$$

Combining (34-36) and (42), the optimization scheme can be formulated as

$$\begin{cases} W^{(k)} = [(1 - \gamma) I - \lambda_1 \gamma \widetilde{L}] X^{(k)} + \gamma X_{input}, \\ \overline{X}^{(k+1)} = W^{(k)} - \gamma \widehat{\Delta}^T Z^{(k)}, \\ \overline{Z}^{(k)} = Z^{(k)} + \beta \widehat{\Delta} \overline{X}^{(k)}, \\ Z^{(k+1)} = \text{prox}_{\beta g^*}(\overline{Z}^{(k)} + \beta \widehat{\Delta} \overline{X}^{(k+1)}), \\ X^{(k+1)} = W^{(k)} - \gamma \widehat{\Delta}^T Z^{(k+1)}. \end{cases} \tag{44}$$

It is sufficient to take $\gamma = \frac{1}{1+\lambda_1}$ and $\beta = \frac{1}{2\gamma}$ for convergence (This will be proved in Theorem 1.). With this in mind, we have the following scheme:

$$\begin{cases} W^{(k)} = \gamma X_{input} + (1 - \gamma) \widetilde{A} X^{(k)}, \\ \overline{X}^{(k+1)} = W^{(k)} - \gamma \widehat{\Delta}^T Z^{(k)}, \\ \overline{Z}^{(k)} = Z^{(k)} + \beta \widehat{\Delta} \overline{X}^{(k)}, \\ Z^{(k+1)} = \text{prox}_{\beta g^*}(\overline{Z}^{(k)} + \beta \widehat{\Delta} \overline{X}^{(k+1)}), \\ X^{(k+1)} = W^{(k)} - \gamma \widehat{\Delta}^T Z^{(k+1)}. \end{cases} \tag{45}$$

if $X^{(k)}$ and $Z^{(k)}$ are regarded as the node's embedding of the k-th layer and connection parameters, then we can construct a graph neural network by stacking layer by layer. For intuition, we change (45) as the language of graph neural networks and provide the network architecture in Figure 2.

VI. CONVERGENCE ANALYSIS

In this section, we provide the iteration scheme of (45) with convergence guarantee in Theorem 1. Since there is a subproblem of solving proximal operator in (45), which is related to (39). We propose a foxed-point iteration scheme to solve (39) with the analysis of linear convergence rate.

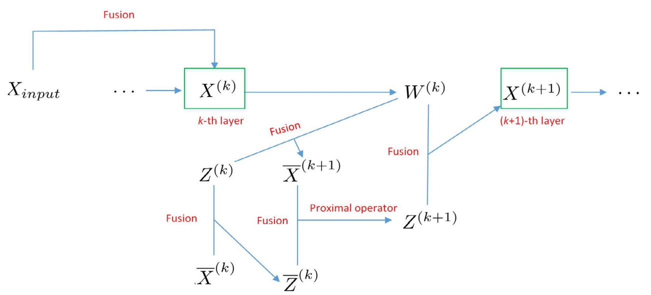

FIGURE 2. **The visualization of the iteration scheme Eq. (45). The detailed process from $X^{(k)}$ to $X^{(k+1)}$ is illustrated. This process includes several fusion operations and computation of a proximal operator. The fusion operation is indeed the linear operation on matrices.**

Lemma 1: Let $u(W) = \|W\|^p_{\mathcal{S}_p}$ be the function from the graph signal denoising model (19). The function $u(W) = \|W\|^p_{\mathcal{S}_p}$ is convex for $p \geq 1$.

Proof: Recall from [40] that the Schatten p-norm $\| \bullet \|_{S_p}$ is convex for $p \geq 1$. Note that $l(t) = t^p$ is convex over $\{t \in \mathbb{R} : t \geq 0\}$ for $p \geq 1$ because the second-order derivative of l satisfies $l''(t) = p(p-1)t^{p-2} \geq 0$. The function u can be decomposed as the composition of l and $\| \bullet \|_{S_p}$, i.e. $u(\bullet) = l \circ \| \bullet \|_{S_p}$. Since the composition of two convex function is still convex, $u(\bullet)$ is convex. □

Theorem 1: Let $\gamma = \frac{1}{1+\lambda_1}$ and $\beta = \frac{1}{2\gamma}$. The message-passing scheme (45) converges to the optimal solution of graph signal model (19).

