Label Propagation via Teaching-to-Learn and Learning-to-Teach

Chen Gong, Dacheng Tao, *Fellow, IEEE*, Wei Liu, *Member, IEEE*, Liu Liu, and Jie Yang

***Abstract*—How to propagate label information from labeled examples to unlabeled examples over a graph has been intensively studied for a long time. Existing graph-based propagation algorithms usually treat unlabeled examples equally, and transmit seed labels to the unlabeled examples that are connected to the labeled examples in a neighborhood graph. However, such a popular propagation scheme is very likely to yield inaccurate propagation, because it falls short of tackling ambiguous but critical data points (e.g., outliers). To this end, this paper treats the unlabeled examples in different levels of difficulties by assessing their reliability and discriminability, and explicitly optimizes the propagation quality by manipulating the propagation sequence to move from simple to difficult examples. In particular, we propose a novel iterative label propagation algorithm in which each propagation alternates between two paradigms, teaching-to-learn and learning-to-teach (TLLT). In the teaching-to-learn step, the learner conducts the propagation on the simplest unlabeled examples designated by the teacher. In the learning-to-teach step, the teacher incorporates the learner's feedback to adjust the choice of the subsequent simplest examples. The proposed TLLT strategy critically improves the accuracy of label propagation, making our algorithm substantially robust to the values of tuning parameters, such as the Gaussian kernel width used in graph construction. The merits of our algorithm are theoretically justified and empirically demonstrated through experiments performed on both synthetic and real-world data sets.**

***Index Terms*—Label propagation, machine teaching, semisupervised learning.**

I. Introduction

LABEL propagation has been intensively exploited in semisupervised learning [1], which aims to classify a massive number of unlabeled examples in the presence of a few labeled examples. Recent years have witnessed the widespread applications of label propagation in various areas, such as object tracking [2], saliency detection [3], social network analysis [4], and so on.

The notion of label propagation was introduced in [5], which proposed to iteratively propagate class labels on a weighted graph by executing random walks with clamping operations. Similarly, [6] and [7] are also random walk-based propagation algorithms. Unlike [5]–[7], which worked on asymmetric normalized graph Laplacians, Zhou and Bousquet [8] deployed a symmetric normalized graph Laplacian to implement propagation. In contrast to the aforementioned methods that used graphs with pairwise edges, Wang *et al.* [9] introduced the multiple-wise edge graph, and predicted the label of an example according to its neighbors in a linear way. Considering that the fixed adjacency (or affinity) matrix of a graph cannot always faithfully reflect the similarities between examples during propagation, Wang *et al.* [10] developed dynamic label propagation (DLP) to update the edge weights dynamically by fusing available multilabel and multiclass information. Recently, some researchers adapted the traditional label propagation methods to large-scale scenarios via either efficient graph construction [11] or efficient labeling [12]. Other representative works related to label propagation include [13]–[16].

Although the existing label propagation algorithms obtained encouraging results to some extent, they may become fragile under certain circumstances. For example, the bridge points located across different classes, and the outliers that incur abnormal distances from the normal samples of their classes are very likely to mislead the propagation and result in error-prone classifications. The reason for this is that the label propagation yielded by conventional methods is completely governed by the adjacency relationships between given examples, including labeled and unlabeled ones, in which the seed labels are blindly diffused to the unlabeled neighbors without considering the difficulty or risk of propagation. Consequently, the mutual label transmission between different classes will probably occur if the above referred ambiguous points are incorrectly activated to receive the propagated labels.

Based on this consideration, we propose a novel propagation scheme, dubbed teaching-to-learn and learning-to-teach (TLLT), to explicitly manipulate the propagation sequence, so that the unlabeled examples are logically activated from simple to difficult. The framework of our TLLT is shown in Fig. 1. An undirected weighted graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is first built [see Fig. 1(a)], where $\mathcal{V}$ is the node set representing all the examples and $\mathcal{E}$ is the edge set encoding the similarities between these nodes. In the teaching-to-learn step, a teaching

Manuscript received April 6, 2015; revised November 25, 2015 and December 28, 2015; accepted December 31, 2015. Date of publication April 5, 2016; date of current version May 15, 2017. This work was supported in part by the 973 plan of China under Grant 2015CB856004, in part by the National Natural Science Foundation of China under Grant 61572315, and in part by the Australian Research Council under Grant FT-130101457 and Grant DP-140102164. *(Corresponding authors: D. Tao and J. Yang.)*

C. Gong is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology Sydney, 81 Broadway Street, Ultimo, NSW 2007, Australia (e-mail: goodgongchen@sjtu.edu.cn).

D. Tao and L. Liu are with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology Sydney, 81 Broadway Street, Ultimo, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au; liu.liu-2@student.uts.edu.au).

W. Liu is with Didi Research, Beijing 100085, China (e-mail: weiliu@didichuxing.com).

J. Yang is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jieyang@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2514360

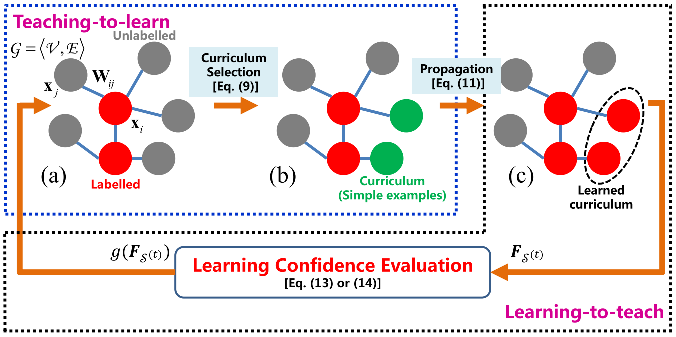
Teaching-to-learn
$\mathcal{G}=\langle\mathcal{V},\mathcal{E}\rangle$
Unlabelled
$\mathbf{x}_j$
$\mathbf{W}_{ij}$
$\mathbf{x}_i$
(a)
Labelled
Curriculum Selection [Eq. (9)]
(b)
Curriculum (Simple examples)
Propagation [Eq. (11)]
(c)
Learned curriculum
$g(\boldsymbol{F}_{\mathcal{S}^{(t)}})$
Learning Confidence Evaluation [Eq. (13) or (14)]
$\boldsymbol{F}_{\mathcal{S}^{(t)}}$
Learning-to-teach

Fig. 1. TLLT framework for label propagation. The labeled examples, unlabeled examples, and curriculum are represented by red, gray, and green balls, respectively. (a) Established graph $\mathcal{G}$, in which the examples/nodes are represented by balls and the edges are denoted by blue lines. (b) Selection of curriculum examples, where the green balls are considered as simple. (c) Selected examples in (b) are propagated by the learner. The steps of Teaching-to-Learn and Leaning-to-Teach are marked with blue and black dashed boxes.

model that serves as a teacher is established to select the simplest examples [i.e., a curriculum; see Fig. 1(b) (green balls)] from the pool of unlabeled examples (gray balls) for the current propagation. This selection is performed by solving an optimization problem that integrates the reliability and discriminability of each unlabeled example. In the learning-to-teach step, a learner activates the simplest examples to conduct label propagation using the classical method presented in [5] [see Fig. 1(c)], and meanwhile delivers its learning confidence to the teacher in order to assist the teacher in deciding the subsequent simplest examples. Such a two-step procedure iterates until all the unlabeled examples are properly handled. As a result of the interactions between the teacher and the learner, the originally difficult (i.e., ambiguous) examples are handled at a late time, so that they can be reliably labeled via leveraging the previously learned knowledge.

The argument of learning from simple to difficult levels has been acknowledged in the human cognitive domain [17], [18], and also gradually applied to advance the existing machine learning algorithms in recent years. Bengio *et al.* [19] proposed curriculum learning, which treats available examples as curriculums with different levels of difficulties in running a stepwise learner. Kumar *et al.* [20] adaptively decided which and how many examples are taken as curriculum according to the learner's ability, and termed their algorithm self-paced learning. By introducing the antigroup-sparsity term, Jiang *et al.* [21] picked up curriculums that are not only simple but also diverse. Jiang *et al.* [22] also combined curriculum learning with self-paced learning so that the proposed model can exploit both the estimation of example difficulty before learning and information about the dynamic difficulty rendered during learning.

The early works related to machine teaching mainly focus on the teaching dimension theory [23], [24]. Recently, some teaching algorithms have been developed, such as [25]–[30]. In the literature, a teacher is supposed to know the exact labels of a curriculum. However, in our case, a teacher is assumed to only know the difficulties of examples without accessing their real labels, which poses a great challenge to teaching and learning.

TABLE I

IMPORTANT NOTATIONS USED IN THIS PAPER

Notation	Description	Notation	Description
$\mathcal{L}$	the labelled set of size l	$\mathbf{\Sigma}$	kernel matrix over all the examples
$\mathcal{U}$	the unlabelled set of size u	$\mathbf{W}$	the adjacency matrix of graph
$\mathcal{B}$	the candidate set directly connected to $\mathcal{L}$, and its size is b.	$\mathbf{L}$	graph Laplacian matrix
$\mathcal{S}$	curriculum set of size s, the corresponding selection matrix is $\mathbf{S}$	$\mathbf{F}$	the label matrix with each row $\mathbf{F}_i$ representing the label vector of example $\mathbf{x}_i$

To the best of our knowledge, this paper is the first work to model label propagation as a teaching and learning framework, so that abundant unlabeled examples are activated to receive the propagated labels in a well-organized sequence. We employ the state-of-the-art label propagation algorithm [5] as the learner, because it is naturally incremental without retraining when a new curriculum comes. Empirical studies on synthetic and real-world data sets demonstrate the effectiveness of the proposed TLLT approach.

Notations: In this paper, we use the bold capital letter, bold lowercase letter, and curlicue letter to denote matrix, vector, and set, respectively. The scalar is represented by italic letters. The symbol $\mathbf{A}_{ij}$ stands for the (i, j)th element of matrix $\mathbf{A}$, and the superscript (t) associated with the variables, e.g., $\mathbf{A}^{(t)}$, is to indicate the formation of $\mathbf{A}$ under the tth propagation. Some key notations used in this paper are listed in Table I.

II. TEACHING-TO-LEARN STEP

The investigated problem is defined as follows. Suppose we have a set of $n = l + u$ examples $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$, where the first l elements constitute the labeled set $\mathcal{L}$ and the remaining u examples form the unlabeled set $\mathcal{U}$ with typically $l \ll u$. The purpose of label propagation is to iteratively propagate the label information from $\mathcal{L}$ to $\mathcal{U}$. In order to record the labels of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the label matrix is defined as $\mathbf{F} = (\mathbf{F}_1^\top, \ldots, \mathbf{F}_l^\top, \mathbf{F}_{l+1}^\top, \ldots, \mathbf{F}_n^\top)^\top$, where the ith row vector $\mathbf{F}_i \in \{1, 0\}^{1\times c}$ (c is the number of classes) satisfying $\sum_{j=1}^{c} \mathbf{F}_{ij} = 1$ denotes $\mathbf{x}_i$'s soft labels with $\mathbf{F}_{ij}$ is the probability of $\mathbf{x}_i$ belonging to the jth class $\mathcal{C}_j$. In addition, we define a set $\mathcal{S}$ to denote the curriculum that is mentioned in the introduction [e.g. the green balls in Fig. 1(b)]. For the general case, we assume that $\mathcal{S}$ contains s unlabeled examples that are selected for one propagation iteration. When one iteration of label propagation is completed, $\mathcal{L}$ and $\mathcal{U}$ are updated by $\mathcal{L} := \mathcal{L} \cup \mathcal{S}$ and $\mathcal{U} := \mathcal{U} - \mathcal{S}$, respectively.[1]

In order to quantify the graph $\mathcal{G}$ showed in Fig. 1(a), we adopt the adjacency (or affinity) matrix $\mathbf{W}$, which is

[1] Note that the notations such as l, u, s, $\mathcal{L}$, $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}_j$ are all related to the iteration number t. We drop the superscript (t) for simplicity if no confusion is incurred.

formed by $\mathbf{W}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\xi^2))$ (ξ is the Gaussian kernel width), if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge in $\mathcal{G}$, and $\mathbf{W}_{ij} = 0$ otherwise. Based upon $\mathbf{W}$, we introduce the diagonal degree matrix $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$ and graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The matrix $\mathbf{L}$ will play an important role in evaluating the difficulty levels of unlabeled examples in our model.

