Enhancing Locally Adaptive Smoothing of Graph Neural Networks Via Laplacian Node Disagreement

Yu Wang, Liang Hu, Xiaofeng Cao, Yi Chang, *Senior Member, IEEE*, and Ivor W. Tsang, *Fellow, IEEE*

***Abstract*—Graph neural networks (GNNs) are designed to perform inference on data described by graph-structured node features and topology information. From the perspective of graph signal denoising, the typical message passing schemes of GNNs act as a globally uniform smoothing that minimizes disagreements between embeddings of connected nodes. However, the level of smoothing over different regions of the graph should be different, especially for those inter-class regions. This deviation limits the expressiveness of GNNs, and then renders them fragile to over-smoothing, long-range dependencies, and non-homophily settings. In this paper, we find that the node disagreements of initial graph features can present more trustworthy constraints on node embeddings, thereby enhancing the locally adaptive smoothing of GNNs. To spread the inherent disagreements of nodes, we propose the Laplacian node disagreement to jointly measure the initial features and output embeddings. With such a measurement, we then present a new graph signal denoising objective deriving a more effective message passing scheme and further incorporate it into the GNN architecture, named Laplacian node disagreement-based GNN (LND-GNN). Learning from its output node representations, we integrate an auxiliary disagreement constraint into the overall classification loss. Experiments demonstrate the expressive ability of LND-GNN in the downstream semi-supervised node classification task.**

***Index Terms*—Graph neural networks, locally adaptive smoothing, message passing scheme, node disagreement.**

Manuscript received 14 August 2022; revised 4 April 2023; accepted 3 August 2023. Date of publication 8 August 2023; date of current version 6 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62206108, in part by Key scientific and technological R&D Plan of Jilin Province of China under Grant 20230201066GX, in part by the NSF of China under Grants 61976102 and U19A2065, in part by Maritime AI Research Programme under Grant SMI-2022-MTP-06, and in part by AI Singapore OTTC under Grant AISG2-TC-2022-006. Recommended for acceptance by X. Chen. *(Corresponding author: Xiaofeng Cao.)*

Yu Wang is with the College of Computer Science and Technology, and the School of Artificial Intelligence, Jilin University, Changchun, Jilin 130012, China (e-mail: yu_w18@mails.jlu.edu.cn).

Liang Hu is with the College of Computer Science and Technology, Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun, Jilin 130012, China (e-mail: hul@jlu.edu.cn).

Xiaofeng Cao is with the School of Artificial Intelligence, Jilin University, Changchun, Jilin 130012, China, and also with the Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China (e-mail: xiaofengcao@jlu.edu.cn).

Yi Chang is with the School of Artificial Intelligence, Jilin University, Changchun, Jilin 130012, China, also with the Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China, and also with the International Center of Future Science, Jilin University, Changchun, Jilin 130012, China (e-mail: yichang@jlu.edu.cn).

Ivor W. Tsang is with the Centre for Frontier AI Research and the Institute of High-Performance Computing, Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: ivor.tsang@uts.edu.au).

Digital Object Identifier 10.1109/TKDE.2023.3303212

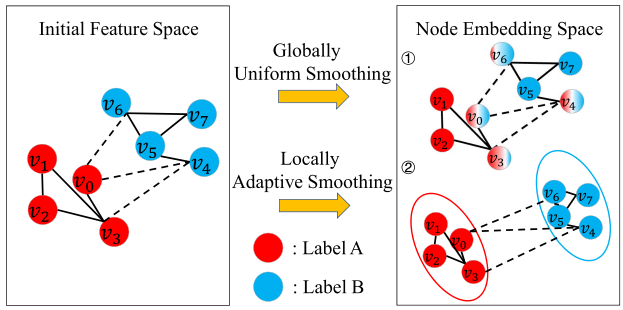

Fig. 1. Illustrations of globally uniform and locally adaptive smoothing, where the color indicates the label class, and the solid and dashed lines indicate intra-class and inter-class relationships, respectively. Globally uniform smoothing requires that node embeddings are smoothed with the consistent level shared across the whole graph, yielding the mixed label in inter-class nodes marked with mixed colors. Locally adaptive smoothing requires that node embeddings are smoothed with various levels spread across the subgraphs.

I. Introduction

NETWORKS are ubiquitous in the real world, such as social networks, biological networks, citation networks, etc [1], [2], [3]. Recently, the attempts utilizing deep neural networks to handle graph analysis, have attracted considerable attention in both academia and industry. Among those attempts, graph neural networks (GNNs), which learn the node representations by encoding node features and graph topology simultaneously, have achieved promising performance in various graph-related tasks, including node classification [4], [5], link prediction [6], [7], and graph classification [2], [8].

Most GNNs adopt a message passing scheme [9] to learn the representation for each node by iteratively aggregating the representations of all its neighbors. Essentially, the message passing of GNNs can be generalized as a low-pass filter that globally smoothes node features over the graph topology [10], [11], where globally uniform smoothing [12] denotes that the level of smoothing is shared across the whole graph. For instance, GCN [4] enforces globally uniform smoothing by aggregating feature information from all neighbors equally. Although common approaches with globally uniform smoothing have achieved satisfactory performance in semi-supervised node classification on homophily graphs, they often exhibit significant limitations, such as the over-smoothing problem [13], long-range dependencies [14], [15], and their sensitivity to edge perturbations on heterophily graphs [16], [17], [18] and adversarial attacks [19], [20].

Problem. To relieve these problems, locally adaptive smoothing [12], [21] across classes shows expressive ability, in which

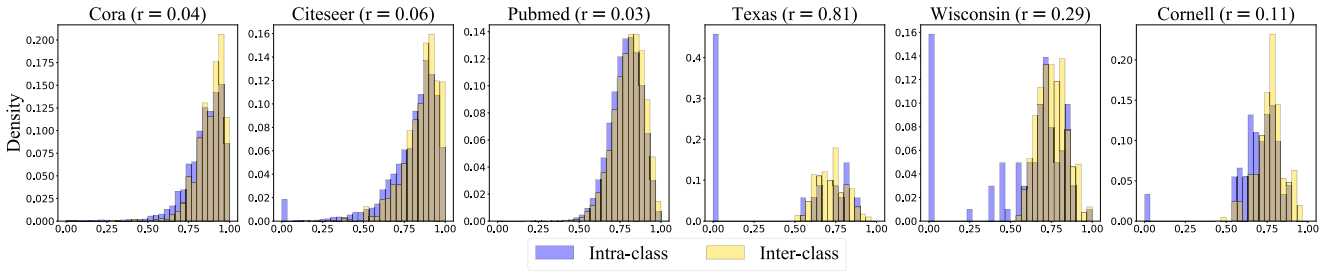

Fig. 2. Distributions of intra-class node disagreement and inter-class node disagreement in initial node features.

the motivation is that node features and labels can vary significantly across classes while being smooth within classes (Fig. 1). For instance, GAT [5] adopts adaptive weights for different neighboring nodes; DAGNN [22] and GPR-GNN [18] adaptively control the contributions of various receptive fields; ElasticGNN [21] facilitates locally adaptive smoothing by exploiting the sparsity tendency of ℓ_1-based graph smoothing. However, existing locally adaptive smoothing methods iteratively invoke the smoothed node embeddings of the former layer, inevitably degrading the constraint on node disagreement as the number of propagation layers increases, thus failing to completely solve the problems posed by globally uniform smoothing.

Motivation. In scenarios where globally uniform smoothing fails, disagreements of node embeddings across classes can derive locally adaptive smoothing. Generally, those connected nodes across different classes usually reveal more significant disagreement levels than nodes within the same class. Especially, this property is more trustworthy in its initial stage before being smoothed. For example, Fig. 2 shows several graphs with various distributions of node disagreement in the initial features (as measured by the metric proposed in Section IV-A, namely disagreement ratio r). We thus notice that the initial node features can impose more trustworthy constraints on node disagreement to enhance the locally adaptive smoothing of GNNs.

Our Solution. In this paper, we investigate Laplacian node disagreement that jointly measures the node disagreement from the perspectives of initial features and output embeddings, rather than only in terms of output embeddings, as in the popular local adaptive smoothing methods. Technically, we present a new graph signal denoising constrained by Laplacian node disagreement and derive a novel message passing scheme. Then, we incorporate it into a GNN architecture termed Laplacian node disagreement-based graph neural network (LND-GNN). After feeding the node embeddings generated by the LND-GNN into the downstream semi-supervised node classification, we further integrate an auxiliary disagreement constraint into the overall loss function to enhance the model training. Our contributions are summarized as follows:

- We present a novel concept of Laplacian node disagreement (LND) to measure node disagreement from the perspectives of initial node features and output embeddings.
- With this concept, we propose a novel GNN model, called LND-GNN, which enhances locally adaptive smoothing with a message passing scheme derived from graph signal denoising implemented via LND.
- Extensive experiments evaluate that LND-GNN can achieve state-of-the-art performance in varying scenarios, including homophily graphs, heterophily graphs, adversarial attacks, and long-range dependencies.

Organizations. The rest of this paper is organized as follows. Section II surveys the related work. Section III presents notations and necessary background. In Section IV, we introduce the proposed model in detail. The experimental results are reported in Section V. Finally, Section VI summarizes this paper.

II. Related Work

GNNs generalize convolutional neural networks (CNNs) to graph-structured data, among which the most representative is the message passing GNNs that propagate feature information over the graph topology via neighborhood aggregation [4], [9]. The aggregation operation can be replaced by various forms of attention mechanisms that learn the similarity between connected nodes [5], [23]. We next review the work of GNNs along two lines of designing message passing schemes, e.g., globally uniform smoothing and locally adaptive smoothing.

Globally Uniform Smoothing. Recent works have shown that the message passing GNNs can be regarded as low-pass graph signal filters, which derive globally uniform smoothing over the graph topology [10], [11]. Generally, globally uniform smoothing denotes that the intensity of smoothing is shared across the whole graph [12]. When feature information is aggregated from large receptive fields, globally uniform smoothing will lead to indistinguishable node embeddings from different classes, namely the over-smoothing problem [13], [24], [25]. To alleviate this problem, some researchers propose attaching residual connections to the propagation layer [26], [27], [28]. For example, APPNP [27] designs an improved message passing scheme based on personalized PageRank to construct a new GNN model; GCNII [28] proposes an extension of vanilla GCN with initial residual and identity mapping. In addition, globally uniform smoothing also limits the inference capability of GNNs on heterophily graphs, where connected nodes tend to belong to different classes. A typical solution is to jointly aggregate information from higher-order neighborhoods and immediate neighborhoods [16], [17].