Proof: By Lemma 1, the function u is convex in the graph signal denoising model (19). The objective function in (19) is the sum of $f(X)$ and $u(\widehat{\Delta} X)$, where $\widehat{\Delta}$ is a bounded linear operator. The gradient ∇f satisfies Lipschitz condition. By [41] and [42], the iteration scheme (45) is convergent under $\gamma < \frac{2}{L}$ and $\beta < \frac{4}{3\gamma\lambda_{\max}(\widehat{\Delta}\widehat{\Delta}^T)}$, where L is the Lipschitz constant. L can be computed as $L = \lambda_{\max}(\nabla^2 f(X)) = 1 + \lambda_1 \|\widetilde{L}\|_2$. For a matrix R, let $\|R\|_2$ denotes its spectral norm. Note that $\|L\|_2 = \|\widehat{\Delta}\widehat{\Delta}^T\|_2 = \|\widehat{\Delta}^T\widehat{\Delta}\|_2 \leq 2$. Hence the given values of λ and β satisfy

$$\gamma = \frac{1}{1+\lambda_1} < \frac{1}{1+2\lambda_1} \leq \frac{1}{1+\lambda_1\|\widetilde{L}\|_2} = \frac{2}{L}$$

and

$$\beta = \frac{1}{2\gamma} < \frac{2}{3\gamma} \leq \frac{4}{3\|\widehat{\Delta}\widehat{\Delta}^T\|_2} \leq \frac{4}{3\gamma\lambda_{\max}(\widehat{\Delta}\widehat{\Delta}^T)}.$$

respectively. □

Note that there exits a subproblem of solving proximal operator. The key is to solve (39). We find that (39) can be efficiently solved by fixed point iteration. As a matter of fact, for a sufficiently small positive number ε, the (39) has an identical solution to

$$\min_{\delta > \varepsilon} h(\delta) = \frac{1}{2}(\delta - \sigma)^2 + \omega\delta^p. \tag{46}$$

In practice, we find that $\varepsilon = 0.1$ works well.

We propose the fixed-point iteration as follows:

$$\delta^{(k+1)} = J(\delta^{(k)}), \tag{47}$$

where

$$J(\delta) = \sigma - \omega p \delta^{p-1}. \tag{48}$$

The following theorem provides the global convergence analysis with a convergence rate.

Theorem 2: Let $p \in (1, 2)$. Assume positive ω and ε satisfy

$$\rho = \frac{\omega p(p-1)}{\varepsilon^{2-p}} \in (0, 1). \tag{49}$$

The fixed point iteration scheme (47) is convergent to a unqiue point $\widetilde{\delta}^*$ with rate $\mathcal{O}(\rho^k)$, i.e.

$$|\delta^{(k)} - \widetilde{\delta}^*| \leq \rho^k |\delta^{(0)} - \widetilde{\delta}^*|. \tag{50}$$

Furthermore, the solution to (46) is

$$\delta^* = \max\{\widetilde{\delta}^*, \varepsilon\}. \tag{51}$$

Proof: First of all, we prove the contraction property of $J(\delta)$ on $\mathbb{R}$, i.e., for δ and δ' in the interval $\mathbb{R} - [-\varepsilon, \varepsilon]$,

$$|J(\delta) - J(\delta')| \leq c|\delta - \delta'|, \tag{52}$$

where $c \in (0, 1)$.

By Mean Value Theorem, there exists ξ between δ and δ' such that

$$|J(\delta) - J(\delta')| \leq |J'(\xi)||\delta - \delta'|. \tag{53}$$

Note that

$$J'(\xi) = -\omega p(p-1)\delta^{p-2}. \tag{54}$$

Since $|\xi| > \varepsilon$,

$$|J'(\xi)| < \rho = \frac{\omega p(p-1)}{\varepsilon^{2-p}} \in (0, 1). \tag{55}$$

Combing (53) with (55), we have

$$|J(\delta) - J(\delta')| \leq \rho|\delta - \delta'|. \tag{56}$$

According to the Hahn-Banach Theorem [28], we have that the function J has a unique fixed point $\widetilde{\delta}^*$ in the interval $\mathbb{R} - [-\varepsilon, \varepsilon]$.

$$\begin{aligned} |\delta^{(k)} - \widetilde{\delta}^*| &= |J(x^{(k-1)}) - \widetilde{\delta}^*| \\ &= |J(x^{(k-1)}) - J(\widetilde{\delta}^*)| \\ &\leq \rho|\delta^{(k-1)} - \widetilde{\delta}^*| \\ &\leq \cdots \\ &\leq \rho^k|\delta^{(0)} - \widetilde{\delta}^*|, \end{aligned} \tag{57}$$

where the first equality holds by the (47) and the second equality is true by the meaning of fixed point.