A. Curriculum Selection

This section introduces a teacher, which is essentially a teaching model, to decide the curriculum $\mathcal{S}$ for each iteration of propagation.

Above all, we associate each example $\mathbf{x}_i$ with a random variable y_i, which can be understood as the class label of $\mathbf{x}_i$. We also view the propagations on the graph as a Gaussian process, which is modeled as a multivariate Gaussian distribution over the random variables $\mathbf{y} = (y_1, \ldots, y_n)^\top$, that is [31]

$$p(\mathbf{y}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^\top(\mathbf{L} + \mathbf{I}/\kappa^2)\mathbf{y}\right). \tag{1}$$

In (1), $\mathbf{L} + \mathbf{I}/\kappa^2$ ($\mathbf{I}$ denotes the identity matrix with proper size in this paper and κ^2 is fixed to 100) is the regularized graph Laplacian [31]. The modeled Gaussian process has a concise form $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with its covariance matrix being $\Sigma = (\mathbf{L} + \mathbf{I}/\kappa^2)^{-1}$.

Then, we define reliability and discriminability to assess the difficulty of the examples selected in $\mathcal{S} \subseteq \mathcal{U}$.

Definition 1 (Reliability): A curriculum $\mathcal{S} \subseteq \mathcal{U}$ is reliable with respect to the labeled set $\mathcal{L}$ if the conditional entropy $H(\mathbf{y}_\mathcal{S}|\mathbf{y}_\mathcal{L})$ is small, where $\mathbf{y}_\mathcal{S}$ and $\mathbf{y}_\mathcal{L}$ represent the subvectors of $\mathbf{y}$ corresponding to the sets $\mathcal{S}$ and $\mathcal{L}$, respectively.

Definition 2 (Discriminability): A curriculum $\mathcal{S} \subseteq \mathcal{U}$ is discriminative if $\forall \mathbf{x}_i \in \mathcal{S}$, the value of

$$\min_{j' \in \{1,\ldots,c\}\backslash\{q\}} \bar{T}(\mathbf{x}_i, \mathcal{C}_{j'}) - \min_{j \in \{1,\ldots,c\}} \bar{T}(\mathbf{x}_i, \mathcal{C}_j)$$

is large, where $\bar{T}(\mathbf{x}_i, \mathcal{C}_j)$ denotes the average commute time between $\mathbf{x}_i$ and all the labeled examples in class $\mathcal{C}_j$, and $q = \arg\min_{j \in \{1,\ldots,c\}} \bar{T}(\mathbf{x}_i, \mathcal{C}_j)$.

In Definition 1, we use reliability to measure the correlation between a curriculum $\mathcal{S}$ and the labeled set $\mathcal{L}$. The curriculum examples highly correlated with the labeled set are obviously simple and reliable to classify. Such reliability is modeled as the entropy of $\mathcal{S}$ conditioned on $\mathcal{L}$, which implies that the simple examples in $\mathcal{S}$ should have small conditional entropy, since they come as no surprise to the labeled examples. In Definition 2, we introduce the discriminability to model the tendency of $\mathbf{x}_i$ belonging to certain classes. An example $\mathbf{x}_i$ is simple if it is significantly inclined to a category. Therefore, Definition 1 considers the hybrid relationship between $\mathcal{S}$ and $\mathcal{L}$, while Definition 2 associates the examples in $\mathcal{S}$ with the concrete class information, so they complement to each other in optimally selecting the simplest examples.

According to Definition 1, we aim to find a reliable set $\mathcal{S}$, such that it is most deterministic with respect to the labeled set $\mathcal{L}$, which is formulated as

$$\mathcal{S}^* = \arg\min_{\mathcal{S}\subseteq\mathcal{U}} H(\mathbf{y}_\mathcal{S}|\mathbf{y}_\mathcal{L}) := H(\mathbf{y}_{\mathcal{S}\cup\mathcal{L}}) - H(\mathbf{y}_\mathcal{L}). \tag{2}$$

Based on the property of Gaussian process [32, Theorem 8.4.1], we deduce (2) as follows:

$$\begin{aligned}\mathcal{S}^* &= \arg\min_{\mathcal{S}\subseteq\mathcal{U}} \left(\frac{s+l}{2}(1 + \ln 2\pi) + \frac{1}{2}\ln|\Sigma_{\mathcal{S}\cup\mathcal{L},\mathcal{S}\cup\mathcal{L}}|\right) \\ &\quad - \left(\frac{l}{2}(1 + \ln 2\pi) + \frac{1}{2}\ln|\Sigma_{\mathcal{L},\mathcal{L}}|\right) \\ &= \arg\min_{\mathcal{S}\subseteq\mathcal{U}} \frac{s}{2}(1 + \ln 2\pi) + \frac{1}{2}\ln\frac{|\Sigma_{\mathcal{S}\cup\mathcal{L},\mathcal{S}\cup\mathcal{L}}|}{|\Sigma_{\mathcal{L},\mathcal{L}}|}\end{aligned}$$

where $\Sigma_{\mathcal{L},\mathcal{L}}$ and $\Sigma_{\mathcal{S}\cup\mathcal{L},\mathcal{S}\cup\mathcal{L}}$ are the submatrices of Σ associated with the corresponding subscripts. By further partitioning $\Sigma_{\mathcal{S}\cup\mathcal{L},\mathcal{S}\cup\mathcal{L}} = \left(\begin{smallmatrix}\Sigma_{\mathcal{S},\mathcal{S}} & \Sigma_{\mathcal{S},\mathcal{L}} \\ \Sigma_{\mathcal{L},\mathcal{S}} & \Sigma_{\mathcal{L},\mathcal{L}}\end{smallmatrix}\right)$, we have

$$\frac{|\Sigma_{\mathcal{S}\cup\mathcal{L},\mathcal{S}\cup\mathcal{L}}|}{|\Sigma_{\mathcal{L},\mathcal{L}}|} = \frac{|\Sigma_{\mathcal{L},\mathcal{L}}|\left|\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\right|}{|\Sigma_{\mathcal{L},\mathcal{L}}|} = |\Sigma_{\mathcal{S}|\mathcal{L}}|$$

where $\Sigma_{\mathcal{S}|\mathcal{L}}$ is the covariance matrix of the conditional distribution $p(\mathbf{y}_\mathcal{S}|\mathbf{y}_\mathcal{L})$ and is naturally positive semidefinite. Therefore, minimizing $\ln|\Sigma_{\mathcal{S}|\mathcal{L}}| = \ln|\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}|$ is equivalent to minimizing $\mathrm{tr}(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}})$. Given a fixed s (we defer its determination to Section III), the most reliable curriculum $\mathcal{S}$ is then found by

$$\mathcal{S}^* = \arg\min_{\mathcal{S}\subseteq\mathcal{U}} \mathrm{tr}\big(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\big). \tag{3}$$

In Definition 2, the commute time between two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ is the expected time cost starting from $\mathbf{x}_i$, reaching $\mathbf{x}_j$, and then returning to $\mathbf{x}_i$ again, which is computed by [33][2]

$$T(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{n} h(\lambda_k)(u_{ki} - u_{kj})^2. \tag{4}$$

In (4), the values of $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of $\mathbf{L}$, and the values of $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are the associated eigenvectors. u_{ki} is the ith element of $\mathbf{u}_k$. $h(\lambda_k) = 1/\lambda_k$ if $\lambda_k \neq 0$ and $h(\lambda_k) = 0$ otherwise. Based on (4), Definition 2 calculates the average commute time $\bar{T}(\mathbf{x}_i, \mathcal{C}_j)$ between $\mathbf{x}_i$ and the examples in the jth class $\mathcal{C}_j$, which is

$$\bar{T}(\mathbf{x}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_j|}\sum_{\mathbf{x}_{i'} \in \mathcal{C}_j} T(\mathbf{x}_i, \mathbf{x}_{i'}). \tag{5}$$

Definition 2 characterizes the discriminability of an unlabeled example $\mathbf{x}_i \in \mathcal{U}$ as the average commute time difference between $\mathbf{x}_i$'s two closest classes $\mathcal{C}_{j_1}$ and $\mathcal{C}_{j_2}$, that is, $M(\mathbf{x}_i) = \bar{T}(\mathbf{x}_i, \mathcal{C}_{j_2}) - \bar{T}(\mathbf{x}_i, \mathcal{C}_{j_1})$. $\mathbf{x}_i$ is thought of as discriminative, if it is significantly inclined to a certain class, namely it has a large $M(\mathbf{x}_i)$. From Definition 2, the most discriminative curriculum that consists of s discriminative examples is equivalently found by

$$\mathcal{S}^* = \arg\min_{\mathcal{S}=\{\mathbf{x}_{i_k} \in \mathcal{U}\}_{k=1}^{s}} \sum\nolimits_{k=1}^{s} 1/M(\mathbf{x}_{i_k}). \tag{6}$$

[2] Strictly, the original commute time is $T(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{vol}(\mathcal{G})\sum_{k=1}^{n} h(\lambda_k)(u_{ki} - u_{kj})^2$, where $\mathrm{vol}(\mathcal{G})$ is a constant denoting the volume of graph $\mathcal{G}$. Here, we drop this term, since it will not influence our derivations.

Now, we propose that the simplest curriculum is not only reliable but also discriminative. Hence, we combine (3) and (6) to arrive at the following curriculum selection criterion:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}=\{\mathbf{x}_{i_k}\in\,\mathcal{U}\}_{k=1}^{s}} \mathrm{tr}(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}) + \alpha\sum_{k=1}^{s} 1/M(\mathbf{x}_{i_k}) \tag{7}$$

where $\alpha > 0$ is the tradeoff parameter.

Considering that the seed labels will be first propagated to the unlabeled examples, which are the direct neighbors of the labeled examples in $\mathcal{L}$, we collect such unlabeled examples in a set $\mathcal{B}$ with cardinality b. Since only s ($s < b$) distinct examples from $\mathcal{B}$ are needed, we introduce a binary selection matrix $\mathbf{S} \in \{1, 0\}^{b\times s}$, such that $\mathbf{S}^\top\mathbf{S} = \mathbf{I}_{s\times s}$ ($\mathbf{I}_{s\times s}$ denotes the $s \times s$ identity matrix). The element $\mathbf{S}_{ik} = 1$ means that the ith example in $\mathcal{B}$ is selected as the kth example in the curriculum $\mathcal{S}$. The orthogonality constraint $\mathbf{S}^\top\mathbf{S} = \mathbf{I}_{s\times s}$ imposed on $\mathbf{S}$ ensures that no repetitive example is included in $\mathcal{S}$.

We reformulate problem (7) in the following matrix form:

$$\begin{aligned}\mathbf{S}^* = \arg\min_{\mathbf{S}}\ & \mathrm{tr}(\mathbf{S}^\top\Sigma_{\mathcal{B},\mathcal{B}}\mathbf{S} - \mathbf{S}^\top\Sigma_{\mathcal{B},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{B}}\mathbf{S})\\ & + \alpha\mathrm{tr}(\mathbf{S}^\top\mathbf{M}\mathbf{S})\\ & \text{s.t.}\ \ \mathbf{S}\in\{1,0\}^{b\times s},\ \mathbf{S}^\top\mathbf{S} = \mathbf{I}_{s\times s}\end{aligned} \tag{8}$$

where $\mathbf{M} \in \mathbb{R}^{b\times b}$ is a diagonal matrix whose diagonal elements are $\mathbf{M}_{ii} = 1/M(\mathbf{x}_{i_k})$ for $k = 1, \ldots, b$. We notice that problem (8) falls into an integer program and is generally NP-hard. To make problem (8) tractable, we relax the discrete constraint $\mathbf{S} \in \{1, 0\}^{b\times s}$ to be a continuous nonnegative constraint $\mathbf{S} \geq \mathbf{O}$. By doing so, we pursue to solve a simpler problem

$$\begin{aligned}\mathbf{S}^* = \arg\min_{\mathbf{S}}\ & \mathrm{tr}(\mathbf{S}^\top\mathbf{R}\mathbf{S})\\ & \text{s.t.}\ \ \mathbf{S}\geq\mathbf{O},\ \mathbf{S}^\top\mathbf{S} = \mathbf{I}_{s\times s}\end{aligned} \tag{9}$$

where $\mathbf{R} = \Sigma_{\mathcal{B},\mathcal{B}} - \Sigma_{\mathcal{B},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{B}} + \alpha\mathbf{M}$ is a positive definite matrix.