Locally Adaptive Smoothing. Recently, some studies show that locally adaptive smoothing is beneficial for the inference of node embeddings due to the varying levels of smoothness across subgraphs [12], [21]. For instance, GAT [5] implicitly specifies different weights to different neighbor nodes, which facilitates

locally adaptive smoothing. DAGNN [22] adaptively balances the information from local and global neighborhoods for each node. GPR-GNN [18] adaptively exploits node features and graph topology, aggregating the node information from the structural neighborhood built by the continuous space underlying graphs. ElasticGNN [21] further enhances the locally adaptive smoothing of GNNs via ℓ_1-based graph smoothing. Moreover, UGNN [12] proposes a general framework that unifies several representative GNN models as natural instances of graph signal denoising problems. It also instantiates the framework with the attention mechanism as a new GNN model, ADA-UGNN, addressing adaptive local smoothness across nodes. However, these methods iteratively invoke the node embeddings smoothed in the former iteration, losing sensitivity to the disagreement in node features as the number of iterations increases, and thus may yield a sub-optimal conclusion.

III. Preliminaries

In this section, we introduce some necessary notations and backgrounds. Section III-A first presents notations of the graph. Section III-B then presents a brief background of the message passing scheme of GNNs, and Section III-C presents GNNs from the perspective of graph signal denoising.

A. Graph

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the node set $\mathcal{V}$ and the edge set $\mathcal{E}$. Suppose that each node is associated with a d_f-dimensional feature vector, and the initial node feature matrix is denoted as $\mathbf{X}_{in} \in \mathbb{R}^{n\times d_f}$, where n is the number of nodes. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ denote the adjacency matrix and $\mathbf{D}$ the diagonal degree matrix. Consequently, the adjacency matrix and the diagonal degree matrix with self-loop are defined as $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\hat{\mathbf{D}} = \mathbf{D} + \mathbf{I}$, respectively. The normalized graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{A}} = \mathbf{I} - \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}$. Generally, we use bold upper-case letters to denote matrices, such as the initial feature matrix $\mathbf{X}_{in}$ and the node embedding matrix in the hidden layer $\mathbf{H} \in \mathbb{R}^{n\times d}$, where d denotes the dimension of the node embedding, and bold lower-case letters to denote vectors, such as node v's feature vector $\mathbf{x}_v$ and the embedding of node v in the hidden layer $\mathbf{h}_v$. For clarity, Table I summarizes the main notations used in this paper.

B. Message Passing Scheme of GNNs

Message passing scheme [4], [9] is the key building block of GNNs to jointly encode node features and graph topology, and it learns a node embedding by iteratively propagating the information of its neighbors.

Following the message passing scheme of GNNs presented in [9], during the k-th layer propagation phase, the node embeddings $\mathbf{H}^{(k)} \in \mathbb{R}^{n\times d}$ are updated according to

$$\begin{aligned}\mathbf{H}^{(k)} &= \text{Propagation}(\mathbf{W}^{(k)}, \mathbf{H}^{(k-1)}, \tilde{\mathbf{A}}) \\ &= \sigma\left(\mathbf{W}^{(k)} \cdot \text{Aggregation}(\mathbf{H}^{(k-1)}, \tilde{\mathbf{A}})\right), \end{aligned} \tag{1}$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{d\times d}$ denotes a trainable transformation matrix of the k-th layer, Aggregation$(\cdot)$ denotes a neighborhood aggregation function, and σ denotes a non-linear activation function, e.g., a ReLU. Specifically, $\mathbf{H}^{(0)}$ is initialized as $\mathbf{H}^{(0)} = \mathbf{X}_{in}\mathbf{W}_e$, where $\mathbf{W}_e \in \mathbb{R}^{d_f\times d}$ is the transformation matrix for initial features. Most of the existing GNNs models adopt (1) to aggregate and transform node features, e.g., GCN [4], GAT [5], and GraphSAGE [29].

TABLE I
Summary of the Main Notations

$\mathcal{G}, \mathcal{V}, \mathcal{E}$	Graph, Node set, Edge set
n	Number of nodes
m	Number of nodes in training set
d_f	Dimension of initial feature
Θ	Model parameters
$\mathbf{X}_{in} \in \mathbb{R}^{n\times d_f}$	Initial feature matrix
$\mathbf{A} \in \mathbb{R}^{n\times n}$	Adjacency matrix
$\mathbf{D} \in \mathbb{R}^{n\times n}$	Diagonal degree matrix
$\mathbf{I} \in \mathbb{R}^{n\times n}$	Identity matrix
$\hat{\mathbf{A}} \in \mathbb{R}^{n\times n}$	Adjacency matrix with self-loop
$\hat{\mathbf{D}} \in \mathbb{R}^{n\times n}$	Diagonal degree matrix with self-loop
$\tilde{\mathbf{A}} \in \mathbb{R}^{n\times n}$	Normalized adjacency matrix with self-loop
$\mathbf{L} \in \mathbb{R}^{n\times n}$	Graph Laplacian matrix
K	Pre-specified total number of propagation layers
k	Index of k-th propagation layers
d	Pre-specified dimension of hidden embedding
$\mathbf{X} \in \mathbb{R}^{n\times d}$	Transformed feature matrix
$\mathbf{x}_i \in \mathbb{R}^{1\times d}$	Transformed feature vector of node i
$\mathbf{W}^{(k)} \in \mathbb{R}^{d\times d}$	Transformation matrix of the k-th layer
$\mathbf{H}^{(k)} \in \mathbb{R}^{n\times d}$	Embedding matrix of k-th hidden layer
$\mathbf{h}_i^{(k)} \in \mathbb{R}^{1\times d}$	Embedding vector of k-th hidden layer of node i
l	Dimension of label vector
$\mathbf{H}^* \in \mathbb{R}^{n\times l}$	Embedding matrix with the dimension of label
$\hat{\mathbf{y}}_i \in \mathbb{R}^{1\times l}$	Predicted label vector of node i
$\mathbf{y}_i \in \mathbb{R}^{1\times l}$	Ground-truth label vector of node i
$\mathbf{W}_e \in \mathbb{R}^{d_f\times d}$	Learnable matrix used to transform the initial features to the hidden embeddings
$\mathbf{W}_o \in \mathbb{R}^{d\times l}$	Learnable matrix used to transform the hidden embeddings to the output embeddings
α	Trade-off parameter used to tune the importance of the global constraint
β	Trade-off parameter used to tune the importance of the local constraint
γ	Trade-off parameter used to tune the importance of the label disagreement

C. Graph Signal Denoising: A New Perspective

Generally, the aggregation operations of GNNs models share a common goal that guarantees the feature smoothness of connected nodes [10]. With this purpose, recent work [12] has established a connection between many popular GNNs and graph signal denoising with Laplacian smoothing regularization, deriving a new perspective of GNNs. Specifically, the graph signal denoising can be formulated as

$$\arg\min_{\mathbf{H}} \mathcal{L}_{\mathbf{H}} = \|\mathbf{H} - \mathbf{X}\|_F^2 + \alpha \cdot \text{tr}[\mathbf{H}^\top \mathbf{L}\mathbf{H}], \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{H} \in \mathbb{R}^{n\times d}$ denotes the output node embedding matrix, $\mathbf{X} = \mathbf{X}_{in}\mathbf{W}_e \in \mathbb{R}^{n\times d}$ denotes the transformed node feature matrix, and α denotes a trade-off parameter. Moreover, $\|\mathbf{H} - \mathbf{X}\|_F^2$ denotes the *fitting constraint*, which guides the output embeddings $\mathbf{H}$ close to the transformed features $\mathbf{X}$, and $\text{tr}[\mathbf{H}^\top \mathbf{L}\mathbf{H}]$ denotes the *globally uniform smoothing constraint*, which guides the smoothing of

output embeddings $\mathbf{H}$ over graph $\mathcal{G}$. Next, we briefly present two representative examples.

Example 1: (GCN) For Graph Convolutional Network (GCN) [4], the aggregation operation of the message passing scheme is defined as

$$\mathbf{H}^{(k+1)} = \tilde{\mathbf{A}}\mathbf{H}^{(k)}, \tag{3}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$. From the perspective of graph signal denoising, it is can be considered as the iterative solution to minimize $\mathrm{tr}[\mathbf{H}^\top \mathbf{L}\mathbf{H}]$ [12]. Note that this case only considers the globally uniform smoothing and ignores the node feature information.

Example 2 (GCNII & APPNP): GCNII [28] and Approximated Personalized Propagation of Neural Predictions (APPNP) [27] both introduce an aggregation operation based on Personalized PageRank [30], which can be approximately written as

$$\mathbf{H}^{(k+1)} = (1-c)\cdot \tilde{\mathbf{A}}\mathbf{H}^{(k)} + c\cdot \mathbf{X}. \tag{4}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$. The aggregation operation can be considered as approximately solving the graph signal denoising of (2) with the trade-off parameter $\alpha = (\frac{1}{c} - 1)$ as

$$\arg\min_{\mathbf{H}} \mathcal{L}_{\mathbf{H}} = \|\mathbf{H} - \mathbf{X}\|_F^2 + (\frac{1}{c} - 1)\cdot \mathrm{tr}[\mathbf{H}^\top \mathbf{L}\mathbf{H}]. \tag{5}$$

Both the globally uniform smoothing and node feature fitting are considered in the graph signal denoising of (5), while locally adaptive smoothing is still ignored.

IV. Laplacian Node Disagreement-Based Graph Neural Network

In this section, we develop a concrete model called LND-GNN, which implements locally adaptive smoothing via Laplacian node disagreement. Specifically, Section IV-A presents the Laplacian node disagreement, Section IV-B integrates it into the typical graph signal denoising, Section IV-C derives a new message passing scheme deployed in GNNs from the optimization iterations of the enhanced graph signal denoising, Section IV-D integrates an auxiliary disagreement constraint into the overall loss of semi-supervised node classification, and Section IV-E presents the training process and the time complexity analysis.

A. Laplacian Node Disagreement

Smoothness is a metric that reflects the disagreement of node features [22], [31] or labels [16], [17], [31] in the graph. Compared with previous metrics on single property in the graph, a metric that jointly considers node features and labels may provide a better expression for the graph. Therefore, we further propose two smoothness metrics in terms of node disagreement:

Definition 1 (Intra-class Node Disagreement): Consider features $\mathbf{X}$ and labels $\mathbf{Y}$, we define the intra-class node disagreement as

$$\lambda_{\text{intra}} = \frac{\|\sum_{(v_i,v_j)\in\mathcal{E}\wedge y_i = y_j}(\mathbf{x}_i - \mathbf{x}_j)^2\|_1}{|\{(v_i,v_j):(v_i,v_j)\in\mathcal{E}\wedge \mathbf{y}_i = \mathbf{y}_j\}|\cdot d}, \tag{6}$$

where $\|\cdot\|_1$ denotes the Manhattan norm, $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the feature and the label of node v_i, and d denotes the dimension of node features.