Now we need show that

$$\delta^* = \max\{\widetilde{\delta}^*, \varepsilon\}. \tag{58}$$

is indeed the solution to (46).

The second-order derivative of h is

$$h''(\delta) = 1 + \omega p(p-1)\delta^{p-2}. \tag{59}$$

Note that

$$|\omega p(p-1)\delta^{p-2}| \leq \rho. \tag{60}$$

TABLE 1. Statistics on benchmark datasets.

Attributes	Datasets						
	Cora	CiteSeer	PubMed	CS	Physics	Computers	Photo
Classes	7	6	3	15	5	10	8
Nodes	2078	3327	19717	18333	34493	13381	7487
Edges	5278	4552	44324	81894	247962	245778	119043
Features	1433	3703	500	6085	8415	767	745
Training nodes	20/class	20/class	20/class	20/class	20/class	20/class	20/class
Validation nodes	500	500	500	30/class	30/class	30/class	30/class
Test nodes	1000	1000	1000	Rest nodes	Rest nodes	Rest nodes	Rest nodes

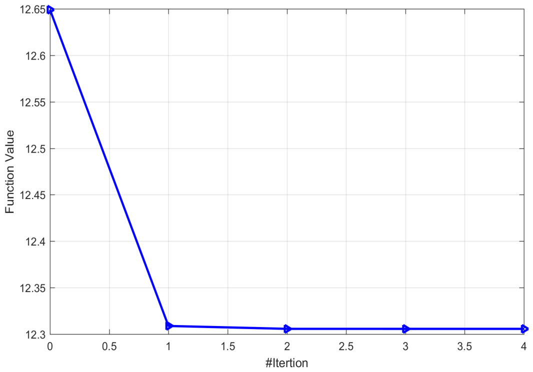

FIGURE 3. The illustration of the change of function value withe respect to the iteration number.

Then

$$h''(\delta) \in (1-\rho, 1+\rho). \tag{61}$$

Since $\rho \in (0,1)$, $h''(\delta) > 0$ which implies that h is strictly convex on $\mathbb{R} - [-\varepsilon, \varepsilon]$. If $\widetilde{\delta}^*$ falls into $(-\infty, -\varepsilon)$, then the minimizer of (46) is just ε. Otherwise, the minimizer is $\widetilde{\delta}^*$ itself. □

In this theorem, $p = 1$ is excluded. In fact, when $p = 1$, the problem (39) has a closed-form solution [17].

Example: We give an example as follows. Take $\sigma = 10$, $\omega = 0.4$, $\varepsilon = 0.1$ and $p = 1.5$.

$$\min_{\delta > \varepsilon = 0.1} h(\delta) = \frac{1}{2}(\delta - 10)^2 + 0.4 * \delta^{1.5}. \tag{62}$$

The initial value is set as 10. The minimizer is $\delta = 9.8084$. We display the iteration process in Figure 3. The iteration converges with very few iteration. This examples empirically show the efficiency of the proposed iteration scheme.

VII. COMPUTATIONAL COMPLEXITY

In this section, we provide the analysis of computational complexity with respect to the iteration Eq. (45). The computational cost mainly comes from the matrix multiplication and the proximal operating which needs SVD and fixed point iterations. Let d be the hidden dimension. For the k-iteration of Eq. (45), the first equation needs the cost of $\mathcal{O}(n^2 d)$. The second equation needs the cost $\mathcal{O}(nmd)$. The third equation needs the cost of $\mathcal{O}(mnd)$. The fourth equation needs the cost of $\mathcal{O}(md^2 + \min\{m, d\}T_{\max})$, where $T_{\max}$ is the maximum iteration number of fixed point iteration scheme Eq. (47). The fifth equation needs the cost of $\mathcal{O}(mnd)$. Assume the maximum iteration number of Eq. (45) is $K_{\max}$, then the overall computational cost is $\mathcal{O}(((n^2+mn)d + md^2 + \min\{m, d\}T_{\max})K_{\max})$.