B. Optimization

Note that (9) is a nonconvex optimization problem because of the orthogonality constraint. In fact, the feasible solution region is on the Stiefel manifold $\mathcal{M} = \{\mathbf{X} \in \mathbb{R}^{m_1\times m_2} : \mathbf{X}^\top\mathbf{X} = \mathbf{I}_{m_2\times m_2}\}$, which makes the conventional gradient methods easily trapped into local minima. Instead, we adopt the method of partial augmented Lagrangian multiplier (PALM) [34] to solve problem (9). In particular, only the nonnegative constraint is incorporated into the objective function of the augmented Lagrangian expression, while the orthogonality constraint is explicitly retained and imposed on the subproblem for updating $\mathbf{S}$. As such, the $\mathbf{S}$-subproblem is a Stiefel-manifold constrained optimization problem, and can be efficiently solved by the curvilinear search method [35].

Algorithm 1 Curvilinear Search Method for Solving $\mathbf{S}$-Subproblem

```
1:  Input: $\mathbf{S}$ satisfying $\mathbf{S}^\top\mathbf{S} = \mathbf{I}$, $\varepsilon = 10^{-5}$, $\tau = 10^{-3}$,
    $\vartheta = 0.2$, $\eta = 0.85$, $Q = 1$, $\nu = L(\mathbf{S})$, $iter = 0$
2:  repeat
3:    // Define searching path $\bar{\mathbf{P}}(\tau)$ and step size on the Stiefel
      manifold
4:    $\mathbf{A} = \nabla L(\mathbf{S})\cdot\mathbf{S}^\top - \mathbf{S}\cdot(\nabla L(\mathbf{S}))^\top$;
5:    repeat
6:      $\bar{\mathbf{P}}(\tau) = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right)\mathbf{S}$;
7:      $\tau := \vartheta\cdot\tau$;
8:      // Check Barzilai-Borwein condition
9:    until $L(\bar{\mathbf{P}}(\tau)) \leq \nu - \tau L'(\bar{\mathbf{P}}(0))$
10:   // Update variables
11:   $\mathbf{S} := \bar{\mathbf{P}}(\tau)$;
12:   $Q := \eta Q + 1$; $\nu := (\eta Q\nu + L(\mathbf{S}))/Q$;
13:   $iter := iter + 1$;
14: until $\|\nabla L(\mathbf{S})\|_{\mathrm{F}} < \varepsilon$
15: Output: $\mathbf{S}$
```

Updating $\mathbf{S}$: By degenerating the nonnegative constraint and preserving the orthogonality constraint in problem (9), the partial augmented Lagrangian function is

$$L(\mathbf{S}, \Lambda, \mathbf{T}, \sigma) := \mathrm{tr}(\mathbf{S}^\top\mathbf{R}\mathbf{S}) + \mathrm{tr}(\Lambda^\top(\mathbf{S}-\mathbf{T})) + \frac{\sigma}{2}\|\mathbf{S}-\mathbf{T}\|_{\mathrm{F}}^2 \tag{10}$$

where $\Lambda \in \mathbb{R}^{b\times s}$ is the Lagrangian multiplier, $\mathbf{T} \in \mathbb{R}^{b\times s}$ is a nonnegative auxiliary matrix that enforces the obtained $\mathbf{S}^*$ to be nonnegative, and $\sigma > 0$ is the penalty coefficient. Therefore, $\mathbf{S}$ is updated by minimizing (10) subject to $\mathbf{S}^\top\mathbf{S} = \mathbf{I}_{s\times s}$ using the curvilinear search method [35] (see Algorithm 1).

Fig. 2 sketches the main procedures of Algorithm 1. Starting from the point $\mathbf{S}^{(\text{iter})}$, Algorithm 1 first finds an initial searching path $-\tau\mathbf{A}\mathbf{S}^{(\text{iter})}$ in the tangent plane [i.e., $\mathcal{T}_{\mathcal{M}}(\mathbf{S}^{(\text{iter})})$] of the Stiefel manifold $\mathcal{M}$, where τ is the step size and $-\mathbf{A}\mathbf{S}^{(\text{iter})}$ is a valid descent direction. The associated matrix $\mathbf{A}$ is computed in Line 4 of Algorithm 1. Second, a retraction mapping [36] [see the red arrow in Fig. 2] is conducted by projecting $-\tau\mathbf{A}\mathbf{S}^{(\text{iter})}$ onto the manifold $\mathcal{M}$ (Line 6). As a result, a searching curve $\bar{\mathbf{P}}(\tau)$ is generated along the manifold $\mathcal{M}$. Finally, a suitable step size τ is found by Barzilai–Borwein method [37], so that the optimal $\mathbf{S}^{(\text{iter}+1)}$ can be located (Lines 7–11). In Algorithm 1, $\nabla L(\mathbf{S}) = 2\mathbf{R}\mathbf{S} + \Lambda + \sigma(\mathbf{S}-\mathbf{T})$ is the gradient of $L(\mathbf{S}, \Lambda, \mathbf{T}, \sigma)$ with respect to $\mathbf{S}$, and $L'(\bar{\mathbf{P}}(\tau)) = \mathrm{tr}(\nabla L(\mathbf{S})^\top\bar{\mathbf{P}}'(\tau))$ calculates the derivate of $L(\mathbf{S}, \Lambda, \mathbf{T}, \sigma)$ with respect to the step size τ, in which $\bar{\mathbf{P}}'(\tau) = -(\mathbf{I} + (\tau/2)\mathbf{A})^{-1}\mathbf{A}((\mathbf{S} + \bar{\mathbf{P}}(\tau)/2))$.

The retraction step in Algorithm 1 critically preserves the orthogonality constraint through the skew-symmetric matrix $\mathbf{A}$-based Cayley transformation $(\mathbf{I} + (\tau/2)\mathbf{A})^{-1}(\mathbf{I} - (\tau/2)\mathbf{A})$, which transforms $\mathbf{S}$ to $\bar{\mathbf{P}}(\tau)$ to guarantee that $\bar{\mathbf{P}}(\tau)^\top\bar{\mathbf{P}}(\tau) = \mathbf{I}$ always holds.

Updating $\mathbf{T}$: In (10), $\mathbf{T}$ is the auxiliary variable to enforce $\mathbf{S}$ nonnegative, whose update is the same as that in the

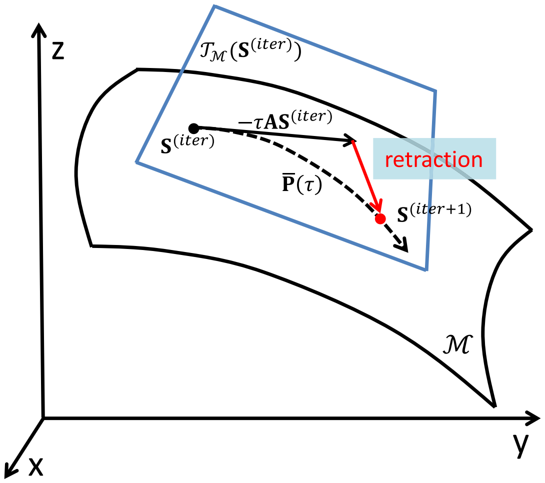

Fig. 2. Illustration of curvilinear search presented in Algorithm 1. $\mathcal{M}$ denotes the Stiefel manifold and $\mathcal{T}_{\mathcal{M}}(\mathbf{S}^{(\text{iter})})$ denotes the tangent plane at the point $\mathbf{S}^{(\text{iter})}$. First, a searching path $-\tau\mathbf{AS}^{(\text{iter})}$ in $\mathcal{T}_{\mathcal{M}}(\mathbf{S}^{(\text{iter})})$ is computed, in which $-\mathbf{AS}^{(\text{iter})}$ is a valid descent direction. Second, a retraction mapping (see the red arrow) is conducted by projecting $-\tau\mathbf{AS}^{(\text{iter})}$ onto the Stiefel manifold $\mathcal{M}$, which guarantees that the projected searching curve $\bar{\mathbf{P}}(\tau)$ for $\mathbf{S}^{(\text{iter}+1)}$ is always on $\mathcal{M}$. After finding a suitable step size τ, we may locate the feasible $\mathbf{S}^{(\text{iter}+1)}$ on the manifold.

Algorithm 2 PALM for Solving Problem (9)

```
Input: $\mathbf{R}$, $\mathbf{S}$ satisfying $\mathbf{S}^\top\mathbf{S} = \mathbf{I}$, $\Lambda = \mathbf{O}$, $\sigma = 1$, $\rho = 1.2$, $iter = 0$
repeat
    // Compute $\mathbf{T}$
    $\mathbf{T}_{ik} = \max(0,\ \mathbf{S}_{ik} + \Lambda_{ik}/\sigma)$;
    // Update $\mathbf{S}$ by minimizing Eq. (10) using Algorithm 1
    $\mathbf{S} := \arg\min_{\mathbf{S}^\top\mathbf{S}=\mathbf{I}_{s\times s}} \mathrm{tr}(\mathbf{S}^\top\mathbf{RS}) + \mathrm{tr}\left[\Lambda^\top(\mathbf{S}-\mathbf{T})\right] + \frac{\sigma}{2}\|\mathbf{S}-\mathbf{T}\|_F^2$;
    // Update variables
    $\Lambda_{ik} := \max(0, \Lambda_{ik}+\sigma\mathbf{S}_{ik})$; $\sigma := \min(\rho\sigma, 10^{10})$; $iter := iter + 1$;
until Convergence
Output: $\mathbf{S}^*$
```

traditional augmented Lagrangian multiplier (ALM) method, namely $\mathbf{T}_{ik} = \max(0,\ \mathbf{S}_{ik} + \Lambda_{ik}/\sigma)$.

We summarize the complete optimization procedure of PALM in Algorithm 2, by which a stationary point can be efficiently obtained. PALM inherits the merits of the conventional ALM, such as the nonnecessity for driving the penalty coefficient to infinity, and is also guaranteed to converge [38].

Note that the solution $\mathbf{S}^*$ generated by Algorithm 2 is continuous, which does not comply with the original binary constraint in problem (8). Therefore, we discretize $\mathbf{S}^*$ to binary values via a simple greedy procedure. In detail, we find the largest element in $\mathbf{S}^*$, and record its row and column; then from the unrecorded columns and rows, we search the largest element and mark it again; this procedure repeats until s elements are found. The rows of these s elements indicate the selected simplest examples to be propagated.

III. Learning-to-Teach Step

This section first introduces a learner, which is a propagation model, and then elaborates how the learning feedback is established for the subsequent teaching.

Suppose that the curriculum in the tth propagation iteration is $\mathcal{S}^{(t)}$. The learner learns (i.e., labels) the $s^{(t)}$ examples in $\mathcal{S}^{(t)}$ by propagating the labels of the labeled examples in $\mathcal{L}^{(t)}$ to $\mathcal{S}^{(t)}$. We adopt the following iterative propagation model [5]:

$$\mathbf{F}_i^{(t)} := \begin{cases} \mathbf{F}_i^{(0)}, & \mathbf{x}_i \in \mathcal{L}^{(0)} \\ \mathbf{P}_{i\cdot}\mathbf{F}^{(t-1)}, & \mathbf{x}_i \in \mathcal{S}^{(1:t-1)} \cup \mathcal{S}^{(t)} \end{cases} \qquad (11)$$

where $\mathcal{S}^{(1:t-1)}$ is the set $\mathcal{S}^{(1)} \cup \cdots \cup \mathcal{S}^{(t-1)}$ and $\mathbf{P}_{i\cdot}$ is the ith row of the transition matrix $\mathbf{P}$ calculated by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. The element $\mathbf{P}_{ij}$ shows the probability of a particle jumping from node j to i in the random walks interpretation [5], [6], [39]. Equation (11) reveals that the labels of the tth curriculum $\mathcal{S}^{(t)}$ along with the previously learned examples $\mathcal{S}^{(1:t-1)}$ will change during the propagation, while the labels of the initially labeled examples in $\mathcal{L}^{(0)}$ are clamped, as suggested in [5]. The initial state for $\mathbf{x}_i$'s label vector $\mathbf{F}_i^{(0)}$ is

$$\mathbf{F}_i^{(0)} := \begin{cases} \underbrace{(1/c, \ldots, 1/c)}_{c}, & \mathbf{x}_i \in \mathcal{U}^{(0)} \\ (0, \ldots, \underset{\substack{\downarrow \\ j\text{-th element}}}{1}, \ldots, 0), & \mathbf{x}_i \in \mathcal{C}_j \in \mathcal{L}^{(0)}. \end{cases} \qquad (12)$$

The formulations of (11) and (12) maintain the probability interpretation $\sum_{j=1}^{c} \mathbf{F}_{ij}^{(t)} = 1$ for any example $\mathbf{x}_i$ and all iterations $t = 0, 1, 2, \ldots$

After the tth propagation, the learner should deliver a learning feedback to the teacher and assist the teacher to determine the $(t+1)$th curriculum $\mathcal{S}^{(t+1)}$. If the tth learning result is correct, the teacher may assign a heavier curriculum to the learner for the next propagation. In this sense, the teacher should also learn the learner's feedback to arrange the proper $(t+1)$th curriculum, which is a learning-to-teach mechanism. However, the correctness of the propagated labels generated by the tth iteration remains unknown to the teacher, so the learning confidence is explored to blindly evaluate the tth learning performance.