Definition 2 (Inter-class Node Disagreement): Consider features $\mathbf{X}$ and labels $\mathbf{Y}$, we define the inter-class node disagreement as

$$\lambda_{\text{inter}} = \frac{\|\sum_{(v_i,v_j)\in\mathcal{E}\wedge y_i \neq y_j}(\mathbf{x}_i - \mathbf{x}_j)^2\|_1}{|\{(v_i,v_j):(v_i,v_j)\in\mathcal{E}\wedge \mathbf{y}_i \neq \mathbf{y}_j\}|\cdot d}. \tag{7}$$

According to Definitions 1 and 2, λ_{intra} and λ_{inter} denote the disagreement between connected nodes with the same label and distinct labels, respectively, where smaller node disagreement indicates higher smoothness between connected nodes. Generally, inter-class node pairs reveal more significant disagreement levels than intra-class node pairs. To more explicitly express the disagreement level of the whole graph, based on the two node disagreement metrics in (6) and (7), we further define a metric r named disagreement ratio.

Definition 3 (Disagreement Ratio): To measure the disagreement level between intra-class node pairs and inter-class node pairs in the graph, we define the disagreement ratio r as

$$r = \frac{\lambda_{\text{inter}} - \lambda_{\text{intra}}}{\lambda_{\text{intra}}}. \tag{8}$$

According to Definition 3, a larger r indicates that connected nodes across different classes in the graph have a more significant disagreement level than connected nodes within the same class. Here, node disagreement is negatively related to the smoothness of the graph. From the perspective of locally adaptive smoothing, a larger r means that the inter-class edges should be assigned a lower level of smoothing. Conversely, a smaller r means that the inter-class node pairs have fewer disagreements and should be assigned a larger level of smoothing. Node disagreements in the initial features are more trustworthy, thus we conjecture that locally adaptive smoothing works better on graphs with large r in terms of initial features. A statistical presentation of the disagreement ratios on different graphs is shown in Fig. 2, and a detailed experimental analysis is provided in Section V-H.

In this work, we aim to exploit node disagreement in the initial node features to provide more trustworthy constraints on the node disagreement in the output node embeddings, enhancing the locally adaptive smoothing of GNNs. To jointly measure node disagreement from the perspectives of initial features and output embeddings, we formally define the Laplacian node disagreement as

Definition 4 (Laplacian Node Disagreement): For each pair of connected nodes, the Laplacian node disagreement denotes the disagreement between the node disagreement in the output embeddings and that in the initial features.

Remark 1: As an example, for a pair of connected nodes $(v_i, v_j) \in \mathcal{E}$, $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the node features, $\mathbf{h}_i$ and $\mathbf{h}_j$ denote the output node embeddings, the Laplacian node disagreement can be expressed as $\|(\mathbf{h}_i - \mathbf{h}_j) - (\mathbf{x}_i - \mathbf{x}_j)\|_2^2$. In a graph with a large disagreement ratio r, if $\mathbf{x}_i$ and $\mathbf{x}_j$ have a relatively significant disagreement, the edge (v_i, v_j) is more likely to be an inter-class edge. Accordingly, tightening the

Laplacian node disagreement allows the output embeddings $\mathbf{h}_i$ and $\mathbf{h}_j$ to retain node disagreement as much as possible, that is, a lower level of graph smoothing will be adopted adaptively.

Overall, by tightening the Laplacian node disagreement, the more trustworthy disagreement in the initial features can be spread to the output node embeddings to enhance locally adaptive smoothing.

B. Graph Signal Denoising Via Laplacian Node Disagreement

In this section, we integrate the Laplacian node disagreement into the graph signal denoising of (2) to enhance the locally adaptive smoothing. Specifically, the joint constraint of Laplacian node disagreement on the transformed features $\mathbf{X} \in \mathbb{R}^{n\times d}$ and output embeddings $\mathbf{H} \in \mathbb{R}^{n\times d}$ in the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is defined as

$$\sum_{(v_i,v_j)\in\mathcal{E}} \|(\mathbf{h}_i - \mathbf{h}_j) - (\mathbf{x}_i - \mathbf{x}_j)\|_2^2, \tag{9}$$

then to be consistent with the *globally uniform smoothing constraint* in (2), it can be written as

$$\sum_{(v_i,v_j)\in\mathcal{E}} \|(\mathbf{h}_i - \mathbf{x}_i) - (\mathbf{h}_j - \mathbf{x}_j)\|_2^2. \tag{10}$$

Assuming that we adopt the unnormalized graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, the Laplacian node disagreement constraint can be defined as

$$\sum_{(v_i,v_j)\in\mathcal{E}} \|(\mathbf{h}_i - \mathbf{x}_i) - (\mathbf{h}_j - \mathbf{x}_j)\|_2^2 = \mathrm{tr}[(\mathbf{H}-\mathbf{X})^\top \mathbf{L}(\mathbf{H}-\mathbf{X})]. \tag{11}$$

Then, to enhance the locally adaptive smoothing via Laplacian node disagreement constraint, we propose the improved graph signal denoising as

$$\begin{aligned}\arg\min_{\mathbf{H}} \mathcal{L}_{\mathbf{H}} = \|\mathbf{H}-\mathbf{X}\|_F^2 + \alpha \cdot \mathrm{tr}[\mathbf{H}^\top \mathbf{L}\mathbf{H}] \\ + \beta \cdot \mathrm{tr}[(\mathbf{H}-\mathbf{X})^\top \mathbf{L}(\mathbf{H}-\mathbf{X})],\end{aligned} \tag{12}$$

where $\|\mathbf{H}-\mathbf{X}\|_F^2$ denotes the *fitting constraint*, which guides the output embeddings $\mathbf{H}$ close to the transformed features $\mathbf{X}$, $\mathrm{tr}[\mathbf{H}^\top\mathbf{L}\mathbf{H}]$ denotes the *globally uniform smoothing constraint*, which guides the output embeddings $\mathbf{H}$ similar between connected nodes over $\mathcal{G}$, and $\mathrm{tr}[(\mathbf{H}-\mathbf{X})^\top\mathbf{L}(\mathbf{H}-\mathbf{X})]$ denotes the *locally adaptive smoothing constraint*, which enhances the locally adaptive smoothing by tightening Laplacian node disagreement.

Normalization. Commonly, to alleviate the problems of numerical instability and exploding/vanishing gradients caused by the repeated application of aggregation operations in GNNs, a *renormalization trick* [4] is applied to Laplacian matrix. Following the *renormalization trick* in GCN [4]: $\mathbf{L} = \mathbf{I} - \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2} = \mathbf{I} - \tilde{\mathbf{A}}$, where $\hat{\mathbf{A}}$ and $\hat{\mathbf{D}}$ are the matrices with self-loop, we update the unnormalized Laplacian matrix $\mathbf{L}$ of (12) to the normalized version.

C. A New Message Passing Scheme

The message passing scheme can be derived from the gradient descent iterations of the graph signal denoising [12].

Theorem 1: When we adopt the normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{A}}$, the aggregation operation of the message passing scheme approximately solves the graph signal denoising problem (12) by iterative gradient descent with stepsize $b = \frac{1}{2(1+\alpha+\beta)}$. Thus, the message passing scheme could be derived as

$$\begin{aligned}\mathbf{H}^{(k+1)} = \frac{\alpha+\beta}{1+\alpha+\beta}\cdot\tilde{\mathbf{A}}\mathbf{H}^{(k)} + \frac{1}{1+\alpha+\beta}\cdot\mathbf{X} \\ + \frac{\beta}{1+\alpha+\beta}\cdot\mathbf{L}\mathbf{X}.\end{aligned} \tag{13}$$

Proof: To solve the minimization problem of (12) iteratively, we take the iterative gradient method with stepsize b. Specifically, the k-th step is as follows:

$$\begin{aligned}\mathbf{H}^{(k+1)} &= \mathbf{H}^{(k)} - b\cdot\left.\frac{\partial\mathcal{L}_{\mathbf{H}}}{\partial\mathbf{H}}\right|_{\mathbf{H}=\mathbf{H}^{(k)}} \\ &= (1-2b-2b\alpha-2b\beta)\cdot\mathbf{H}^{(k)} \\ &\quad + 2b\cdot\mathbf{X} + 2b\beta\cdot\mathbf{L}\mathbf{X} \\ &\quad + (2b\alpha+2b\beta)\cdot\tilde{\mathbf{A}}\mathbf{H}^{(k)}\end{aligned}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$. When we set the stepsize b as $\frac{1}{2+2\alpha+2\beta}$, we have the following iterative steps:

$$\begin{aligned}\mathbf{H}^{(k+1)} = \frac{\alpha+\beta}{1+\alpha+\beta}\cdot\tilde{\mathbf{A}}\mathbf{H}^{(k)} + \frac{1}{1+\alpha+\beta}\cdot\mathbf{X} \\ + \frac{\beta}{1+\alpha+\beta}\cdot\mathbf{L}\mathbf{X}.\end{aligned}$$

Incorporating the message passing scheme of (13) derived from the graph signal denoising of (12) into the graph neural network, we propose Laplacian node disagreement-based GNN (LND-GNN). Specifically, following [11], [22], [27], we decouple the embedding transformation from the aggregation operation so that a deeper model can be built without negatively affecting performance. Formally, the mathematical expression of LND-GNN is defined as

$$\begin{aligned}&\mathbf{X} = \sigma(\mathbf{X}_{in}\mathbf{W}_e) && \in\mathbb{R}^{n\times d},\\ &\mathbf{H}^{(K)} = \mathrm{Propagation}^K(\mathbf{X},\tilde{\mathbf{A}}) && \in\mathbb{R}^{n\times d},\\ &\mathbf{H}^* = \mathbf{H}^{(K)}\mathbf{W}_o && \in\mathbb{R}^{n\times l},\\ &\hat{\mathbf{Y}} = \mathrm{Softmax}(\mathbf{H}^*) && \in\mathbb{R}^{n\times l},\end{aligned} \tag{14}$$

where $\mathbf{W}_e$, $\mathbf{W}_o$ are the transformation matrices, $\sigma(\cdot)$ is the nonlinear activation function and $\mathrm{Propagation}(\cdot)$ denotes the propagation function formulated by (13) with initial $\mathbf{H}^{(0)} = \mathbf{X}$. Besides, $\hat{\mathbf{Y}}$ denotes the label predictions, where l denotes the dimension of label vector.

D. Semi-Supervised Node Classification

With the label predictions $\hat{\mathbf{Y}}$ of (14), we can straightly compute the task loss of the semi-supervised node classification task in an end-to-end manner. To further enhance the model training,

an auxiliary constraint is integrated into the overall loss of node classification.

Task Loss. For the semi-supervised node classification task, given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a set of ground-truth labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\}$ for node subset $\mathcal{V}_{tr} = \{v_1, v_2, \ldots, v_m\}$, the task loss can be formulated by cross-entropy as:

$$\mathcal{L}_t = \frac{1}{m} \sum_{i=1}^{m} \text{Cross Entropy}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right), \tag{15}$$

where $\hat{\mathbf{y}}_i$ denotes the predicted label of node v_i yielded by LND-GNN, $\mathbf{y}_i$ denotes the ground-truth label of node v_i and is an one-hot vector that the k-th element is 1 if the node i belongs to class k.