VIII. EXPERIMENTS

In this section, extensive experiments are performed to verify the effectiveness of the proposed Schatten graph neural networks. We state the used datasets and baselines. The parameter setting strategy is also formulated in detail. The robustness of the proposed Schatten graph neural networks is considered. We also provide an ablation study to measure the impact of parameters.

A. DATASETS AND BASELINES

We conduct experiments on eight graph-structural datasets that contain three citation graphs (Cora, Citeseer, Pubmed [29]), two co-authorship graphs (Coauthor CS and Coauthor Physics [30]), two co-purchase graphs, (Amazon Computers and Amazon Photo [30]), and one blog graph (Polblogs [31]). In the Polblogs graph, node features are not available and we specify the feature map as an identity matrx. For all of these datasets, the statistics such as classes, edges and features are displayed in Table 1.

The baselines are selected as some recent approaches, such as GCN [5], GAT [10], SGC [32], GraphSAGE [33], APPN [34], ElasticGNN [12] and EigenGCN [35]. For fair comparison, a two-layer network architecture with 64 hidden dimensional representations is adopted in all models. This setting strategy follows the Elastic GNN [12]. We choose the classification accuracy as the comparison criterion of the performance.

B. PARAMETER SETTING AND SUMMARY

The average performance together with the standard variance within 10 runs is reported in Table 2. The learning rate is selected from $\{0.05, 0.01, 0.005\}$. The weight decay is tuned over the set $\{5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}\}$. The Adam optimizer is used in our experiments. The choice of optimizer is based on our experience. We found that Adam optimizter typically returns a better local minimizer than SGD optimizer in experiments. The dropout rate lies in the set $\{0.5, 0.8\}$. The hidden dimension of node embedding is fixed as 64. The number of layers of the proposed Schatten GNN is

TABLE 2. Node classification accuracies (%) on benchmark datasets. '–' denotes that there is no available result in the original paper.

Methods	Datasets						
	Cora	CiteSeer	PubMed	CS	Physics	Computers	Photo
GCN	79.6±1.1	68.9±1.2	77.6±2.3	91.6±0.6	93.3±0.8	79.8±1.6	90.3±1.2
GAT	80.1±1.2	68.9±1.8	77.6±2.2	91.1±0.5	93.3±0.7	79.3±2.4	89.6±1.6
SGC	80.2±1.5	68.9±1.3	75.5±2.9	90.1±1.3	93.1±0.6	73.0±2.0	83.5±2.9
APPN	82.2±1.3	70.4±1.2	78.9±2.2	92.5±0.3	93.7±0.7	80.1±2.1	90.8±1.3
EigenGCN	81.4±1.5	70.1±1.8	78.9±2.1	-	-	-	-
EasticGNN	82.7±1.0	70.9±1.4	79.4±1.8	92.5±0.3	94.2±0.5	80.7±1.8	91.3±1.3
Schatten GNN	**82.9±1.5**	**71.0±1.4**	**80.0±2.0**	**92.8±0.4**	**94.5±0.6**	**81.0±1.8**	**91.7±1.8**

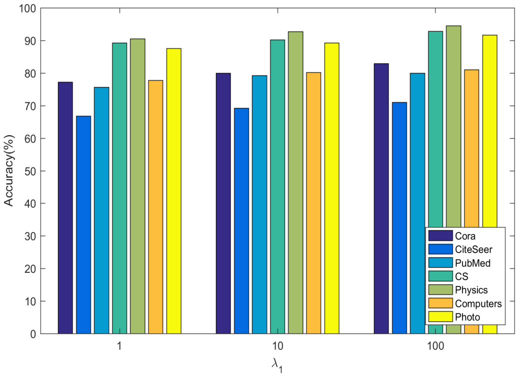

FIGURE 4. The sensitivity of λ_1.