To be specific, we restrict the learning confidence to the range $[0, 1]$, in which 1 is achieved if all the curriculum examples in $\mathcal{S}^{(t)}$ obtain definite label vectors, and 0 is reached if the curriculum examples are assigned similar label values over all the possible classes. For example, suppose we have $c = 3$ classes in total, then for a single example $\mathbf{x}_i$, it is well learned if it has a label vector $\mathbf{F}_i = [1, 0, 0]$, $[0, 1, 0]$, or $[0, 0, 1]$, which means that $\mathbf{x}_i$ definitely belongs to the class 1, 2, or 3, respectively. In contrast, if $\mathbf{x}_i$'s label vector is $\mathbf{F}_i = [1/3, 1/3, 1/3]$, it will be an ill-learned example because $[1/3, 1/3, 1/3]$ cannot provide any cue for determining its class. Therefore, we integrate the learning confidence of all the examples in $\mathcal{S}^{(t)}$ and define a learning evaluation function $g(\mathbf{F}_{\mathcal{S}^{(t)}}) : \mathbb{R}^{s^{(t)} \times c} \to \mathbb{R}$ to assess the tth propagation quality, based on which the number of examples $s^{(t+1)}$ for the

$(t+1)$th iteration can be adaptively decided. Here, $\mathbf{F}_{\mathcal{S}^{(t)}}$ denotes the obtained label matrix of the tth curriculum $\mathcal{S}^{(t)}$. A valid $g(\mathbf{F}_{\mathcal{S}^{(t)}})$ is formally described by Definition 3.

Definition 3 (Learning Evaluation Function): A learning evaluation function $g(\mathbf{F}_{\mathcal{S}^{(t)}}) : \mathbb{R}^{s^{(t)}\times c} \rightarrow \mathbb{R}$ assesses the tth learning confidence revealed by the label matrix $\mathbf{F}_{\mathcal{S}^{(t)}}$, which satisfies: 1) $0 \leq g(\mathbf{F}_{\mathcal{S}^{(t)}}) \leq 1$; 2) $g(\mathbf{F}_{\mathcal{S}^{(t)}}) \rightarrow 1$ if $\forall \mathbf{x}_i \in \mathcal{S}^{(t)}$, $\mathbf{F}_{ij} \rightarrow 1$ while $\mathbf{F}_{ik} \rightarrow 0$ for $k \neq j$; and $g(\mathbf{F}_{\mathcal{S}^{(t)}}) \rightarrow 0$ if $\mathbf{F}_{ij} \rightarrow 1/c$ for $i = 1, \ldots, s^{(t)}$, $j = 1, \ldots, c$.

Definition 3 suggests that a large $g(\mathbf{F}_{\mathcal{S}^{(t)}})$ can be achieved if the label vectors $\mathbf{F}_{i_k}$ $(k = 1, 2, \ldots, s^{(t)})$ in $\mathbf{F}_{\mathcal{S}^{(t)}}$ are almost binary. In contrast, the ambiguous label vectors $\mathbf{F}_{i_k}$ with all entries around $1/c$ cause $\mathbf{F}_{\mathcal{S}^{(t)}}$ to obtain a rather low confidence evaluation $g(\mathbf{F}_{\mathcal{S}^{(t)}})$. According to Definition 3, we propose two learning evaluation functions by, respectively, utilizing $\mathbf{F}_{\mathcal{S}^{(t)}}$'s norm and entropy

$$g_1(\mathbf{F}_{\mathcal{S}^{(t)}}) = \frac{2}{1+\exp\left[-\gamma_1\left(\|\mathbf{F}_{\mathcal{S}^{(t)}}\|_{\mathrm{F}}^2 - s^{(t)}/c\right)\right]} - 1 \tag{13}$$

$$\begin{aligned} g_2(\mathbf{F}_{\mathcal{S}^{(t)}}) &= \exp\left[-\gamma_2 \frac{1}{s^{(t)}} H(\mathbf{F}_{\mathcal{S}^{(t)}})\right] \\ &= \exp\left[\frac{\gamma_2}{s^{(t)}} \sum_{k=1}^{s^{(t)}} \sum_{j=1}^{c} (\mathbf{F}_{\mathcal{S}^{(t)}})_{kj} \log_c (\mathbf{F}_{\mathcal{S}^{(t)}})_{kj}\right] \end{aligned} \tag{14}$$

where γ_1 and γ_2 are the parameters controlling the learning rate. Increasing γ_1 in (13) or decreasing γ_2 in (14) will incorporate more examples into one curriculum.

It can be easily verified that both (13) and (14) satisfy the two requirements in Definition 3. For (13), we may write $g_1(\mathbf{F}_{\mathcal{S}^{(t)}})$ as $g_1(\mathbf{F}_{\mathcal{S}^{(t)}}) = 2\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}}) - 1$ where $\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}}) = 1/(1+\exp[-\gamma_1(\|\mathbf{F}_{\mathcal{S}^{(t)}}\|_{\mathrm{F}}^2 - s^{(t)}/c)])$ is a monotonically increasing logistic function with respect to $\|\mathbf{F}_{\mathcal{S}^{(t)}}\|_{\mathrm{F}}$. Therefore, $\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}})$ reaches its minimum value $1/2$ when $\|\mathbf{F}_{\mathcal{S}^{(t)}}\|_{\mathrm{F}}^2 = s^{(t)}/c$, which means that all the elements in $\mathbf{F}_{\mathcal{S}^{(t)}}$ equal to $1/c$. The value of $\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}})$ gradually approaches to 1 when $\|\mathbf{F}_{\mathcal{S}^{(t)}}\|_{\mathrm{F}}$ becomes larger, which requires that all the row vectors in $\mathbf{F}_{\mathcal{S}^{(t)}}$ are almost binary. Therefore, $\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}}) \in [1/2, 1)$ and $g_1(\mathbf{F}_{\mathcal{S}^{(t)}})$ maps $\bar{g}_1(\mathbf{F}_{\mathcal{S}^{(t)}})$ to $[0, 1)$ so that the two requirements in Definition 3 are satisfied.

For (14), it is evident that the entropy $H(\mathbf{F}_{\mathcal{S}^{(t)}}) = -\sum_k \sum_j (\mathbf{F}_{\mathcal{S}^{(t)}})_{kj} \log_c (\mathbf{F}_{\mathcal{S}^{(t)}})_{kj}$ falls into the range $[0, 1]$, where 0 is obtained when each row of $\mathbf{F}_{\mathcal{S}^{(t)}}$ is a {0,1}-binary vector with only one element 1, and 1 is attained if every element in $\mathbf{F}_{\mathcal{S}^{(t)}}$ is $1/c$. As a result, $g_2(\mathbf{F}_{\mathcal{S}^{(t)}})$ is valid as a learning evaluation function.

Based on a defined learning evaluation function, the number of examples included in the $(t+1)$th curriculum is

$$s^{(t+1)} = \lceil b^{(t+1)} \cdot g(\mathbf{F}_{\mathcal{S}^{(t)}}) \rceil \tag{15}$$

where $b^{(t+1)}$ is the size of set $\mathcal{B}^{(t+1)}$ in the $(t+1)$th iteration, $\lceil \cdot \rceil$ rounds up the element to the nearest integer, and $g(\cdot)$ can be either $g_1(\cdot)$ or $g_2(\cdot)$. Note that $g(\cdot)$ is simply set to a very small number, e.g., 0.05, for the first propagation, because no feedback is available at the beginning of the propagation process.

TLLT proceeds until all the unlabeled examples are learned, and the obtained label matrix is denoted by $\bar{\mathbf{F}}$. Then, we set $\bar{\mathbf{F}}^{(0)} := \bar{\mathbf{F}}$ and use the following iterative formula to drive the entire propagation process to the steady state [8], [9]:

$$\bar{\mathbf{F}}^{(t)} = \theta \mathbf{P} \bar{\mathbf{F}}^{(t-1)} + (1-\theta)\bar{\mathbf{F}} \tag{16}$$

where $\theta > 0$ is the weighting parameter balancing the labels propagated from other examples and $\bar{\mathbf{F}}$ that is produced by the TLLT process. We set $\theta = 0.05$ to enforce the final result to maximally preserve the labels generated by teaching and learning. By employing the Perron–Frobenius theorem [40], we take the limit of $\bar{\mathbf{F}}^{(t)}$ as follows:

$$\begin{aligned} \bar{\mathbf{F}}^* &= \lim_{t\rightarrow\infty} \bar{\mathbf{F}}^{(t)} = \lim_{t\rightarrow\infty} \; (\theta\mathbf{P})^t \bar{\mathbf{F}} + (1-\theta)\sum_{i=0}^{t-1} (\theta\mathbf{P})^i \bar{\mathbf{F}} \\ &= (1-\theta)(\mathbf{I} - \theta\mathbf{P})^{-1}\bar{\mathbf{F}}. \end{aligned} \tag{17}$$

Eventually, $\mathbf{x}_i$ is assigned to the j^*th class, such that $j^* = \arg\max_{j\in\{1,\ldots,c\}} \bar{\mathbf{F}}^*_{ij}$.

IV. Efficient Computations

The computational bottlenecks of TLLT are the calculation of pairwise commute time in (4) and the updating of $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$ in (3) for each propagation.

Note that (4) involves computing the eigenvectors of $\mathbf{L}$, which is time-consuming when the size of $\mathbf{L}$ (i.e., n) is large. Considering that $\mathbf{L}$ is positive semidefinite, we follow [41] and apply the Nyström approximation to reduce the computational burden. The merit of this method is that the eigenvectors of $\mathbf{L}$ can be efficiently computed via conducting singular value decomposition (SVD) on a matrix, which is much smaller than the previous matrix $\mathbf{L}$.

It is very inefficient if $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$ in (3) is computed from scratch in each propagation, so we develop an incremental way to update $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$ by using the matrix blockwise inversion [42].

The details for efficiently computing commute time and $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$ can be found in Appendixes A and B, respectively.

V. Complexity Analysis

Up to now, the entire TLLT framework for label propagation has been presented, and it is summarized in Algorithm 3. Before explaining its complexity, we first analyze the complexities of Algorithms 1 and 2 as they are the important components of Algorithm 3.

In Algorithm 1, the complexities for obtaining $\mathbf{A}$, inverting $\mathbf{I} + (\tau/2)\mathbf{A}$, and computing the value of objective function $L(\mathbf{S}, \Lambda, \mathbf{T}, \sigma)$ are $\mathcal{O}(b^2 s)$, $\mathcal{O}(b^3)$, and $\mathcal{O}(bs^2)$, respectively. Therefore, suppose the Lines 5–9 in Algorithm 1 are repeated T_1 times, and the Lines 2–14 are iterated T_2 times, then the complexity of Algorithm 1 is $\mathcal{O}([b^2 s + (b^3 + bs^2)T_1]T_2)$. As a result, the entire PALM outlined in Algorithm 2 takes $\mathcal{O}([b^2 s + (b^3 + bs^2)T_1]T_2 T_3)$ complexity, where T_3 is the iteration times of Lines 2–9 in Algorithm 2. Note that the established k-NN graph $\mathcal{G}$ is very sparse; therefore, the amount of examples directly linked to the labeled set (i.e., b) will not be extremely large. Consequently, the computational burden of Algorithm 2 is acceptable even though the complexity of our optimization is cubic to b.