Loss for Laplacian Node Disagreement Constraint. To make the node representations more powerful, we further apply the Laplacian node disagreement on the predicted labels. We consider $\mathbf{H}^* \in \mathbb{R}^{n \times c}$ of (14) as the predicted labels, where each row denotes the predicted probability distribution for each node's class. Then, we transform the node features $\mathbf{X}$ into node representations with the same dimensions as the predicted labels, denoted as $\mathbf{Y}'$, by multiplying the trainable transformation weight matrix $\mathbf{W}_o$, which is formally expressed as follows:

$$\mathbf{Y}' = \mathbf{X}\mathbf{W}_o. \tag{16}$$

With the transformed initial features and predicted labels, we formulate the following loss to constrain the node disagreement:

$$\mathcal{L}_d = \frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} \left((\mathbf{h}_i^* - \mathbf{h}_j^*) - (\mathbf{y}_i' - \mathbf{y}_j')\right)^2, \tag{17}$$

where $\mathbf{h}_i^*, \mathbf{h}_j^*$ denote the predicted labels of node v_i, v_j, respectively, and $\mathbf{y}_i', \mathbf{y}_j'$ denote the node representations of node v_i, v_j, respectively.

Overall Loss. Finally, we integrate the task loss and the auxiliary disagreement constraint to form the overall loss as follows:

$$\mathcal{L} = \mathcal{L}_t + \gamma \cdot \mathcal{L}_d, \tag{18}$$

where γ is a hyperparameter to control the importance of the disagreement constraint in the loss function.

Connection to Label Propagation. Recently, there has been some work on unifying message passing schemes in GNNs and label propagation [32]. The idea is that both GNNs methods and Label Propagation Algorithm (LPA) are message passing algorithm on graphs, thus they may be unified for the semi-supervised node classification. As we will show next, the overall loss constrained by Laplacian node disagreement in (18) is also motivated from this.

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a set of ground-truth labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\}$ for node subset $\mathcal{V}_{tr} = \{v_1, v_2, \ldots, v_m\}$, the well-known LPA [33] learns a mapping function $u : \mathcal{V} \to \mathcal{L}_c$ to predict the labels of all nodes in the graph. Motivated by the intuition that nearby nodes in the graph may have similar labels, LPA presents that all nodes iteratively propagate labels to their neighbors to minimize the energy function

$$\min_u E[u] = \frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} \left(u(v_i) - u(v_j)\right)^2, \quad \text{s.t. } u(v_i) = \mathbf{y}_i|_{1 \le i \le m}. \tag{19}$$

Recent work [34] further proposes to modify the LPA by introducing constraints in the form of a guidance field [35]. Let a vector field $\mathbf{v}$ defined over $\mathcal{G}$ be a *guidance field* to modify the problem of (19) as

$$\min_u E[u] = \frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} \left(u(v_i) - u(v_j) - \mathbf{v}_{ij}\right)^2, \quad \text{s.t. } u(v_i) = \mathbf{y}_i|_{1 \le i \le m}. \tag{20}$$

Now, if we choose the discrete gradient field on the node representations $\mathbf{Y}'$ over graph $\mathcal{G}$ as the *guidance field* $\mathbf{v}$, that is

$$\text{for each } (v_i, v_j) \in \mathcal{E}, \mathbf{v}_{ij} = \mathbf{y}_i' - \mathbf{y}_j', \tag{21}$$

the problem of (20) becomes quite similar to our overall loss of (18). Specifically, the energy function corresponds to the node disagreement loss $\mathcal{L}_d$, and the task loss $\mathcal{L}_t$ serves as the boundary condition. In summary, our proposed overall loss function for semi-supervised node classification constrained by Laplacian node disagreement can be regarded as motivated by the LPA constrained by the *guidance field* defined by node disagreement in the initial features.

An illustration of our proposed LND-GNN for the semi-supervised node classification task is provided in Fig. 3. We first utilize a learnable matrix $\mathbf{W}_e$ for feature transformation, and apply the message passing scheme defined in (13) to the transformed features for information propagation. Then, the transformed feature $\mathbf{X}$ and the aggregated feature $\mathbf{H}^{(K)}$ are transformed to the label space as $\mathbf{Y}'$ and $\mathbf{H}^*$, respectively, utilizing a shared learnable matrix $\mathbf{W}_o$. Finally, the overall loss for the semi-supervised node classification task is calculated to train the LND-GNN model.

E. Model Learning Process

Our proposed LND-GNN model is trained with (18) as the loss function and optimized by the back-propagation algorithm. The pseudo-code for the training process of LND-GNN is given in Algorithm 1. Specifically, Lines 5-9 aim at transforming the node features and propagating the features over graph topology with the message passing scheme in (13), yielding label predictions; Line 10 computes the overall loss (18) of semi-supervised node classification and updates the model parameters, namely the transformation matrices $\mathbf{W}_e$ and $\mathbf{W}_o$.

Then, we perform the time complexity analysis on the proposed LND-GNN. First, the time complexity of the feature transformation $\mathbf{X}$ with dimension d for all n nodes can be calculated as $\mathcal{O}(n \cdot d_f \cdot d)$, where d_f is the dimension of the initial features. Second, for K propagation layers, we compute the feature aggregation $\tilde{\mathbf{A}}\mathbf{H}$ with matrix multiplication. Since the normalized adjacency matrix $\tilde{\mathbf{A}}$ is stored as a sparse matrix with s non-zero entities, an efficient implementation of feature

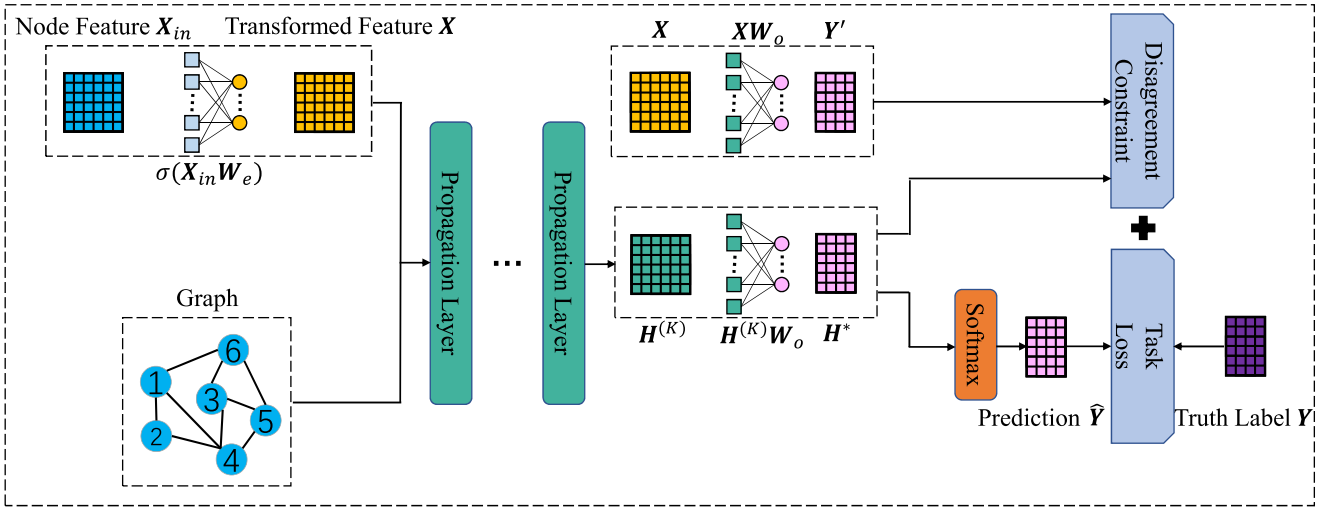

Fig. 3. Illustration of the proposed Laplacian Node Disagreement-based Graph Neural Network (LND-GNN).

Algorithm 1: Training Process of LND-GNN.

Input: Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with adjacency matrix $\mathbf{A}$, node feature matrix $\mathbf{X}_{in}$, node label matrix $\mathbf{Y}$, set of labeled nodes $\mathcal{V}_{tr}$, number of propagation K, hyperparameters α, β, and γ, maximal training epochs $MaxEpochs$.

Output: Model parameters: $\Theta = \{\mathbf{W}_e, \mathbf{W}_o\}$.

Randomly initialize model parameters Θ.
Compute the normalized adjacency matrix with self-loop $\tilde{\mathbf{A}}$.
$epochs \leftarrow 1$
while not converge and $epochs \leq MaxEpochs$ **do**
  $\mathbf{X} \leftarrow \sigma(\mathbf{X}_{in}\mathbf{W}_e)$
  $\mathbf{H}^{(K)} \leftarrow \text{Propagation}^K(\mathbf{X}, \tilde{\mathbf{A}})$ via (13)
  $\mathbf{H}^* \leftarrow \mathbf{H}^{(K)}\mathbf{W}_o$
  $\hat{\mathbf{Y}} \leftarrow \text{Softmax}(\mathbf{H}^*)$
  $\mathbf{Y}' \leftarrow \mathbf{X}\mathbf{W}_o$
  Optimize model parameters Θ by back propagation via (18) with $\mathbf{H}^*$, $\mathbf{Y}'$, $\hat{\mathbf{Y}}$, $\mathbf{Y}$, and hyperparameters as inputs.
end while

aggregation takes $\mathcal{O}(s \cdot d)$. Third, the embedding matrix of the hidden layer is transformed into the label matrix with the time complexity of $\mathcal{O}(n \cdot d \cdot l)$, where l is the number of node labels. Therefore, the overall time complexity of LND-GNN is $\mathcal{O}(n \cdot d_f \cdot d + K \cdot s \cdot d + n \cdot d \cdot l)$.