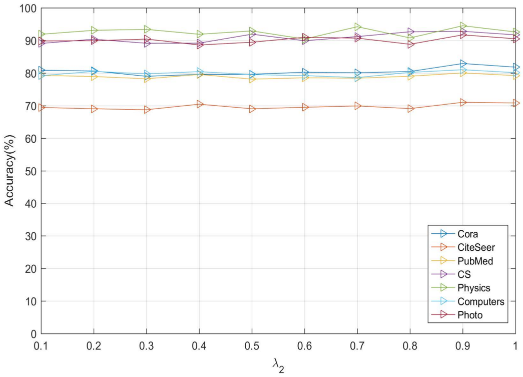

FIGURE 5. The sensitivity of λ_2.

TABLE 3. Statistics on graph under adversarial attack.

Name	Attributes			
	$\mathrm{N_{LCC}}$	$\mathrm{E_{LCC}}$	Classes	Features
Cora	2485	5069	7	1433
CiteSeer	2110	3668	6	3703
Polblogs	1222	16714	2	/
PubMed	19717	4438	3	500

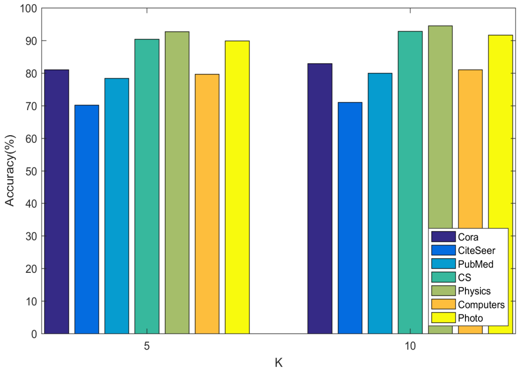

FIGURE 6. The sensitivity of K.

tuned from $\{5, 10\}$. λ_1 is tuned over $\{1, 10, 100\}$. λ_2 is tuned from selected from $\{0.1, 0.2, \ldots, 1\}$. Set $\gamma = \frac{1}{1+\lambda_1}$ and $\beta = \frac{1}{2\gamma}$. The parameters can also be determined by 10-fold cross validation.

The proposed Schatten GNN is derived from graph signal processing problem Eq. (19) with a specific optimization scheme Eq. (45). In fact, all the chosen comparison methods in Table 2 can be derive from a kind of graph signal processing problems with different regularizers and optimization schemes. The difference between the proposed Schatten GNN and recent Elastic GNN [12] is the last regularization in Eq. (29). The experimental result reveals that the mixed property of low rank and l_2 smoothness will makes the performance of graph neural networks stronger than just l_2 smoothness. All experiments are conducted on 1 Tesla V100 GPU. The average running time is within few minutes per task.

C. SENSITIVITY ANALYSIS

We empirically analyze the parameter sensitivity of the graph signal denoising model under the Schatten p-norm. λ_1 lies in $\{1, 10, 100\}$. λ_2 is selected from $\{0.1, 0.2, \ldots, 1\}$. K is tuned from $\{5, 10\}$. p ranges from $\{0.1, 0.5, 1, 1.5\}$. The sensitivity curves are shown in Figure 4-7. The performance becomes better as the λ_1 takes larger values. The cause is that the larger λ_1 imposes stronger constraint of l_2 sparsity. When λ_2 takes small value, the performance is good. when p approaches to 2, the Schatten p-norm is far from low-rank property and close to l_2 smoothness but is looser. In this case, the performance of the proposed approach achieves the best.

D. ABLATION STUDY

We perform an ablation study as follows. Aiming at the graph signal denoising problem with the Schatten p-norm, we set $\lambda_2 = 0$ and observe the performance. In this case, the graph signal denoising reduces to the l_2 norm-based graph smoothing. We show the performance change of the proposed approach along with the number of layers K in Figure 8.

TABLE 4. Classification accuracy (%) under different perturbation rates of adversarial graph attack.