Algorithm 3 TLLT for Label Propagation

```
1: Input: labelled set $\mathcal{L} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ with known labels $\mathbf{F}_1, \ldots, \mathbf{F}_l$, unlabelled set $\mathcal{U} = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$, parameters $\alpha$, $\xi$, $\gamma_1$ (or $\gamma_2$)
2: Pre-compute $\mathbf{W}$, $\mathbf{D}$, $\mathbf{L}$, $\Sigma^{-1}_{\mathcal{L},\mathcal{L}}$ appeared in Section II, and the pairwise commute time Eq. (4) by using the method of Appendix A;
3: repeat
4:     Generate curriculum $\mathcal{S}$ by solving Eq. (9), for which Algorithm 2 is utilized;
5:     Propagate from $\mathcal{L}$ to $\mathcal{U}$ via Eq. (11);
6:     Compute learning feedback via Eq. (13) or Eq. (14);
7:     Update $\Sigma^{-1}_{\mathcal{L},\mathcal{L}}$ by using the method of Appendix B;
       $\mathcal{L} := \mathcal{L} \cup \mathcal{S}$; $\mathcal{U} := \mathcal{U} - \mathcal{S}$;
8: until $\mathcal{U} = \varnothing$
9: Drive the entire propagation process to steady state via Eq. (17);
10: Output: The labels of original unlabelled examples $\mathbf{F}_{l+1}, \ldots, \mathbf{F}_{l+u}$
```

Next, we study the complexity of Algorithm 3. Since the complexities for computing $\mathbf{W}$, $\Sigma^{-1}_{\mathcal{L},\mathcal{L}}$, and pairwise commute time in Line 2 are $\mathcal{O}(n^2)$, $\mathcal{O}(l^3)$, and $\mathcal{O}(q^3)$ ($q = 10\% \cdot n$ as explained in Appendix A), respectively, so Line 2 takes $\mathcal{O}(n^2 + l^3 + q^3)$ complexity. In Line 4, the complexity of Algorithm 2 changes under different iterations because the involved b and s vary all the time, so we are only able to obtain a loose bound as $\mathcal{O}((u^3 + 2T_1u^3)T_2T_3)$ by using the facts that $b \le u$ and $s \le u$. For the same reason, the complexities of Lines 5–7 are upper bounded by $\mathcal{O}(n^2c)$, $\mathcal{O}(c^2u)$, and $\mathcal{O}(u^3)$, respectively. Besides, Line 9 takes $\mathcal{O}(n^2)$ complexity because $\bar{\mathbf{F}}^*$ in (17) can be solved by transforming (17) to a group of linear equations with an $n \times n$ coefficient matrix. Therefore, suppose Lines 3–8 are iterated T_4 times, the upper bound of the complexity for the entire TLLT algorithm is $\mathcal{O}(l^3 + q^3 + n^2 + [(u^3 + 2T_1u^3)T_2T_3 + n^2c + c^2u + u^3]T_4)$. This complexity is not as high as it suggests because l is usually small for semisupervised problems. The parameter q is also set to a small value in our approach. Besides, since the upper bounds of the complexity in Lines 4–7 are very loose as explained above, the practical computational complexity is much lower than the derived upper bound.

VI. Robustness Analysis

For graph-based learning algorithms, the choice of the Gaussian kernel width ξ is critical to achieving good performance. Unfortunately, tuning this parameter is usually nontrivial because a slight perturbation of ξ will lead to a big change in the model output. Several methods [39], [43] have been proposed to decide the optimal ξ via entropy minimization [39] or local reconstruction [43]. However, they are heuristic and not guaranteed to always obtain the optimal ξ. Here, we demonstrate that TLLT is very robust to the variation of ξ, which implies that ξ in our method can be easily tuned.

Theorem 4: Suppose that the adjacency matrix $\tilde{\mathbf{W}}$ of graph $\mathcal{G}$ is perturbed from $\mathbf{W}$ due to the variation of ξ, such that for some $\delta > 1, \forall i, j$, $\mathbf{W}_{ij}/\delta \le \tilde{\mathbf{W}}_{ij} \le \delta\mathbf{W}_{ij}$. The deviation of the tth propagation that result on the initial unlabeled examples $\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}}$ from the accurate $\mathbf{F}^{(t)}_{\mathcal{U}}$[3] is bounded by $\|\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}} - \mathbf{F}^{(t)}_{\mathcal{U}}\|_{\mathrm{F}} \le \mathcal{O}(\delta^2 - 1)$.

Proof: Given $\mathbf{W}_{ij}/\delta \le \tilde{\mathbf{W}}_{ij} \le \delta\mathbf{W}_{ij}$ for $\delta > 1$, the bound for the (i, j)th element in the perturbed transition matrix $\tilde{\mathbf{P}}$ is $\mathbf{P}_{ij}/\delta^2 \le \tilde{\mathbf{P}}_{ij} \le \delta^2\mathbf{P}_{ij}$. Besides, by recalling that $0 \le \mathbf{P}_{ij} \le 1$ as $\mathbf{P}$ has been row normalized, and the difference between $\mathbf{P}_{ij}$ and $\tilde{\mathbf{P}}_{ij}$ satisfies

$$|\tilde{\mathbf{P}}_{ij} - \mathbf{P}_{ij}| \le (\delta^2 - 1)\mathbf{P}_{ij} \le \delta^2 - 1. \tag{18}$$

For the ease of analysis, we rewrite the learning model (11) in a more compact form. Suppose $\mathbf{Q}_{\mathcal{S}^{(1:t)}} \in \{0, 1\}^{u^{(0)} \times u^{(0)}}$ ($\mathcal{S}^{(1:t)} = \mathcal{S}^{(1)} \cup \ldots \cup \mathcal{S}^{(t)}$ as defined in Section III) is a $\{0, 1\}$-binary diagonal matrix where the diagonal elements are set to 1, if they correspond to the examples in the set $\mathcal{S}^{(1:t)}$, then (11) can be reformulated as

$$\mathbf{F}^{(t)}_{\mathcal{U}} = \mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\cdot}\mathbf{F}^{(t-1)} + (\mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}})\mathbf{F}^{(t-1)}_{\mathcal{U}} \tag{19}$$

where $\mathbf{P}_{\mathcal{U},\cdot} = (\mathbf{P}_{\mathcal{U},\mathcal{L}}\ \mathbf{P}_{\mathcal{U},\mathcal{U}})$ denotes the rows in $\mathbf{P}$ corresponding to $\mathcal{U}$. Similarly, the perturbed $\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}}$ is

$$\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}} = \mathbf{Q}_{\mathcal{S}^{(1:t)}}\tilde{\mathbf{P}}_{\mathcal{U},\cdot}\mathbf{F}^{(t-1)} + (\mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}})\mathbf{F}^{(t-1)}_{\mathcal{U}}. \tag{20}$$

As a result, the difference between $\mathbf{F}^{(t)}_{\mathcal{U}}$ and $\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}}$ is computed by

$$\begin{aligned}\|\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}} - \mathbf{F}^{(t)}_{\mathcal{U}}\|_{\mathrm{F}} &= \|\mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\cdot} - \mathbf{P}_{\mathcal{U},\cdot})\mathbf{F}^{(t-1)}\|_{\mathrm{F}}\\ &\le \|\mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\cdot} - \mathbf{P}_{\mathcal{U},\cdot})\|_{\mathrm{F}}\|\mathbf{F}^{(t-1)}\|_{\mathrm{F}}.\end{aligned} \tag{21}$$

By employing (18), we arrive at

$$\|\mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\cdot} - \mathbf{P}_{\mathcal{U},\cdot})\|_{\mathrm{F}} \le (\delta^2 - 1)\sqrt{n\sum_{i=1}^{t} s^{(i)}}. \tag{22}$$

In addition, since the sum of every row in $\mathbf{F}^{(t-1)} \in [0, 1]^{n \times c}$ is 1, we know that

$$\|\mathbf{F}^{(t-1)}\|_{\mathrm{F}} \le \sqrt{n}. \tag{23}$$

Finally, by plugging (22) and (23) into (21) and noting that $\sum_{i=1}^{t} s^{(i)} \le u^{(0)}$, we obtain

$$\|\tilde{\mathbf{F}}^{(t)}_{\mathcal{U}} - \mathbf{F}^{(t)}_{\mathcal{U}}\|_{\mathrm{F}} \le (\delta^2 - 1)\sqrt{n\sum\nolimits_{i=1}^{t} s^{(i)}} \le (\delta^2 - 1)n\sqrt{u^{(0)}}. \tag{24}$$

Since $n\sqrt{u^{(0)}}$ is a constant, Theorem 4 is proved, which reveals that our algorithm is insensitive to the perturbation of Gaussian kernel width ξ in one propagation. □

However, one may argue that the error introduced in every propagation will accumulate and degrade the final parametric stability of TLLT. To show that the error will not be significantly amplified, the error bound between successive propagations is presented in Theorem 5.

[3] For ease of explanation, we slightly abuse the notations in this section by using $\mathcal{L}$ and $\mathcal{U}$ to represent the initial labeled set $\mathcal{L}^{(0)}$ and unlabeled set $\mathcal{U}^{(0)}$. They are not time-varying variables as previously defined. Therefore, the notation $\mathbf{F}^{(t)}_{\mathcal{U}}$ represents the labels of initial unlabeled examples produced by the tth propagation.

Theorem 5: Let $\tilde{\mathbf{F}}_{\mathcal{U}}^{(t-1)}$ be the perturbed label matrix $\mathbf{F}_{\mathcal{U}}^{(t-1)}$ generated by the $(t-1)$th propagation, which satisfies $\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t-1)} - \mathbf{F}_{\mathcal{U}}^{(t-1)}\|_{\mathrm{F}} \leq \mathcal{O}(\delta^2 - 1)$. Let $\tilde{\mathbf{P}}$ be the perturbed transition matrix $\mathbf{P}$. Then, after the tth propagation, the inaccurate output $\tilde{\mathbf{F}}_{\mathcal{U}}^{(t)}$ will deviate from its real value $\mathbf{F}_{\mathcal{U}}^{(t)}$ by $\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t)} - \mathbf{F}_{\mathcal{U}}^{(t)}\|_{\mathrm{F}} \leq \mathcal{O}(\delta^2 - 1)$.

Proof: Theorem 5 can be proved based on the following existing result.

Lemma 6 [44]: Suppose p_1 and p_2 are two uncertain variables with possible errors Δp_1 and Δp_2, then the deviation Δp_3 of their multiplication $p_3 = p_1 \cdot p_2$ satisfies $\Delta p_3 = p_1 \cdot \Delta p_2 + \Delta p_1 \cdot p_2$.