V. Experiments

In this section, we conduct extensive experiments on semi-supervised node classification to evaluate the performance of the proposed LND-GNN model. Specifically, we try to answer the following questions:

1) How does the proposed LND-GNN perform on graphs with homophily? Can it alleviate the over-smoothing problem? (Section V-C)
2) Beyond homophily in GNNs, how does the proposed LND-GNN perform in scenarios where current GNNs are limited, including heterophily graphs, adversarial attacks, and long-range dependencies? (Section V-D, E, and F)
3) Following the performance evaluation in various scenarios, more specifically, how does the proposed Laplacian node disagreement enhance the locally adaptive smoothing to improve the expressiveness of GNNs? (Section V-G and H)

TABLE II
Statistics of the Real-World Datasets

Dataset	Nodes	Edges	Features	Classes	Dis.ratio	Hom.ratio
Cora	2,708	5,429	1,433	7	0.04	0.81
CiteSeer	3,327	4,732	3,703	6	0.06	0.74
PubMed	19,717	44,338	500	3	0.03	0.8
ArXiv	169,343	1,116,243	128	40	0.05	0.66
Texas	183	309	1,703	5	0.81	0.11
Cornell	183	295	1,703	5	0.11	0.21
Wisconsin	251	499	1,703	5	0.29	0.3
Actor	7,600	33,544	931	5	0.16	0.22
Amazon	334,863	2,186,607	–	58	–	–

A. Datasets

We validate the performance of our proposed model on real-world datasets with the homophily ratio $\mathcal{H} = \frac{|\{(v_i,v_j):(v_i,v_j)\in\mathcal{E}\wedge \mathbf{y}_i=\mathbf{y}_j\}|}{|\mathcal{E}|}$ [16] ranging from strong homophily ($\mathcal{H} \approx 1$) to strong heterophily ($\mathcal{H} \approx 0$). The summary of the datasets is given in Table II. Note that the disagreement ratio (r) and homophily ratio ($\mathcal{H}$) have some correspondence since both consider the label types of the connected nodes, but the disagreement ratio additionally considers the feature disagreement.

Homophily Datasets: For homophily datasets, We consider three widely adopted citation networks, namely Cora, Citeseer, and Pubmed [3], plus ogbn-arXiv [36], a relatively large dataset. For the citation networks, nodes represent documents, and edges denote the citation relationship between two documents. The node features are the bag-of-words vectors of the corresponding

documents, and the labels are the research fields of the documents. For these datasets, we use the semi-supervised setting as provided in [4], [37]. And for ogbn-arXiv, we use the standard split setting from the Open Graph Benchmark (OGB) leaderboard [36].

Heterophily Datasets. We consider three sub-datasets of WebKB for heterophily datasets, namely Texas, Wisconsin, and Cornell [17], and an Actor co-occurrence network [38]. For the WebKB datasets, nodes represent web pages, and edges are hyperlinks between them. The node features are the bag-of-words vectors of web pages, and the labels indicate the five categories of web pages. For the Actor co-occurrence network, nodes are corresponding to actors, and edges represent the co-occurrence on the same Wikipedia page. The node features represent some keywords in the Wikipedia pages, and the labels correspond to words on the actor's Wikipedia pages. For these heterophily datasets, we adopt the processed node features, labels, and 10 random splits provided by [17].

B. Comparison Models

We compare the proposed LND-GNN with the following state-of-the-art models in the experiments:

GCN [4]: The aggregation operation of GCN is implemented by the product of a normalized adjacency matrix with a self-loop and a feature matrix, which can be considered fundamental global smoothing.

GAT [5]: GAT specifies different importance to different neighbors via the similarity between node embeddings, which can be considered as a fundamental locally adaptive smoothing.

SGC [11]: SGC eliminates the nonlinear transformations between hidden layers in GCN without sacrificing performance in downstream tasks.

GraphSAGE [29]: GraphSAGE generalizes the aggregation operation beyond averaging and aggregates only the feature information from the sub-sampled neighborhoods.

APPNP [27]: APPNP decouples aggregation and transformation by first transforming the node features into predictions, and then aggregating the predictions.

DAGNN [22]: DAGNN also decouples aggregation and transformation, and adaptively aggregates information from local and global neighborhoods.

GCNII [28]: GCNII extends the vanilla GCN with the initial residual and identity mapping to alleviate the over-smoothing problem.

Geom-GCN [17]: The aggregation operation in Geom-GCN is designed on the continuous space underlying the graph and aggregates information from structural neighborhoods.

GPR-GNN [18]: GPR-GNN performs feature aggregation for multiple steps and then adaptively learns the importance weights for each step.

H_2GCN [16]: H_2GCN accommodates both heterophily and homophily by effectively synthesizing ego- and neighbor-embedding separation, higher-order neighborhoods, and the combination of intermediate representations.

ADA-UGNN [12]: ADA-UGNN introduces a general regularization term with the attention mechanism into the graph signal denoising framework, and develops a resulting GNN model to enforce adaptive local smoothness.

ElasticGNN [21]: ElasticGNN proposes ℓ_1 and ℓ_2-based graph smoothing to enhance the smoothness adaptivity of GNNs.

It is worth noting that SGC, APPNP, DAGNN, and GCNII are proposed to alleviate the over-smoothing problem and obtain deeper neural networks, so they are usually inept at dealing with heterophily graphs. While Geom-GCN, GPR-GNN, and H_2GCN are specifically proposed to improve the handling of heterophily.

For experimental models, the hyperparameters are tuned from the following search space: learning rate in $\{0.1, 0.05\}$, weight decay in $\{1e-3, 5e-4, 1e-4, 5e-5\}$, number of layers in $\{2, 4, 8, 16, 32, 64\}$, hidden unit in $\{32, 64, 128, 256\}$, dropout in $\{0.2, 0.5, 0.7\}$. For ElasticGNN, parameters λ_1 and λ_2 are tuned in $\{0, 3, 6, 9\}$ as suggested by [21]. For three trade-off hyperparameters in the proposed LND-GNN model, α and β are tuned in $\{0.05, 0.1, 0.15, 0.2\}$ on homophily datasets and $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ on heterophily datasets, γ is tuned in $\{0.005, 0.01, 0.05, 0.25, 0.5\}$. All experiments are implemented in Pytorch with Adam optimizer [39]. For each experiment, we run 1000 epochs and apply early stopping with the patience of 100 epochs to train models. Besides, we repeat the experiment ten times and report the average performance and the standard variance.

TABLE III
PERFORMANCE OF NODE CLASSIFICATION ON HOMOPHILY DATASETS, (−) INDICATES OUT-OF-MEMORY ERROR, 95% CI DENOTES THE NUMBER OF IMPROVEMENTS THAT SATISFY THE SIGNIFICANCE TEST (P-VALUE < 0.05)

Model	Cora	Citeseer	Pubmed	ArXiv
GCN	81.3 ± 0.8	71.1 ± 0.4	78.8 ± 0.6	71.7 ± 0.3
GAT	83.1 ± 0.4	70.8 ± 0.5	79.1 ± 0.4	–
SGC	81.7 ± 0.1	71.3 ± 0.2	78.9 ± 0.1	69.8 ± 0.2
GraphSAGE	82.3 ± 0.5	71.8 ± 0.7	78.5 ± 0.4	71.5 ± 0.3
APPNP	83.3 ± 0.5	71.8 ± 0.4	80.1 ± 0.2	71.7 ± 0.3
DAGNN	84.4 ± 0.5	73.3 ± 0.6	80.5 ± 0.5	72.1 ± 0.3
GCNII	85.5 ± 0.5	73.4 ± 0.6	80.3 ± 0.4	**72.7 ± 0.2**
Geom-GCN	84.3 ± 0.5	71.7 ± 0.5	79.9 ± 0.4	–
GPR-GNN	83.5 ± 0.5	71.6 ± 0.6	78.6 ± 0.7	71.8 ± 0.3
H_2GCN	83.4 ± 0.5	71.8 ± 0.6	79.9 ± 0.4	–
ADA-UGNN	84.8 ± 0.5	70.9 ± 0.9	80.5 ± 0.6	–
ElasticGNN	83.6 ± 0.3	72.3 ± 0.6	80.5 ± 0.3	71.8 ± 0.2
LND-GNN	**85.6 ± 0.3**	**74.5 ± 0.5**	**80.9 ± 0.4**	72.4 ± 0.2
95% CI	11/12	12/12	12/12	6/8

C. Performance on Homophily Datasets

We first compare the performance of our models to baselines on four commonly used homophily datasets. The performance of node classification is presented in Table III.

Locally adaptive smoothing can preserve the disagreements of node embeddings across classes to improve the performance of node classification on homophily graphs. Thus, as shown in Table III, our models achieve state-of-the-art performance on three of four datasets, and nearly so on the fourth. In particular, the improvement over ElasticGNN also demonstrates the effectiveness of Laplacian node disagreement on locally adaptive smoothing. In addition, GCNII and DAGNN both achieve favorable results compared to other baselines, which verifies the

TABLE IV
PERFORMANCE OF NODE CLASSIFICATION WITH DIFFERENT NUMBER OF LAYERS

Dataset	Model	Number of Layers					
		2	4	8	16	32	64
Cora	GCN	**81.1**	80.4	69.5	64.9	60.3	28.7
	GCNII	82.2	82.6	84.2	84.6	85.4	**85.5**
	LND-GNN	83.9	85.1	85.2	**85.6**	85.5	85.0
Citeseer	GCN	**70.8**	67.6	30.2	18.3	25.0	20.0
	GCNII	68.2	68.9	70.6	72.9	**73.4**	73.4
	LND-GNN	72.1	72.5	72.7	73.5	74.2	**74.5**
Pubmed	GCN	**79.0**	76.5	61.2	40.9	22.4	35.3
	GCNII	78.2	78.8	79.3	**80.2**	79.8	79.7
	LND-GNN	79.0	79.5	80.2	80.5	**80.9**	80.7

TABLE V
PERFORMANCE OF NODE CLASSIFICATION ON HETEROPHILY DATASETS, 95% CI DENOTES THE NUMBER OF IMPROVEMENTS THAT SATISFY THE SIGNIFICANCE TEST (P-VALUE < 0.05)

Model	Texas	Wisconsin	Cornell	Actor
GCN	59.5 ± 5.2	59.8 ± 6.9	57.0 ± 4.6	30.3 ± 0.7
GAT	58.4 ± 4.4	55.3 ± 8.7	58.9 ± 3.3	26.3 ± 1.7
SGC	58.1 ± 7.5	54.3 ± 3.7	57.6 ± 4.0	26.8 ± 1.7
GraphSAGE	82.4 ± 6.1	81.2 ± 5.6	75.9 ± 5.0	34.3 ± 1.0
APPNP	65.4 ± 4.0	69.0 ± 5.1	63.5 ± 4.4	29.1 ± 1.5
DAGNN	64.3 ± 5.3	62.9 ± 3.7	61.1 ± 7.0	32.9 ± 1.5
GCNII	77.6 ± 3.8	80.4 ± 3.4	77.8 ± 3.7	36.4 ± 1.5
Geom-GCN	66.8 ± 2.7	64.5 ± 3.6	60.5 ± 3.6	31.6 ± 1.2
GPR-GNN	78.4 ± 4.3	82.9 ± 4.2	80.3 ± 8.1	34.6 ± 1.2
H_2GCN	84.8 ± 6.7	**86.6 ± 4.6**	82.2 ± 4.8	35.9 ± 1.0
ADA-UGNN	80.0 ± 6.1	84.1 ± 3.9	78.9 ± 6.9	33.7 ± 1.0
ElasticGNN	81.1 ± 4.7	84.3 ± 3.4	80.3 ± 5.9	34.5 ± 0.7
MLP	81.4 ± 5.0	85.1 ± 4.0	81.1 ± 6.3	35.7 ± 0.9
LND-GNN	**85.7 ± 4.0**	86.1 ± 3.1	**85.1 ± 4.4**	**36.6 ± 0.9**
95% CI	12/13	9/13	10/13	11/13

superiority of learning node representations from large receptive fields.