Datasets	PtbRate	GCN	GAT	ElasticGNN (l_2)	ElasticGNN (l_1+l_2)	ElasticGNN $(l_{21}+l_2)$	Schatten GNN
Cora	0%	83.5±0.4	84.0±0.7	85.8±0.4	85.8±0.4	85.8±0.4	**85.9**±**0.4**
	5%	76.6±0.8	80.4±0.7	81.0±1.0	81.9±1.4	82.2±0.9	**82.2**±**0.8**
	10%	70.4±1.3	75.6±0.6	76.3±1.5	78.2±1.6	78.8±1.7	**79.0**±**1.6**
	15%	65.1±0.7	69.8±1.3	72.2±0.9	76.9±0.9	77.2±1.6	**78.0**±**1.8**
	20%	60.0±2.7	59.9±0.6	67.7±0.7	67.2±5.3	70.5±1.3	**70.9**±**1.1**
CiteSeer	0%	72.0±0.6	73.3±0.8	73.6±0.9	73.6±0.6	73.8±0.6	**73.9**±**0.6**
	5%	70.9±0.6	72.9±0.8	72.8±0.5	73.3±0.6	72.9±0.5	**73.5**±**0.7**
	10%	67.6±0.9	70.6±0.5	70.2±0.6	72.4±0.9	72.6±0.4	**72.9**±**0.5**
	15%	64.5±1.1	69.0±1.1	70.2±0.6	71.3±1.5	71.9±0.7	**72.1**±**0.8**
	20%	63.0±1.5	61.0±1.2	64.9±1.0	64.7±0.8	64.7±0.8	**65.2**±**0.8**
Polblogs	0%	95.7±0.4	95.4±0.2	95.4±0.2	95.8±0.3	95.8±0.3	**95.9**±**0.3**
	5%	73.1±0.8	83.7±1.5	82.8±0.3	82.8±0.4	83.0±0.3	**83.9**±**0.4**
	10%	70.7±1.1	76.3±0.9	73.7±0.3	81.5±0.2	81.6±0.3	**82.7**±**0.3**
	15%	65.0±1.9	68.8±1.1	68.9±0.9	77.8±1.9	78.7±0.5	**79.0**±**0.8**
	20%	51.3±1.2	51.5±1.6	65.5±0.7	77.4±0.2	77.5±0.2	**78.3**±**0.3**
Pubmed	0%	87.2±0.1	83.7±0.4	88.1±0.1	88.1±0.1	88.1±0.1	**88.7**±**0.1**
	5%	83.1±0.1	78.0±0.4	87.1±0.2	87.1±0.2	87.1±0.2	**87.6**±**0.2**
	10%	81.2±0.1	74.0±0.4	86.6±0.1	86.3±0.1	87.0±0.1	**87.5**±**0.1**
	15%	78.7±0.1	71.1±0.5	85.7±0.2	85.5±0.1	86.4±0.2	**86.6**±**0.2**
	20%	77.4±0.2	68.2±1.0	85.8±0.1	85.4±0.1	86.4±0.2	**86.7**±**0.2**

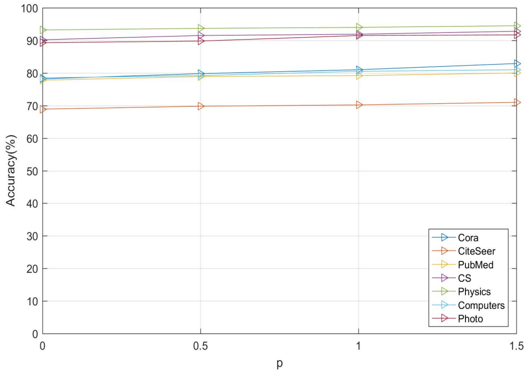

FIGURE 7. The sensitivity of *p*.

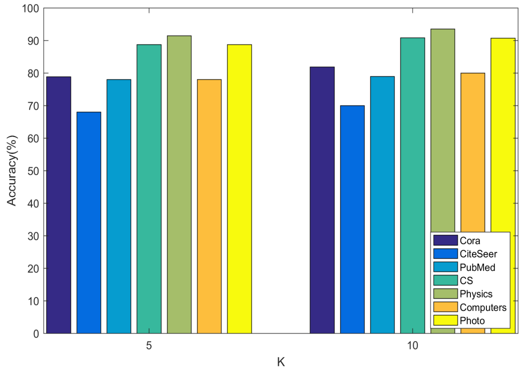

FIGURE 8. The performance after ablation study with respect to the case of $\lambda_2 = 0$.