Based on Lemma 6, next we bound the error accumulation between successive propagations under the perturbed Gaussian kernel width ξ. Equation (19) can be rearranged as

$$\begin{aligned}\mathbf{F}_{\mathcal{U}}^{(t)} &= \mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\cdot}\,\mathbf{F}^{(t-1)} + (\mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}})\mathbf{F}_{\mathcal{U}}^{(t-1)}\\ &= \mathbf{Q}_{\mathcal{S}^{(1:t)}}\big(\mathbf{P}_{\mathcal{U},\mathcal{L}}\mathbf{F}_{\mathcal{L}}^{(t-1)} + \mathbf{P}_{\mathcal{U},\mathcal{U}}\mathbf{F}_{\mathcal{U}}^{(t-1)}\big) + (\mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}})\mathbf{F}_{\mathcal{U}}^{(t-1)}\\ &= \mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{L}}\mathbf{F}_{\mathcal{L}}^{(t-1)} + \big(\mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{U}} + \mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}}\big)\mathbf{F}_{\mathcal{U}}^{(t-1)}\\ &= \big(\mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{L}} \quad \mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{U}} + \mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}}\big)\begin{pmatrix}\mathbf{F}_{\mathcal{L}}^{(t-1)}\\ \mathbf{F}_{\mathcal{U}}^{(t-1)}\end{pmatrix}\\ &= \Phi^{(t)}\mathbf{F}^{(t-1)}\end{aligned}\tag{25}$$

where $\Phi^{(t)} = (\mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{L}} \ \mathbf{Q}_{\mathcal{S}^{(1:t)}}\mathbf{P}_{\mathcal{U},\mathcal{U}} + \mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}})$ and $\mathbf{F}^{(t-1)} = (\mathbf{F}_{\mathcal{L}}^{(t-1)\top} \ \mathbf{F}_{\mathcal{U}}^{(t-1)\top})^{\top}$. Therefore, by leveraging Lemma 6, we know that

$$\tilde{\mathbf{F}}_{\mathcal{U}}^{(t)} - \mathbf{F}_{\mathcal{U}}^{(t)} = (\tilde{\Phi}^{(t)} - \Phi^{(t)})\mathbf{F}^{(t-1)} + \Phi^{(t)}(\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)})\tag{26}$$

where $\tilde{\Phi}^{(t)} = \big(\mathbf{Q}_{\mathcal{S}^{(1:t)}}\tilde{\mathbf{P}}_{\mathcal{U},\mathcal{L}} \ \mathbf{Q}_{\mathcal{S}^{(1:t)}}\tilde{\mathbf{P}}_{\mathcal{U},\mathcal{U}} + \mathbf{I} - \mathbf{Q}_{\mathcal{S}^{(1:t)}}\big)$ is the imprecise $\Phi^{(t)}$ induced by $\tilde{\mathbf{P}}$. Consequently, we obtain

$$\begin{aligned}\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t)} - \mathbf{F}_{\mathcal{U}}^{(t)}\|_{\mathrm{F}} &= \|(\tilde{\Phi}^{(t)} - \Phi^{(t)})\mathbf{F}^{(t-1)} + \Phi^{(t)}(\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)})\|_{\mathrm{F}}\\ &\leq \|\tilde{\Phi}^{(t)} - \Phi^{(t)}\|_{\mathrm{F}}\|\mathbf{F}^{(t-1)}\|_{\mathrm{F}}\\ &\quad + \|\Phi^{(t)}\|_{\mathrm{F}}\|\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)}\|_{\mathrm{F}}.\end{aligned}\tag{27}$$

Next, we investigate the upper bounds of $\|\tilde{\Phi}^{(t)} - \Phi^{(t)}\|_{\mathrm{F}}$, $\|\mathbf{F}^{(t-1)}\|_{\mathrm{F}}$, $\|\Phi^{(t)}\|_{\mathrm{F}}$, and $\|\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)}\|_{\mathrm{F}}$ appeared in (27). Of these, $\|\mathbf{F}^{(t-1)}\|_{\mathrm{F}}$ has been bounded in (23).

It is also straightforward that

$$\begin{aligned}&\|\tilde{\Phi}^{(t)} - \Phi^{(t)}\|_{\mathrm{F}}\\ &\quad = \big\|\big(\mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\mathcal{L}} - \mathbf{P}_{\mathcal{U},\mathcal{L}}) \quad \mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\mathcal{U}} - \mathbf{P}_{\mathcal{U},\mathcal{U}})\big)\big\|_{\mathrm{F}}\\ &\quad = \|\mathbf{Q}_{\mathcal{S}^{(1:t)}}(\tilde{\mathbf{P}}_{\mathcal{U},\cdot} - \mathbf{P}_{\mathcal{U},\cdot})\|_{\mathrm{F}}\\ &\quad \overset{1}{\leq} (\delta^2 - 1)\sqrt{n\sum_{i=1}^{t} s^{(i)}}\\ &\quad \overset{2}{\leq} (\delta^2 - 1)\sqrt{\mathrm{nu}^{(0)}}\end{aligned}\tag{28}$$

where inequality (1) is given by (22), and inequality (2) holds because $\sum_{i=1}^{t} s^{(i)} \leq u^{(0)}$.

By further investigating the structure of the $u^{(0)} \times n$ matrix $\Phi^{(t)}$ in (25), it is easy to find that the ith row of $\Phi^{(t)}$ (i.e., $\Phi_{i\cdot}^{(t)}$) is

$$\Phi_{i\cdot}^{(t)} := \begin{cases}\mathbf{P}_{i\cdot}, & \mathbf{x}_i \in \mathcal{S}^{(1:t)}\\ (0, \ldots, \underset{i\text{-th element}}{\overset{\downarrow}{1}}, \ldots, 0), & \mathbf{x}_i \notin \mathcal{S}^{(1:t)}\end{cases}\tag{29}$$

where $\mathbf{P}_{i\cdot}$ denotes the ith row of $\mathbf{P}$. Therefore, the sum of every row in $\Phi^{(t)}$ is not larger than 1, leading to

$$\|\Phi^{(t)}\|_{\mathrm{F}} \leq \sqrt{u^{(0)}}\tag{30}$$

where we again use the fact that $0 \leq \mathbf{P}_{ij} \leq 1$.

Recalling that the labels of the original labeled examples are clamped after every propagation [see (11)], namely $\tilde{\mathbf{F}}_{\mathcal{L}}^{(t-1)} = \mathbf{F}_{\mathcal{L}}^{(t-1)}$, so the bound obtained in Theorem 4 also applies to $\|\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)}\|_{\mathrm{F}}$, which is

$$\|\tilde{\mathbf{F}}^{(t-1)} - \mathbf{F}^{(t-1)}\|_{\mathrm{F}} = \big\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t-1)} - \mathbf{F}_{\mathcal{U}}^{(t-1)}\big\|_{\mathrm{F}} \leq \mathcal{O}(\delta^2 - 1).\tag{31}$$

Because $u^{(0)}$ and n are constants for a given problem, Theorem 5 is finally proved by substituting (23), (28), (30), and (31) into (27). Theorem 5 implies that under the perturbed $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{F}}^{(t-1)}$, the error bound after the tth propagation $\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t)} - \mathbf{F}_{\mathcal{U}}^{(t)}\|_{\mathrm{F}}$ is the same as that before the tth propagation $\|\tilde{\mathbf{F}}_{\mathcal{U}}^{(t-1)} - \mathbf{F}_{\mathcal{U}}^{(t-1)}\|_{\mathrm{F}}$. Therefore, the labeling error will not be significantly accumulated when the propagations proceed. □

Considering Theorems 4 and 5 together, we conclude that a small variation of ξ will not greatly influence the performance of TLLT, so the robustness of the entire propagation algorithm is guaranteed. Accordingly, the parameter ξ used in our method can be easily tuned. An empirical demonstration of parametric insensitivity can be found in Section VII-G.

VII. Experimental Results

In this section, we compare the proposed TLLT with several representative label propagation methods on both synthetic and practical data sets. In particular, we implement TLLT with two different learning-to-teach strategies presented in (13) and (14), and term them TLLT (Norm) and TLLT (Entropy), respectively. The compared methods include Gaussian field and harmonic functions (GFHF) [5], local and global consistency (LGC) [8], graph transduction via alternating minimization (GTAM) [45], linear neighborhood propagation (LNP) [9], and DLP [10]. Note that GFHF is the learning model (i.e., learner) used by our proposed TLLT, which is not instructed by a teacher.

A. Synthetic Data

We begin by leveraging the 2-D DoubleMoon data set to visualize the propagation process of different methods. DoubleMoon consists of 640 examples, which are equally divided into two moons. This data set was contaminated by Gaussian noise with standard deviation 0.15, and each class had only one initial labeled example [see Fig. 3(a)]. The 8-NN graph with the Gaussian kernel width $\xi = 1$ is established for TLLT, GFHF, LGC, GTAM, and DLP. The parameter μ in GTAM is set to 99 according to [45]. The number of neighbors k for LNP is adjusted to 10. We set

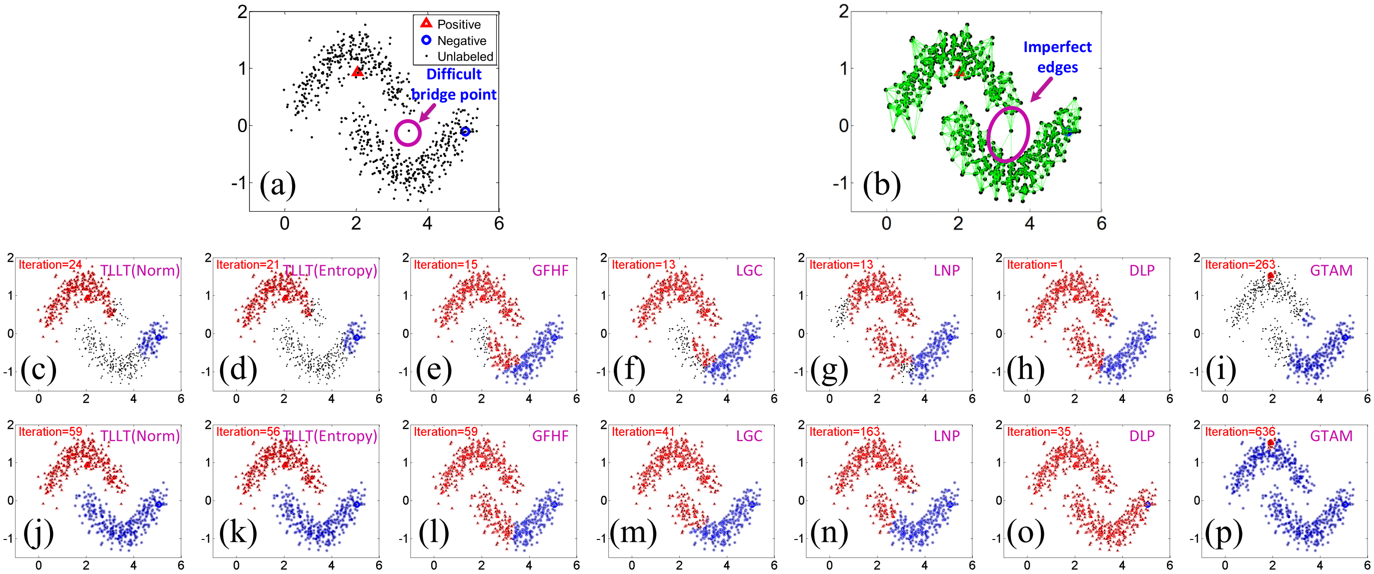

Fig. 3. Propagation process of the methods on the DoubleMoon data set. (a) Initial state with marked labeled examples and difficult bridge point. (b) Imperfect edges during graph construction caused by the bridge point in (a). These unsuitable edges pose a difficulty for all the compared methods to achieve accurate propagation. (c)–(i) show the intermediate propagations of TLLT (Norm), TLLT (Entropy), GFHF, LGC, LNP, DLP, and GTAM. (j)–(p) Compares the results achieved by all the algorithms, which reveals that only the proposed TLLT achieves perfect classification while the other methods are misled by the ambiguous bridge point.

$\gamma_1 = 1$ for TLLT (Norm) and $\gamma_2 = 2$ for TLLT (Entropy). The tradeoff parameter α in (7) is fixed to 1 throughout this paper, and we will show that the result is insensitive to the variation of this parameter in Section VII-G.

From Fig. 3(a), we observe that the distance between the two classes is very small, and that a difficult bridge point is located in the intersection region between the two moons. Therefore, the improper edges [see Fig. 3(b)] caused by the bridge point may lead to the mutual transmission of labels from different classes. As a result, previous label propagation methods, such as GFHF, LGC, LNP, DLP, and GTAM, generate unsatisfactory results, as shown in Fig. 3(l)–(p). In contrast, only our proposed TLLT [including TLLT (Norm) and TLLT (Entropy)] achieves perfect classification without any confusion [see Fig. 3(j) and (k)]. The reason for our accurate propagation can be found in Fig. 3(c) and (d), which indicate that the propagation to ambiguous bridge point is postponed due to the careful curriculum selection. On the contrary, the critical but difficult bridge examples are propagated by GFHF, LGC, LNP, DLP, and GTAM at an early stage as long as they are connected to the labeled examples [see Fig. 3(e)–(i)], resulting in the mutual label transmission between the two moons. This experiment highlights the importance of our teaching-guided label propagation.

B. UCI Benchmark Data

In this section, we compare TLLT with GFHF, LGC, GTAM, LNP, and DLP on ten UCI benchmark data sets [46], including Iris, Wine, Seeds, SPECTF, CNAE9, BreastCancer, BreastTissue, Haberman, Leaf, and Banknote. For each data set, all the algorithms are tested with different numbers of initial labeled examples (i.e., $l^{(0)}$). In order to suppress the influence of different initial labeled sets to the final performance, the accuracies are reported as the mean values of the outputs of 200 independent runs.