As aforementioned, in the vanilla GCN and its variants, a relatively large number of layers may cause the over-smoothing issue. In this part, we further conduct experiments on three citation network datasets with homophily to test the performance of our LND-GNN model with the different number of propagation layers. The vanilla GCN and GCNII, an extension for relieving the over-smoothing problem, are used as baselines. As shown in Table IV, the performance of the proposed LND-GNN model consistently improves and outperforms GCNII on both Citeseer and Pubmed as the number of layers increases. On Cora, our model peaks at 16 layers and then maintains a stable performance as the number of layers increases. Overall, the results show that with the proposed message passing scheme based on Laplacian node disagreement, we can relieve the over-smoothing problem in GNNs.

D. Performance on Heterophily Datasets

We also consider heterophily graphs further to evaluate the effectiveness of the proposed LND-GNN model. Compared to homophily graphs, heterophily graphs contain more label-inconsistent edges, which raises more significant challenges for the capabilities of the message passing scheme. Except for the eight baselines, we also include a competitive baseline under heterophily, namely Multi-layer Perceptron (MLP). MLP merely considers node feature information while ignoring the graph topology. Therefore it can be a threshold to measure whether the model has appropriately utilized the topological information to gain for the category inference. The performance on heterophily datasets is reported in Table V.

A myriad of label inconsistent edges dramatically limit the capabilities of globally uniform smoothing, thus we can observe that our LND-GNN model achieves significant advantages over the baselines. Locally adaptive smoothing based on the Laplacian node disagreement can constrain the disagreements in node embeddings while performing the smoothing operations to mitigate the misleading effect of label inconsistent edges. Specifically, H_2GCN and our LND-GNN model are the only models that outperform MLP on all datasets, which only considers the node features while ignoring graph topology. Compared to H $_2$GCN, which is tailored for heterophily graphs, our LND-GNN model achieves superior performance on three of four datasets, and nearly so on the fourth.

In addition, GraphSAGE achieves better performance than GCN and GAT, which is attributed to neighborhood sampling and the extension of averaging aggregation. Two deeper models, namely GPR-GNN and GCNII, outperform the above shallow models in most settings, which confirms the superiority of aggregating information from higher-order neighborhoods. Our proposed LND-GNN inherits the advantages of the above two types of models while adaptively maintaining the disagreement with label-inconsistent neighbors, resulting in better performance of node classification.

E. Robustness to Adversarial Attacks

To evaluate the robustness of the proposed LND-GNN model against adversarial attacks on graph topology, we further test the performance on three citation graphs attacked by MetaAttack [20]. Technically, MetaAttack perturbs the discrete graph structure by connecting nodes with different labels. Practically, we adopt the MetaAttack implemented in DeepRobust [40], a PyTorch library for adversarial attacks, and set the perturbation rate in $\{5\%, 10\%, 20\%\}$. In addition, we choose GCN as the surrogate model, as well as GCN-Jaccard [41], GCN-SVD [42], and ElasticGNN [21] as the baselines. The results are shown in Table VI.

Locally adaptive smoothing based on the Laplacian node disagreement makes our model more robust to adversarial perturbations on the graph topology. Specifically, initial features impose more trustworthy constraints on the node disagreement in the node embeddings, which enables our LND-GNN model to perform more tolerance to smoothing operations on perturbed edges. As shown in Table VI, our LND-GNN model achieves state-of-the-art defensive performance in the nine results. Furthermore, we observe that ElasticGNN and our model outperform models tailored for adversarial attacks, namely GCN-Jaccard and GCN-SVD, which demonstrates the ability of locally adaptive smoothing for defending against disturbances.

TABLE VI
PERFORMANCE OF NODE CLASSIFICATION UNDER ADVERSARIAL ATTACKS

Dataset	Ptb Rate	GCN	GCN-Jaccard	GCN-SVD	ElasticGNN	LND-GNN
Cora	5%	73.2 ± 0.8	79.2 ± 0.9	71.0 ± 0.6	82.2 ± 0.7	**82.5 ± 0.4**
	10%	62.9 ± 0.6	75.7 ± 0.7	69.5 ± 0.9	76.9 ± 1.8	**80.3 ± 0.9**
	20%	44.1 ± 1.0	64.2 ± 1.4	65.7 ± 1.2	54.1 ± 2.8	**75.5 ± 1.4**
Citeseer	5%	66.9 ± 0.5	70.3 ± 1.4	70.5 ± 0.7	72.5 ± 0.6	**75.6 ± 0.3**
	10%	60.7 ± 0.7	69.3 ± 1.1	69.6 ± 0.4	71.5 ± 0.5	**74.6 ± 0.3**
	20%	45.8 ± 0.8	60.1 ± 1.7	65.3 ± 0.6	62.4 ± 0.7	**71.7 ± 0.7**
Pubmed	5%	83.0 ± 0.1	83.2 ± 0.1	84.4 ± 0.1	86.3 ± 0.1	**87.4 ± 0.1**
	10%	81.1 ± 0.2	81.2 ± 0.1	84.4 ± 0.2	86.0 ± 0.1	**87.0 ± 0.1**
	20%	76.5 ± 0.3	76.9 ± 0.2	84.2 ± 0.2	85.1 ± 0.1	**86.2 ± 0.1**

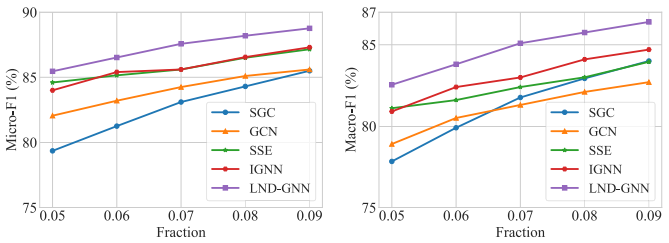

Fig. 4. Performance of the multi-label node classification on long-range dependency dataset, Amazon Co-Purchase network.

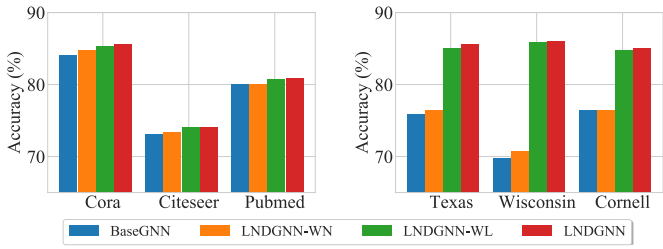

Fig. 5. Ablation study: The classification accuracy (%) results of LND-GNN and its variants.

F. Performance on Long-Range Dependency Dataset

In addition, We evaluate the capability of capturing the underlying long-range dependency in graphs without the risk of the over-smoothing issue. We conduct a multi-class node classification experiment on the Amazon Co-Purchase network dataset, a benchmark commonly used for testing long-range dependencies [14], [15], [43]. In this network, nodes denote products and edges denote co-purchases, but no node features are provided. In order to obtain a fair comparison, we use the dataset processed by [15]. As for splitting, 10% of the total nodes are chosen as the test set, while the training set is varied from 5% to 9% to be consistent with [14], [15]. Additionally, there are no node features, thus we learn a 128-dim feature vector for each node at training. We compare our proposed LND-GNN with a set of baselines, including SGC [11], GCN [4], SSE [14], and IGNN [15]. In particular, SSE and IGNN are models deliberately designed to capture long-range dependencies. The Micro/Macro-F1 performance is shown in Fig. 4, and we can observe that LND-GNN outperforms all baselines by a significant margin. This phenomenon is attributed to the capability of our proposed LND-GNN to capture the underlying long-range dependency and facilitate the performance of multi-label node classification.

G. Ablation Study

To evaluate the effectiveness of considering Laplacian node disagreement, we compare LND-GNN with the following three variants: BaseGNN (without considering Laplacian node disagreement in message passing scheme and loss constraint), LNDGNN-WN (without considering Laplacian node disagreement in message passing scheme), and LNDGNN-WL (without considering loss constraint). As shown in Fig. 5, we can observe that the methods with Laplacian node disagreement, i.e., LNDGNN and LNDGNN-WL have achieved significantly better results than those methods without Laplacian node disagreement, i.e., LNDGNN-WN and BaseGNN, which demonstrates the effectiveness of Laplacian node disagreement. Specifically, on three heterophily graphs, namely Texas, Wisconsin, and Cornell, whose disagreement ratios r are 0.81, 0.29, and 0.11, these methods with Laplacian node disagreement show more dramatic performance improvements. It confirms our previous inference that locally adaptive smoothing performs better on graphs with a high disagreement ratio r. Besides, we can observe that LNDGNN outperforms LNDGNN-WL on all datasets, which validates that our proposed loss constraint in LNDGNN can effectively enhance the embedding learning process.

H. Node Disagreement Study

Measuring the node disagreement of output node embeddings facilitates a more comprehensive assessment of the model's capability in locally adaptive smoothing. With the metrics proposed in Section IV-A, we evaluate the node disagreement in the node embeddings generated by our proposed LND-GNN model and vanilla GCN. As shown in Fig. 6, we plot the distributions of intra-class and inter-class node disagreement in node embeddings. We can observe that LND-GNN learns a more significant level of inter-class node disagreement than GCN, especially in graphs with heterophily, as shown in Fig. 6(d), (e), and (f). This phenomenon is attributed to the capability that our proposed locally adaptive smoothing can smooth intra-class node pairs while preserving the disagreement of inter-class node pairs. Locally adaptive smoothing based on the Laplacian node disagreement can spread the more trustworthy node disagreement in the initial features to the embeddings, thus leading to more significant node disagreement in the embeddings on the inter-class node pairs. Moreover, the numerical values of the node disagreement