E. ROBUSTNESS UNDER GRAPH ADVERSARIAL ATTACK

The robust performance of the proposed Schatten GNN under an adversarial graph attack is examined. The attack harms the GNN model's performance by slightly modifying the underlying graph structure. We adopt the MetaAttack [36] from DeepRobust [37], which is a PyTorch library for adversarial attacks and defenses, to generate graphs of adversarial attack for Cora, CiteSeer, Polblogs and PubMed. We randomly split 10%/10%/80% nodes of these data into a training set, validation set and test set, respectively. The statistics of the modified graph are listed in Table 3. By the works [36], [38], the largest connected component (LCC) is used in the adversarial graphs. We are only concerned with the robustness in the sense of l_1-based graph smoothing regardless of adversarial defense. The experimental results are listed in the Table 4. The robustness of the proposed approaches is best. This may be attributed to the balance of the Schatten p-norm for $p \in (0, 2)$ and l_2 norm-based graph smoothing.

F. SUMMARIZATION AND ANALYSIS

The GCN [5] can be regarded as the base architecture in experiments. In fact, it can be generated by taking λ_2 in the model (19) and performing a gradient scheme with particular stepsize [12]. The proposed network architecture of Schatten GNN is determined by the graph signal denoising problem (19). In this sense, GCN is just a special case of Schatten GNN when $\lambda_2 = 0$.

Based on the experimental results in Table 2 and Table 4, the proposed Schatten GNN achieve the best performance when compared with recent works on GNNs. By [26], the Schatten p-norm is between Frobenius norm and nuclear norm when p lies in the interval $[1, 2)$. The Frobenius norm induces the l_2 sparsity. The nuclear norm has an affinity with low-rank property. Hence the Schatten p-norm indicates the mixed property of l_2 sparsity and low-rank property. Elastic GNNs [12] impose l_1 sparsity. Hence Elastic GNNs consider mixed property of l_1 and l_2 sparsity. We can see that the performance of the proposed methods outperforms the Elastic GNNs a little. The increment of performance is small. This is probably that the low-rank property of the signals is a bit better than the property of l_1 sparsity.

IX. CONCLUSION

Message passing networks are some of the most important graph neural networks. In this paper, we proposed a novel message-passing networks, called the Schatten graph neural network, which is derived from a new proposed graph signal denoising problem with the Schatten p-norm. There is difficulty in solving the proximal operator in the intermediate steps, and we proposed a novel fixed-point iteration scheme for which the linear convergence rate $\mathcal{O}(\rho^k)$ was theoretically proved. Extensive experiments indicated that the proposed approach outperforms the state-of-the-art approaches and is robust under graph adversarial attacks.

REFERENCES

[1] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.

[2] N. Pancino, A. Rossi, G. Ciano, G. Giacomini, S. Bonechi, P. Andreini, F. Scarselli, M. Bianchini, and P. Bongini, "Graph neural networks for the prediction of protein-protein interfaces," in *Proc. ESANN*, 2020, pp. 127–132.

[3] C. Chen, K. Li, S. G. Teo, X. Zou, K. Wang, J. Wang, and Z. Zeng, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. ESANN*, 2020, pp. 485–492.

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2021.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[6] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. NeurIPS*, 2018, pp. 2428–2438.

[7] J. Wu, J. He, and J. Xu, "DEMO-Net: Degree-specific graph neural networks for node and graph classification," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 406–415.

[8] L. Xu, W.-D. Xi, and C.-D. Wang, "Session-based recommendation with heterogeneous graph neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 346–353.

[9] V. K. Garg, S. Jegelka, and T. Jaakkola, "Generalization and representational limits of graph neural networks," 2020, *arXiv:2002.06157*.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[11] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," 2020, *arXiv:2010.01777*.

[12] X. Liu, W. Jin, Y. Ma, Y. Li, H. Liu, Y. Wang, M. Yan, and J. Tang, "Elastic graph neural networks," in *Proc. ICML*, 2021, pp. 6837–6849.

[13] P. Chen, J. Huang, and X. Zhang, "A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration," *Inverse Problems*, vol. 29, no. 2, Feb. 2013, Art. no. 025011.

[14] L. Sun, Y. Dou, C. Yang, J. Wang, P. S. Yu, L. He, and B. Li, "Adversarial attack and defense on graph data: A survey," 2020, *arXiv:1812.10528*.

[15] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *Proc. ICML*, 2018, pp. 1115–1124.

[16] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Jan. 2020.

[17] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4130–4137.

[18] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its Oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[19] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, May 1993.

[20] J. Trzasko and A. Manduca, "Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization," in *Proc. TMI*, 2009.