We established the same 5-NN graphs (i.e., $k = 5$) for TLLT, GFHF, LGC, GTAM, and DLP on all the ten data sets to achieve fair comparison. The kernel width ξ was chosen from $\{0.05, 0.5, 5, 50\}$ and was, respectively, adjusted to 0.5, 0.5, 0.5, 50, 5, 5, 5, 5, 5, and 5 on Iris, Wine, Seeds, SPECTF, CNAE9, BreastCancer, BreastTissue, Haberman, Leaf, and Banknote. For LNP, we set k to 10, 50, 30, 50, 30, 50, 20, 50, 10, and 50 on the ten data sets, since the graph required by LNP is different from other methods. Throughout the experiments of this paper, the parameter α for LGC is set to 0.99 as suggested in [8], and μ in GTAM is tuned to 99 according to [45]. The two parameters α and λ in DLP are, respectively, adjusted to 0.05 and 0.1, which are also given in [10]. The classification accuracies of all the compared methods are presented in Fig. 4.

From Fig. 4, we observe that the proposed TLLT (Norm) and TLLT (Entropy) yield better performance than other baselines in most cases. An exceptional case is that DLP generates the best result on Haberman data set. Besides, we note that GFHF generally obtains impressive performances on all the data sets. However, TLLT is able to further improve the performance of GFHF. Therefore, the well-organized learning sequence produced by our teaching and learning strategy does help to improve the propagation performance. Another notable fact is that the standard error of TLLT is very small when compared with some other baselines, such as DLP and LNP, which suggests that TLLT is not sensitive to the choice of initial labeled examples.

C. Text Categorization

To demonstrate the superiority of TLLT in dealing with practical problems, we first compare the performances of TLLT against GFHF, LGC, GTAM, LNP, and DLP in terms of text categorization. A subset of 20Newsgroups[4] data set

[4] http://qwone.com/~jason/20Newsgroups/

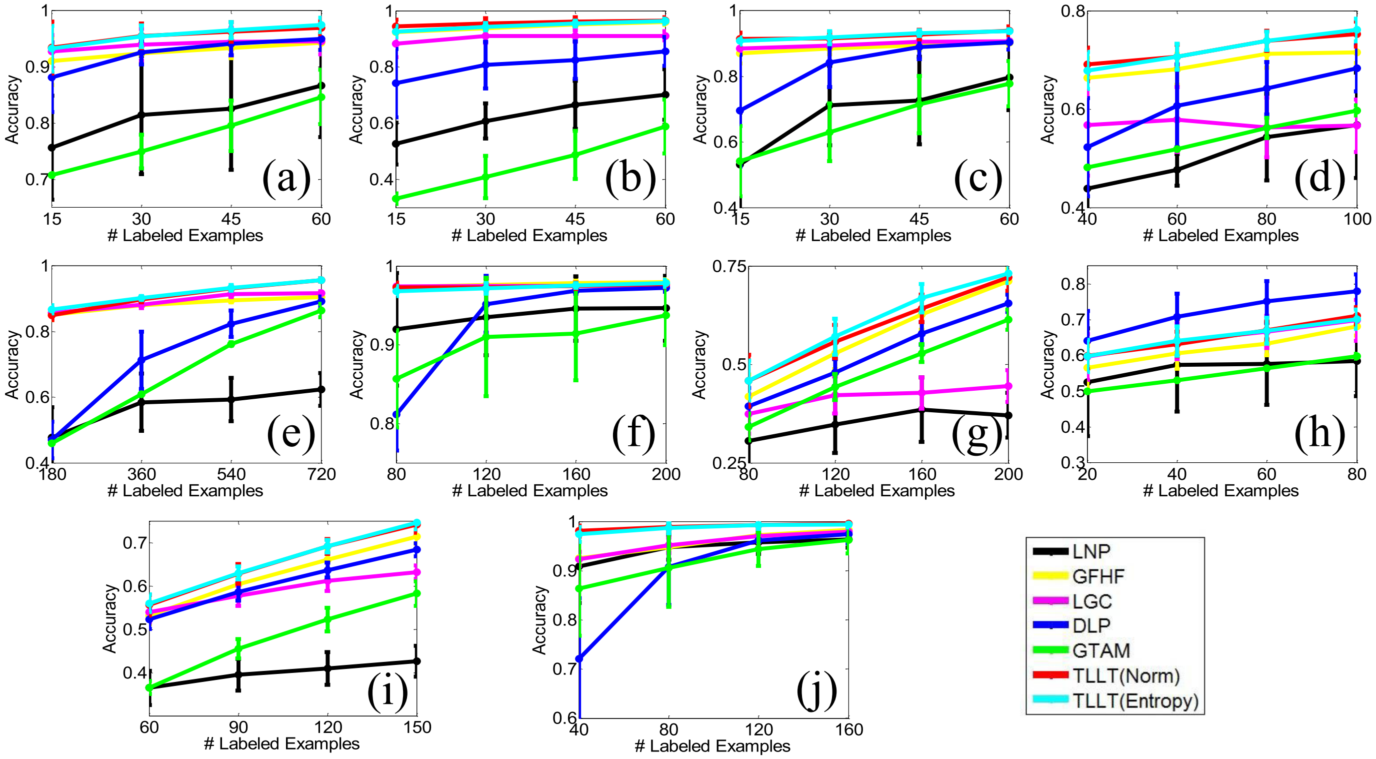

Fig. 4. Experimental results of the compared methods on ten UCI benchmark data sets. The subfigures (a)–(j) represent Iris, Wine, Seeds, SPECTF, CNAE9, BreastCancer, BreastTissue, Haberman, Leaf, and Banknote, respectively.

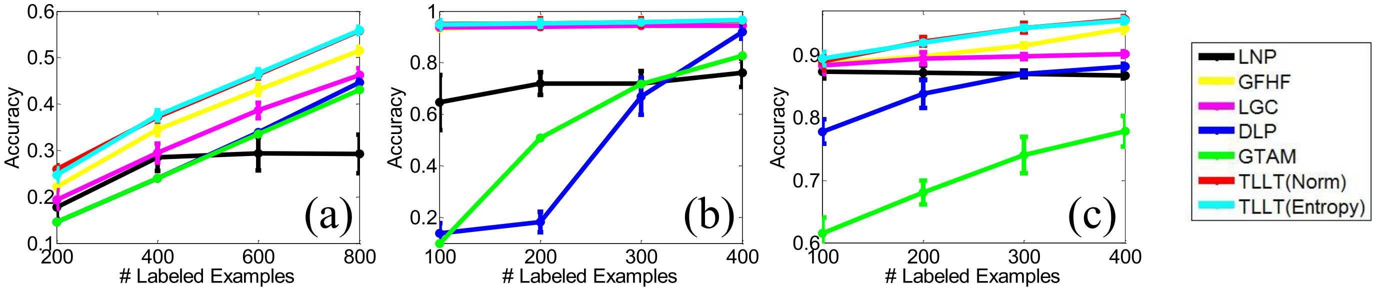

Fig. 5. Comparison of TLLT and other methods on three practical applications. (a) 20Newsgroups data set for text categorization. (b) USPS data set for handwritten digit recognition. (c) COIL20 data set for object recognition. The y-axis in each subfigure represents classification accuracy obtained by various algorithms, and the x-axis records the amount of initial labeled examples $l^{(0)}$.

with 2000 newsgroup documents is employed for our experiment. These 2000 documents are extracted from totally 20 classes, and each class has 100 examples.

The common graph was constructed for GFHF, LGC, GTAM, DLP, and TLLT, and the related parameters are $k = 10$ and $\xi = 5$. The value of k for LNP is adjusted to 50. The learning rates for TLLT (Norm) and TLLT (Entropy) are optimally tuned to $\gamma_1 = 100$ and $\gamma_2 = 0.01$, respectively, by searching the grid $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. We implement all the methods with the number of initial labeled examples $l^{(0)}$ changing from 200 to 800, and report the classification accuracy averaged over the outputs of 200 independent runs under each $l^{(0)}$.

The experimental results are presented in Fig. 5(a), from which we observe that both TLLT (Norm) and TLLT (Entropy) outperform the other competing methods under different choices of $l^{(0)}$. In particular, TLLT (Norm) and TLLT (Entropy) comparably perform on this data set, and they lead GFHF with a margin approximately 3%–4%. Besides, the standard error of TLLT is quite small with different selections of initial labeled examples, which again demonstrate that TLLT is insensitive to the choice of initial labeled examples.

D. Handwritten Digit Recognition

Handwritten digit recognition is a traditional problem in computer vision. This section compares the performances of TLLT and the baseline algorithms, including GFHF, LGC, GTAM, DLP, and LNP on handwritten digit recognition. We adopt the USPS[5] data set for comparison. In this data set, there are 9298 images of digits represented by 255-dimensional feature vectors. The digits 0–9 are considered as ten different classes. We built a ten-NN graph with kernel width $\xi = 5$ for all the methods except LNP, and the number of neighbors k for LNP is tuned to 50. Besides, the learning

[5] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

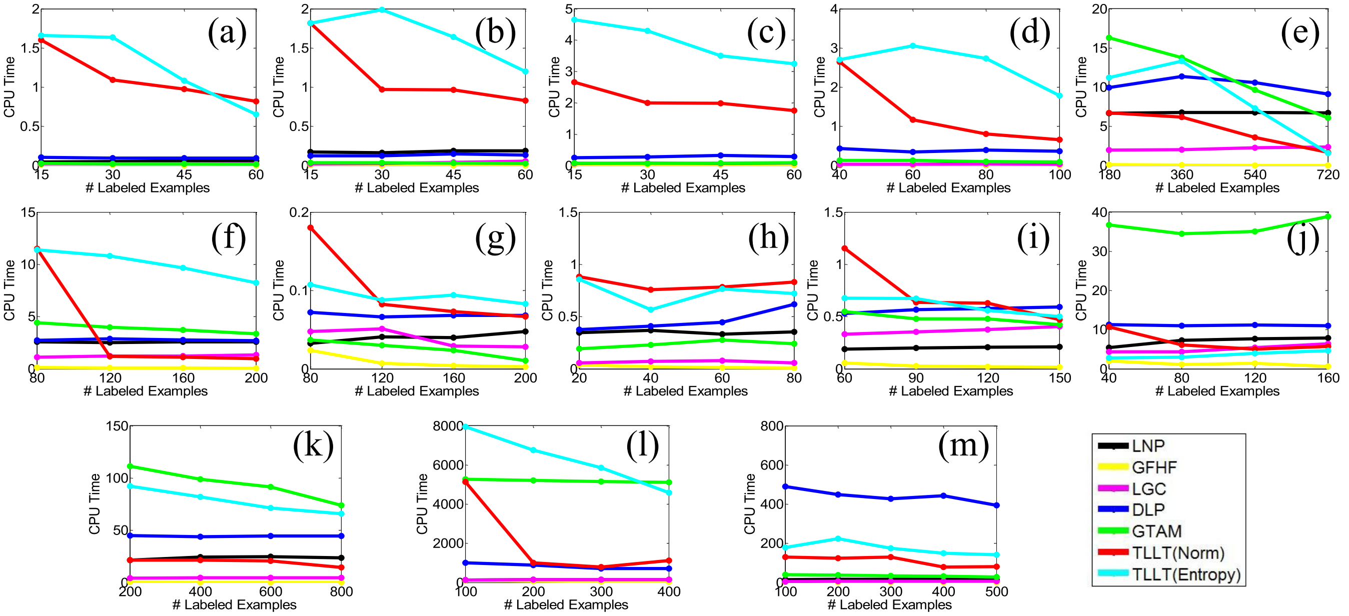

Fig. 6. Running time (unit: second) of all the methods on the ten UCI data sets and three practical data sets. (a)–(j) correspond to UCI datasets Iris, Wine, Seeds, SPECTF, CNAE9, BreastCancer, BreastTissue, Haberman, Leaf, and Banknote. (k)–(m) are practical datasets including 20Newsgroups, USPS, and COIL20, respectively.

rates γ_1 and γ_2 for TLLT (Norm) and TLLT (Entropy) are set to 100 and 0.1, respectively.

When the initial number of labeled examples (i.e., $l^{(0)}$) varies from 100 to 400, the accuracies obtained by the compared methods are plotted in Fig. 5(b). We observe that the baseline methods LGC and GFHF achieve very encouraging performance on this data set, and the corresponding accuracies are $[0.9362, 0.9398, 0.9438, 0.9445]$ and $[0.9338, 0.9421, 0.9449, 0.9480]$ when $l^{(0)} = 100, 200, 300, 400$, respectively. In contrast, the accuracies of TLLT (Norm) and TLLT (entropy) are $[0.9504, 0.9524, 0.9564, 0.9630]$ and $[0.9489, 0.9521, 0.9569, 0.9658]$, respectively, which are superior to LGC and GFHF.