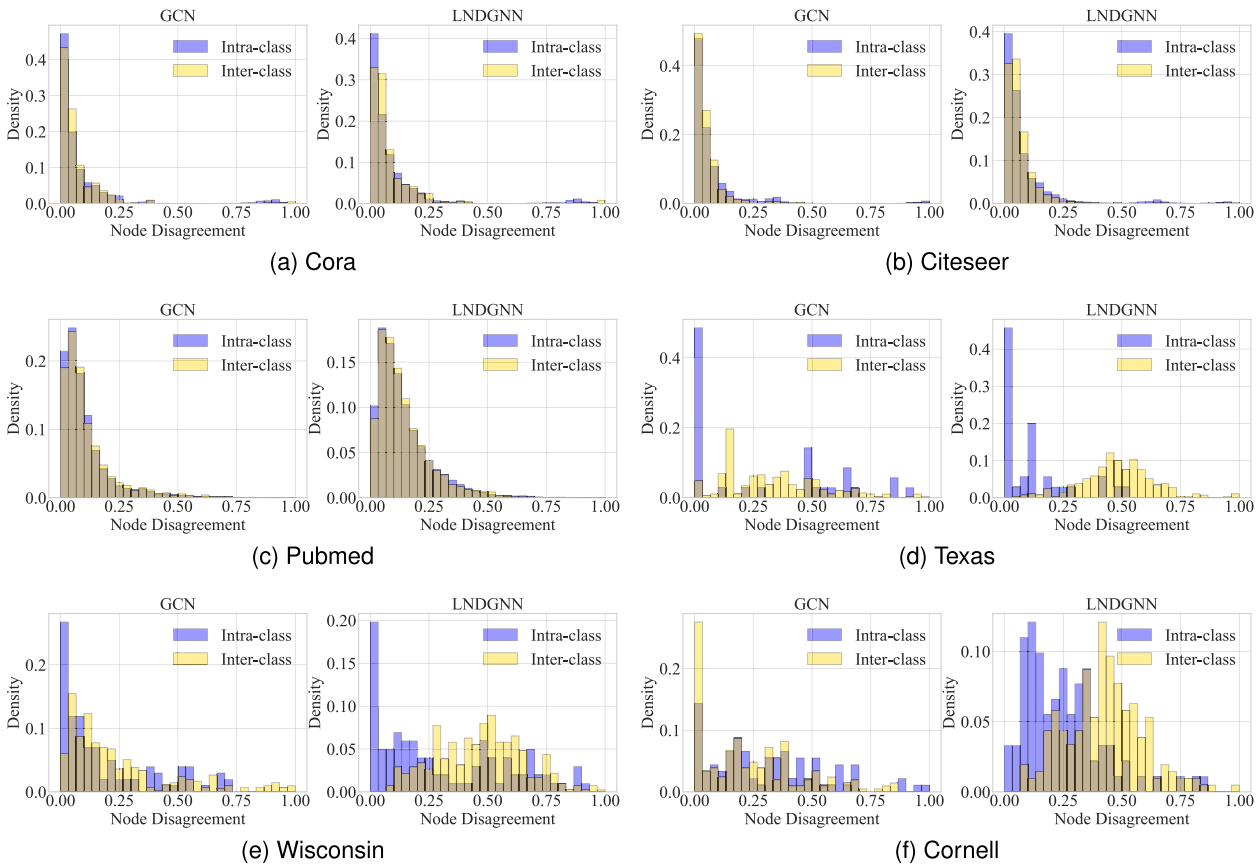

Fig. 6. Node disagreement study: Distributions of intra-class and inter-class node disagreement in node embeddings of GCN and LND-GNN.

TABLE VII
RESULTS OF NODE DISAGREEMENT RATIOS

Dis.ratio	Cora	Citeseer	Pubmed	Texas	Wisconsin	Cornell
Initial Features	0.04	0.06	0.03	0.81	0.29	0.11
GCN	-0.21	-0.28	0.02	0.18	0.34	-0.25
LND-GNN	0.06	0.07	0.07	3.19	0.52	0.69

ratios are shown in Table VII. We can observe that GCN, which employs globally uniform smoothing, generates embeddings with smaller disagreement ratios than initial features on 5 of 6 graphs. This phenomenon is consistent with the theoretical trend that over-smoothing makes nodes indistinguishable. However, the node embeddings generated by our LND-GNN maintain relatively significant inter-class node disagreements, obtaining larger node disagreement ratios under the trend that node disagreement decreases due to the smoothing operation. This phenomenon is attributed to the enhanced locally adaptive smoothing in that the smoothing intensity of inter-class nodes is suppressed as much as possible to obtain more significant inter-class node disagreements and larger node disagreement ratios.

I. Sensitivity Analysis of Hyperparameters

In this section, we study the effect on the classification accuracy of three hyperparameters of our proposed LND-GNN, namely the trade-off parameters α for global constraint, β for local constraint, and γ for disagreement constraint.

First, we study the effect of hyperparameters α for global constraint and β for local constraint, and take the values of α and β both in range $[0, 0.05, 0.1, 0.15, 0.2]$ for three homophily datasets and [0.3,0.4,0.5,0.6,0.7] for three heterophily datasets. The classification accuracy results are plotted in Fig. 7. Specifically, if we set α and β as 0, the global constraint and local constraint will not be considered in the proposed LND-GNN model. According to the results in Fig. 7, we can observe a significant divergence in the effect of trade-off parameters on homophily and heterophily graphs. For three homophily graphs, namely Cora, Citeseer, and Pubmed, setting the trade-off parameter α for global constraint as 0 is not enough while setting it as 0.2 is overflowing, and the best results are all obtained when setting $\alpha = 0.05$ on three datasets. The trade-off parameter β also exhibits a similar effect and the best results are obtained when setting $\beta = 0.1$ on three homophily datasets. However, for heterophily datasets, namely Texas, Wisconsin, and Cornell, setting α and β as a lower value of 0.3 is not enough and larger values of α and β are helpful. The best results are obtained when setting $\alpha = 0.4$ on Wisconsin and $\alpha = 0.6$ on the other two datasets, and setting $\beta = 0.6$ on all three datasets. Overall, the optimal values of α and β are around 0.05 to 0.1 for homophily datasets and 0.4 to 0.6 for heterophily datasets respectively. For heterophily graphs, where neighbors tend to be from different

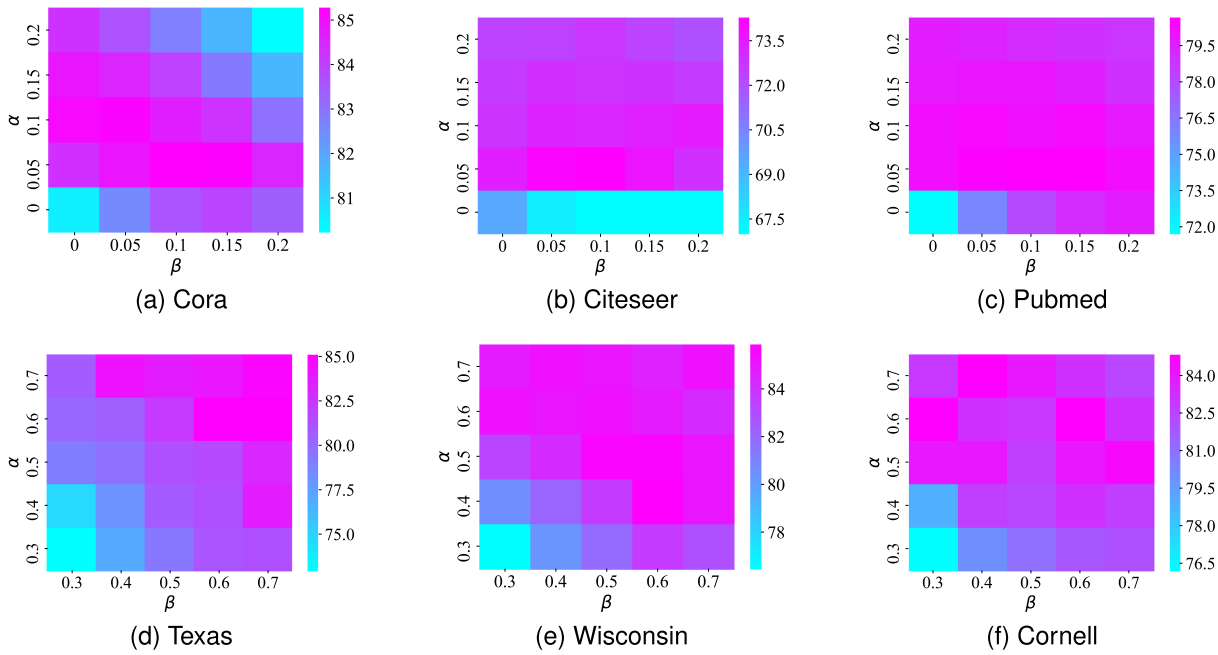

Fig. 7. Parameter analysis: The classification accuracy (%) results of LND-GNN with varying trade-off parameters α and β used to tune the importance of the global and local constraint.

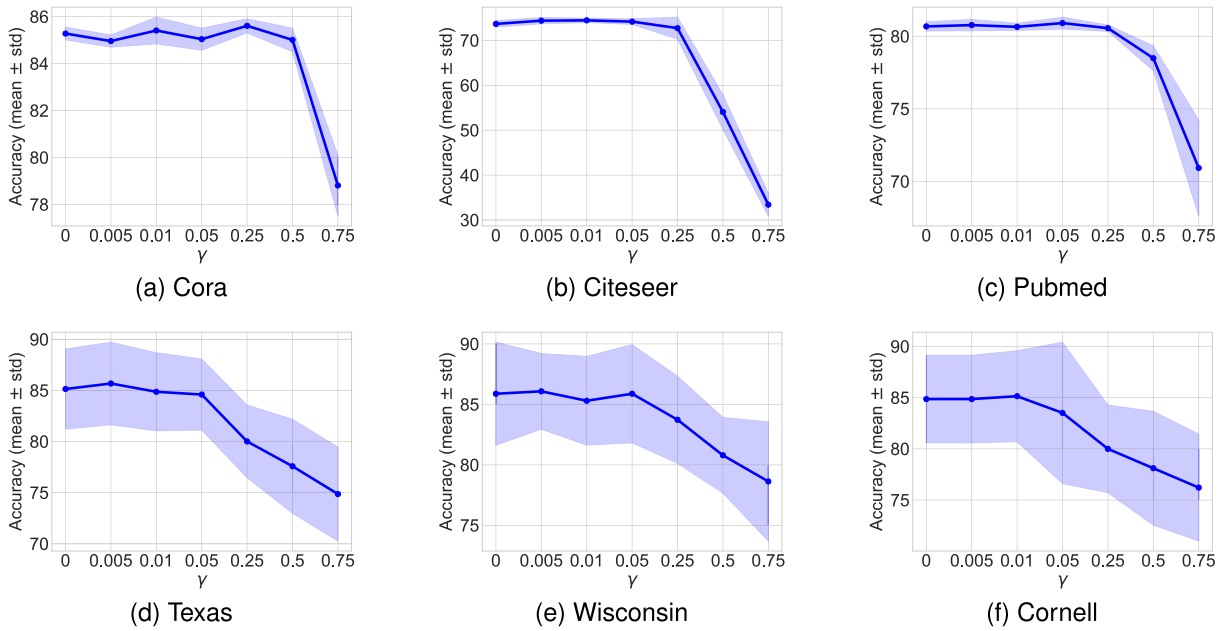

Fig. 8. Parameter analysis: The classification accuracy (%) results of LND-GNN with varying trade-off parameters γ used to tune the importance of the label disagreement constraint.

classes, tighter constraints on global and local smoothing may enhance the performance of the proposed LND-GNN model.