[21] A. Kovnatsky, K. Glashoff, and M. M. Bronstein, "MADMM: A generic algorithm for non-smooth optimization on manifolds," in *Proc. ECCV*, 2016, pp. 680–696.

[22] R. A. Polyak, "Smooth optimization methods for minimax problems," *SIAM J. Control Optim.*, vol. 26, no. 6, pp. 1274–1286, Nov. 1988.

[23] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[24] A. Ghosh, R. K. Maity, A. Mazumdar, and K. Ramchandran, "Communication efficient distributed approximate Newton method," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1000–1008.

[25] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[26] C. Xu, Z. Lin, and H. Zha, "A unified convex surrogate for the Schatten-p norm," in *Proc. AAAI*, 2017, pp. 926–932.

[27] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 1st ed. Cham, Switzerland: Springer, 2011.

[28] L. Narici and E. Beckenstein, "The hahn-banach theorem: The life and times," *Topol. Appl.*, vol. 77, no. 2, pp. 193–211, Jun. 1997.

[29] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, p. 93, 2008.

[30] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," 2018, *arXiv:1811.05868*.

[31] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. Election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discovery (LinkKDD)*, 2005, pp. 36–43.

[32] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. PMLR*, 2019, pp. 6861–6871.

[33] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, *arXiv:1706.02216*.

[34] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized PageRank," 2018, *arXiv:1810.05997*.

[35] S. Yao, D. Yu, and X. Jiao, "Perturbing eigenvalues with residual learning in graph convolutional neural networks," in *Proc. ACML*, 2021, pp. 1569–1584.

[36] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2847–2856.

[37] Y. Li, W. Jin, H. Xu, and J. Tang, "DeepRobust: A PyTorch library for adversarial attacks and defenses," 2020, *arXiv:2005.06149*.

[38] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proc. KDD*, 2020, pp. 66–74.

[39] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted Schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, Aug. 2016.

[40] L. Liu, W. Huang, and D.-R. Chen, "Exact minimum rank approximation via Schatten p-norm minimization," *J. Comput. Appl. Math.*, vol. 267, pp. 218–227, Feb. 2014.

[41] I. Loris and C. Verhoeven, "On a generalization of the iterative soft-thresholding algorithm for the case of nonseparable penalty," in *Inverse Problems*, vol. 27, no. 12, 2011, Art. no. 125007.

[42] Z. Li and M. Yan, "New convergence analysis of a primal-dual algorithm with large stepsizes," 2017, *arXiv:1711.06785*.

[43] L. Zhou, Q. Zeng, and B. Li, "Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series," *IEEE Access*, vol. 10, pp. 40967–40978, 2022.

[44] C. Xu and W. Xu, "Causal structure learning with one-dimensional convolutional neural networks," *IEEE Access*, vol. 9, pp. 162147–162155, 2021.

[45] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," in *Proc. ICML*, 2021, pp. 9323–9332.

[46] A. Tomy, M. Razzanelli, F. Di Lauro, D. Rus, and C. Della Santina, "Estimating the state of epidemics spreading with graph neural networks," in *Nonlinear Dynamics*. Springer, Jan. 2022.

YOUFA LIU received the M.S. degree in mathematics from the Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Beijing, China, in 2017, and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2020. He is currently a Lecturer with the College of Informatics, Huazhong Agricultural University, China. He has published some articles in top journals, such as IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, and *INS*. His research interests include machine learning and computer vision.

YONGYONG CHEN (Member, IEEE) received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Macau, Macau, in 2020. He is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He has published more than 30 research papers in top-tier journals and conferences, including IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Geoscience and Remote Sensing, IEEE Transactions on Computational Imaging, IEEE Journal of Selected Topics in Signal Processing, *Pattern Recognition*, and *ACM MM*. His research interests include image processing, data mining, and computer vision.

GUO CHEN received the B.S. degree in international economics and trade and the M.S. degree in world economics from Hubei University, in 2012 and 2015, respectively, and the Ph.D. degree in world economics from Wuhan University, in 2021. She is currently a Postdoctoral Researcher with the School of Business of Theoretical Economics, Hubei University of China. Her research interests include application of machine learning in global value chain and productivity, and environmental economics.

JIAWEI ZHANG is currently pursuing the master's degree with the College of Informatics, Huazhong Agricultural University, China. His current interests include machine learning and bioinformatics.