Second, we study the effect of the trade-off parameter γ for the disagreement constraint and take the value in the range $[0, 0.005, 0.01, 0.05, 0.25, 0.5, 0.75]$. The classification accuracy results are presented in Fig. 8. Specifically, if we set the value as 0, the disagreement constraint will not be considered in the proposed LND-GNN model. We can observe that although LND-GNN achieves the best results when taking various γ on various datasets, they all have an improvement over the results if setting $\gamma = 0$. It demonstrates that the proposed disagreement constraint can improve the performance of the proposed LND-GNN model. If setting $\gamma > 0.25$ on homophily and $\gamma > 0.05$ on heterophily, performance declines consistently, which shows that the unsupervised disagreement constraint is only auxiliary loss and should not be set too large weights to limit the role

TABLE VIII
PROPAGATION RULES OF VANILLA AND LND-BASED SCHEMES, $\mathcal{S}$ DENOTES THE ATTENTIVE MATRIX AND $\odot$ DENOTES HADAMARD PRODUCT

Architectures	Vanilla	LND
GCN	$\mathbf{H}^{(k)} = \tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}$	$\mathbf{H}^{(k)} = (\tilde{\mathbf{A}}\mathbf{H}^{(k-1)} + \mathbf{X} + \mathbf{L}\mathbf{X})\mathbf{W}^{(k)}$
GAT	$\mathbf{H}^{(k)} = \mathcal{S} \odot \tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}$	$\mathbf{H}^{(k)} = (\mathcal{S} \odot \tilde{\mathbf{A}}\mathbf{H}^{(k-1)} + \mathbf{X} + \mathbf{L}\mathbf{X})\mathbf{W}^{(k)}$
GraphSAGE	$\mathbf{H}^{(k)} = \mathbf{H}^{(k-1)}\mathbf{W}_1^{(k)} + \mathbf{A}\mathbf{H}^{(k-1)}\mathbf{W}_2^{(k)}$	$\mathbf{H}^{(k)} = \mathbf{H}^{(k-1)}\mathbf{W}_1^{(k)} + (\mathbf{A}\mathbf{H}^{(k-1)} + \mathbf{X} + \mathbf{L}\mathbf{X})\mathbf{W}_2^{(k)}$

TABLE IX
PERFORMANCE OF NODE CLASSIFICATION ON THREE GENERALIZED ARCHITECTURES

Architectures	Cora	Citeseer	Pubmed	Texas	Wisconsin	Cornell	Actor
GCN	81.3 ± 0.8	71.1 ± 0.7	78.8 ± 0.6	59.4 ± 5.2	59.8 ± 6.9	57.0 ± 4.6	30.2 ± 0.7
LND-GCN	81.4 ± 0.5	71.3 ± 0.8	79.0 ± 0.9	84.6 ± 5.4	85.3 ± 4.1	84.1 ± 5.9	36.3 ± 0.9
GAT	83.1 ± 0.4	70.8 ± 0.5	79.1 ± 0.4	58.4 ± 4.4	55.3 ± 8.7	58.9 ± 3.3	26.3 ± 1.7
LND-GAT	83.9 ± 0.5	71.0 ± 0.8	79.5 ± 0.5	78.1 ± 3.9	80.0 ± 3.1	74.9 ± 5.4	34.2 ± 1.0
GraphSAGE	82.3 ± 0.5	71.8 ± 0.7	78.5 ± 0.4	82.4 ± 6.1	81.2 ± 5.6	75.9 ± 5.0	34.3 ± 1.0
LND-SAGE	82.4 ± 0.6	72.0 ± 0.8	79.0 ± 0.6	84.6 ± 5.8	85.3 ± 3.6	80.3 ± 5.9	36.0 ± 1.3
LND-GNN	85.6 ± 0.3	74.5 ± 0.5	80.9 ± 0.4	85.7 ± 4.0	86.1 ± 3.1	85.1 ± 4.4	36.6 ± 0.9

of the supervised cross-entropy loss. Overall, the disagreement constraint benefits the performance of the node classification within an effective interval. However, outside of this regular interval, the joint performance significantly decreases.

J. Generalization of Architectural Configuration

Our proposed LND-GNN adopts an architecture that decouples the transformation operation from the aggregation operation, allowing for a deeper model to be built without negatively affecting performance. To further characterize the message passing scheme based on the Laplacian node disagreement, we generalize it to three classical GNN architectures: GCN, GAT, and GraphSAGE. Technically, these three architectures represent the classical spectral graph convolution, the attention mechanism, and ego- and neighbor-embedding separation, respectively. The propagation rules of Vanilla and LND-based schemes are summarized in Table VIII. We then verify the performance of node classification based on three generalized architectures. The results are shown in Table IX. We can observe that the LND-based schemes achieve improvements over the vanilla architectures on all graphs, especially on heterophily graphs, which proves the validity of the Laplace node disagreement. Additionally, the iterative transformations limit the depth of the model, so the improvements of LND-based schemes on homophily graphs are relatively limited, and the performances of the three architectures are slightly poorer than those of LND-GNN.

VI. CONCLUSION

This article aims to enhance the locally adaptive smoothing of GNNs via the initial features. Our motivation is that those features can present more trustworthy constraints to node disagreement in locally adaptive smoothing, which is formulated as Laplacian node disagreement. Based on the proposed graph signal denoising, a new message passing scheme is derived and implemented with GNNs in the end-to-end semi-supervised node classification task, where Laplacian node disagreement is applied to the overall loss of the task as an auxiliary constraint. Extensive experiments on graphs with homophily and heterophily, adversarially attacked graphs, and a long-range dependency dataset, show that the proposed model can achieve superior performance against a comprehensive suite of baselines.

REFERENCES

[1] M. E. Newman, "Finding and evaluating community structurein networks," *Phys. Rev. E*, vol. 69, no. 26113, pp. 1–16, 2004.

[2] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," in *Proc. Int. Conf. Learn. Representations*, 2020.

[3] 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, pp. 93–93, 2008.

[4] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[5] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[6] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5171–5181.

[7] L. Cai, J. Li, J. Wang, and S. Ji, "Line graph neural networks for link prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5103–5113, Sep. 2022.

[8] W. Ye, O. Askarisichani, A. T. Jones, and A. K. Singh, "Learning deep graph representations via convolutional neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2268–2279, May 2022.

[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1263–1272.

[10] H. Nt and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," 2019, *arXiv: 1905.09550*.

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

[12] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1202–1211.

[13] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3538–3545.

[14] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 1106–1114.

[15] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. El Ghaoui, "Implicit graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 11 984–11 995.

[16] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7793–7804.

[17] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2020.
[18] E. Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *Proc. Int. Conf. Learn. Representations*, 2021.
[19] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2847–2856.
[20] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *Proc. Int. Conf. Learn. Representations*, 2019.
[21] X. Liu et al., "Elastic graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 6837–6849.
[22] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 338–348.
[23] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3950–3957.
[24] C. Cai and Y. Wang, "A note on over-smoothing for graph neural networks," 2020, *arXiv: 2006.13318*.
[25] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *Proc. Int. Conf. Learn. Representations*, 2020.
[26] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 5453–5462.
[27] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *Proc. Int. Conf. Learn. Representations*, 2019.
[28] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1725–1735.
[29] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[30] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 422, 1999.
[31] Y. Hou et al., "Measuring and improving the use of graph information in graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020.
[32] H. Wang and J. Leskovec, "Unifying graph convolutional neural networks and label propagation," 2020, *arXiv: 2002.06755*.
[33] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 912–919.
[34] J. Calder, B. Cook, M. Thorpe, and D. Slepcev, "Poisson learning: Graph based semi-supervised learning at very low label rates," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1306–1316.
[35] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH Papers*, 2003, pp. 313–318.
[36] W. Hu et al., "Open graph benchmark: Datasets for machine learning on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 22 118–22 133.
[37] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting semi-supervised learning with graph embeddings," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2016, pp. 40–48.
[38] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 807–816.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[40] Y. Li, W. Jin, H. Xu, and J. Tang, "DeepRobust: A PyTorch library for adversarial attacks and defenses," 2020, *arXiv: 2005.06149*.
[41] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu, "Adversarial examples for graph data: Deep insights into attack and defense," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4816–4823.
[42] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, "All you need is low (rank) defending against adversarial attacks on graphs," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 169–177.
[43] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, 2015.

Yu Wang received the BS degree in network security and the MS and PhD degrees in computer science and technology from Jilin University, Changchun, China, in 2015, 2018, and 2023, respectively. His research interests include graph representation learning and hyperbolic geometry.

Liang Hu received the MS and PhD degrees in computer science from Jilin University, Changchun, China, in 1993 and 1999, respectively. He is currently a professor and a doctoral supervisor with the College of Computer Science and Technology, Jilin University. He is supported by the Hundred-Thousand-Ten Thousand Project of China. His research interests include feature selection, multilabel learning, and information theory. He is a member of the China Computer Federation.

Xiaofeng Cao received the PhD degree from Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. He is currently an associate professor with the School of Artificial Intelligence, Jilin University, China and leading a Machine Perceptron Research Group with more than 15 PhD and master's students. He has published more than 10 technical papers in top tier journals and conferences, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, CVPR, IJCAI. His research interests include PAC learning theory, agnostic learning algorithm, generalization analysis, and hyperbolic geometry.

Yi Chang (Senior Member, IEEE) is the dean of the School of Artificial Intelligence, Jilin University, Changchun, China. He was elected as a Chinese National Distinguished professor, in 2017 and an ACM distinguished scientist, in 2018. Before joining academia, he was the technical vice president with Huawei Research America, and the research director with Yahoo Labs. He is the author of two books and more than 100 papers in top conferences or journals. His research interests include information retrieval, data mining, machine learning, natural language processing, and artificial intelligence. He won the Best Paper Award on KDD'2016 and WSDM'2016. He has served as a Conference general chair for WSDM'2018 and SIGIR'2020.

Ivor W. Tsang (Fellow, IEEE) is professor of Artificial Intelligence, with University of Technology Sydney. He is also the research director of the Australian Artificial Intelligence Institute. In 2019, his paper titled "Towards ultrahigh dimensional feature selection for Big Data" received the International Consortium of Chinese Mathematicians Best Paper Award. In 2020, he was recognized as the AI 2000 AAAI/IJCAI Most Influential Scholar in Australia for his outstanding contributions to the field of Artificial Intelligence between 2009 and 2019. His works on transfer learning granted him the Best Student Paper Award at International Conference on Computer Vision and Pattern Recognition 2010 and the 2014 IEEE Transactions on Multimedia Prize Paper Award. In addition, he had received the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2007. He serves as a senior area chair for Neural Information Processing Systems and area chair for International Conference on Machine Learning, and the editorial board for *Journal Machine Learning Research*, *Machine Learning*, *Journal of Artificial Intelligence Research*, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*.