E. Object Recognition

We also apply the proposed TLLT to object recognition problems. COIL20 is a popular public data set for object recognition, which contains 1440 object images belonging to 20 classes, and each object has 72 images shot from different angles. The resolution of each image is 32×32, with 256 gray levels per pixel. Thus, each image is represented by a 1024-dimensional element-wise vector.

We built a 5-NN graph with $\xi = 50$ for GFHF, LGC, GTAM, DLP, and TLLT. The number of neighbors k for LNP was tuned to 10. Other parameters were $\gamma_1 = 1$ for TLLT (Norm) and $\gamma_2 = 0.1$ for TLLT (Entropy). All the algorithms were implemented under $l^{(0)} = 100, 200, 300, 400$ initial labeled examples, and the reported accuracies are the mean values of the outputs of 200 independent runs. Fig. 5(c) shows the comparison results, from which we observe that TLLT hits the highest records among all comparators with $l^{(0)}$ varying from small to large.

F. Running Time

In this section, we compare the running time of all the methods on the data sets appeared in Sections VII-B–VII-E. The experiments were conducted on a desktop with Intel 3.2-GHz i5 CPU and 8-GB memory. For each data set, we plot the CPU seconds averaged over 200 independent runs under different values of $l^{(0)}$, and the results are shown in Fig. 6. We notice that TLLT generally takes longer time than the competing methods. This is because TLLT has the overhead of selecting the most suitable examples in each iteration. The exceptional cases are the data sets CNAE9, banknote, 20Newsgroups, and COIL20, on which either GTAM or DLP needs the longest computational time. More importantly, TLLT is able to improve the performance of baseline methods as revealed in Figs. 4 and 5.

G. Parametric Sensitivity

In Section VI, we theoretically verify that TLLT is insensitive to the change of Gaussian kernel width ξ. Besides, the weighing parameter α in (8) is also a key parameter to be tuned in our method. In this section, we investigate the parametric sensitivity of each of the parameters ξ and α by examining the classification performance of one while the other is fixed. The above three practical data sets 20Newsgroups, USPS, and COIL20 are adopted here, and the results are shown in Fig. 7.

Fig. 7 reveals that TLLT is very robust to the variations of these two parameters, so they can be easily tuned for practical use. The results in Fig. 7(a), (c), and (e) are also consistent with the theoretical validation in Section VI.

H. Summary of Experiments

Based on the above experiments from Sections VII-B–VII-G, we observe that: 1) the proposed TLLT favorably performs to other baseline algorithms in most cases, including the incorporated learning model GFHF; 2) TLLT is very robust to the selection of initial labeled examples and the variation of the two tuning parameters ξ and α; and 3) TLLT spends overall more time than other

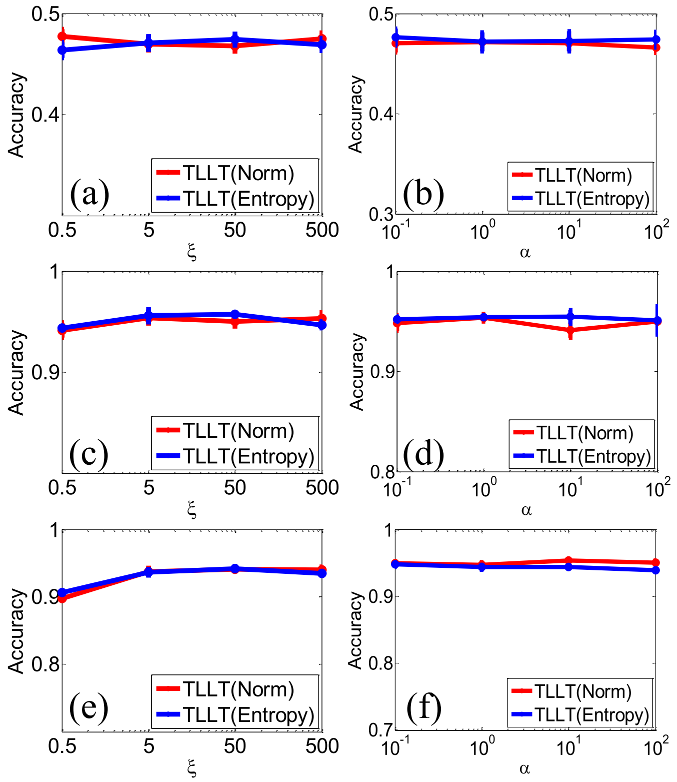

Fig. 7. Parametric sensitivity of TLLT. The first, second, and third rows correspond to 20Newsgroups, USPS, and COIL20 data sets, respectively. (a), (c), and (e) show the variation of accuracy with respect to the kernel width ξ when α is fixed to 1. (b), (d), and (f) evaluate the influence of the tradeoff α to final accuracy under $\xi = 10$.

methods as the teacher has to pick up the simplest examples in each propagation for the stepwise learner.

VIII. Conclusion

This paper proposed a novel label propagation algorithm through iteratively employing a TLLT scheme. The main ingredients contributing to the success of TLLT are: 1) explicitly manipulating the propagation sequence to move from the simple to difficult examples and 2) adaptively determining the feedback-driven curriculums. These two contributions collaboratively lead to higher classification accuracy than the existing algorithms, and exhibit the robustness to the choice of graph parameters. Empirical studies reveal that TLLT can accomplish the state-of-the-art performance in various applications. In the future, we plan to adapt the proposed TLLT framework to more existing algorithms.

Appendix A
Efficient Computation for Commute Time

To apply the Nyström approximation, we uniformly sample q ($q = 10\% \cdot n$ throughout this paper) rows/columns of the original $\mathbf{L}$ to form a submatrix $\mathbf{L}_{q,q}$, and then $\mathbf{L}$ can be approximated by $\tilde{\mathbf{L}} = \mathbf{L}_{n,q}\mathbf{L}_{q,q}^{-1}\mathbf{L}_{q,n}$, where $\mathbf{L}_{n,q}$ represents the $n \times q$ block of $\mathbf{L}$ and $\mathbf{L}_{q,n} = \mathbf{L}_{n,q}^{\top}$. By defining $\mathbf{V} \in \mathbb{R}^{q\times q}$ as an orthogonality matrix, $\tilde{\Theta}$ as a $q \times q$ diagonal matrix, and

$$\mathbf{U} = \begin{pmatrix} \mathbf{L}_{q,q} \\ \mathbf{L}_{n-q,q} \end{pmatrix} \mathbf{L}_{q,q}^{-1/2}\mathbf{V}\tilde{\Theta}^{-1/2} \tag{32}$$

we have (33) according to [41]

$$\begin{aligned} \tilde{\mathbf{L}} &= \mathbf{L}_{n,q}\mathbf{L}_{q,q}^{-1}\mathbf{L}_{q,n} = \begin{pmatrix} \mathbf{L}_{q,q} \\ \mathbf{L}_{n-q,q} \end{pmatrix} \mathbf{L}_{q,q}^{-1} \left(\mathbf{L}_{q,q}^{\top} \quad \mathbf{L}_{n-q,q}^{\top}\right) \\ &= \begin{pmatrix} \mathbf{L}_{q,q} \\ \mathbf{L}_{n-q,q} \end{pmatrix} \mathbf{L}_{q,q}^{-1/2}\mathbf{V}\tilde{\Theta}^{-1/2}\tilde{\Theta}\tilde{\Theta}^{-1/2}\mathbf{V}^{\top}\mathbf{L}_{q,q}^{-1/2}\left(\mathbf{L}_{q,q}^{\top} \quad \mathbf{L}_{n-q,q}^{\top}\right) \\ &= \mathbf{U}\tilde{\Theta}\mathbf{U}^{\top}. \end{aligned} \tag{33}$$

Since $\tilde{\mathbf{L}}$ is positive semidefinite, then according to (32), we require

$$\begin{aligned} \mathbf{I} &= \mathbf{U}^{\top}\mathbf{U} \\ &= \tilde{\Theta}^{-1/2}\mathbf{V}^{\top}\mathbf{L}_{q,q}^{-1/2}\left(\mathbf{L}_{q,q}^{\top} \ \mathbf{L}_{n-q,q}^{\top}\right)\cdot \begin{pmatrix} \mathbf{L}_{q,q} \\ \mathbf{L}_{n-q,q} \end{pmatrix} \mathbf{L}_{q,q}^{-1/2}\mathbf{V}\tilde{\Theta}^{-1/2}. \end{aligned} \tag{34}$$

Multiplying from the left by $\mathbf{V}\tilde{\Theta}^{1/2}$ and from the right by $\tilde{\Theta}^{1/2}\mathbf{V}^{\top}$, we have

$$\begin{aligned} \mathbf{V}\tilde{\Theta}\mathbf{V}^{\top} &= \mathbf{L}_{q,q}^{-1/2}\left(\mathbf{L}_{q,q}^{\top} \ \mathbf{L}_{n-q,q}^{\top}\right)\cdot \begin{pmatrix} \mathbf{L}_{q,q} \\ \mathbf{L}_{n-q,q} \end{pmatrix} \mathbf{L}_{q,q}^{-1/2} \\ &= \mathbf{L}_{q,q} + \mathbf{L}_{q,q}^{-1/2}\mathbf{L}_{n-q,q}^{\top}\mathbf{L}_{n-q,q}\mathbf{L}_{q,q}^{-1/2}. \end{aligned} \tag{35}$$

Therefore, by comparing (33) and (35), we know that the matrix $\mathbf{U}$ containing all the eigenvectors $\mathbf{u}_i$ ($i = 1, \ldots, n$) can be obtained by conducting SVD on $\mathbf{L}_{q,q} + \mathbf{L}_{q,q}^{-1/2}\mathbf{L}_{n-q,q}^{\top}\mathbf{L}_{n-q,q}\mathbf{L}_{q,q}^{-1/2}$, and then plugging $\mathbf{V}$ and $\tilde{\Theta}$ to (32). Similar to [41], we assume that the pseudoinverses are used in place of inverses in above derivations when the matrix $\mathbf{L}_{q,q}$ is not invertible.

The complexity for computing the commute time between examples via Nyström approximation is $\mathcal{O}(q^3)$, which is caused by finding $\mathbf{L}_{q,q}^{-1/2}$ in (35) and the SVD on $\mathbf{L}_{q,q} + \mathbf{L}_{q,q}^{-1/2}\mathbf{L}_{n-q,q}^{\top}\mathbf{L}_{n-q,q}\mathbf{L}_{q,q}^{-1/2}$. This significantly reduces the cost of directly solving the original eigensystem that takes $\mathcal{O}(n^3)$ ($n \gg q$) complexity.

Appendix B
Updating $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$

Given $\Sigma_{\mathcal{S},\mathcal{L}}$, $\Sigma_{\mathcal{L},\mathcal{S}}$, and $\Sigma_{\mathcal{L},\mathcal{L}}$ are the submatrices of the kernel matrix Σ indexed by the associated subscripts; then after one iteration, the kernel matrix on the labeled set is updated by

$$\Sigma_{\mathcal{L},\mathcal{L}} := \begin{pmatrix} \Sigma_{\mathcal{L},\mathcal{L}} & \Sigma_{\mathcal{L},\mathcal{S}} \\ \Sigma_{\mathcal{S},\mathcal{L}} & \Sigma_{\mathcal{S},\mathcal{S}} \end{pmatrix}. \tag{37}$$

$$\Sigma_{\mathcal{L},\mathcal{L}}^{-1} := \begin{pmatrix} \Sigma_{\mathcal{L},\mathcal{L}}^{-1} + \Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\left(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\right)^{-1}\Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1} & -\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\left(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\right)^{-1} \\ -\left(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\right)^{-1}\Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1} & \left(\Sigma_{\mathcal{S},\mathcal{S}} - \Sigma_{\mathcal{S},\mathcal{L}}\Sigma_{\mathcal{L},\mathcal{L}}^{-1}\Sigma_{\mathcal{L},\mathcal{S}}\right)^{-1} \end{pmatrix} \tag{36}$$

As a result, its inverse can be efficiently computed by using the blockwise inversion equation [42] as revealed by (36), shown at the bottom of the previous page.

Note that in (36), we only need to invert an $s \times s$ matrix, which is much more efficient than inverting the original $l \times l$ ($l \gg s$ in later propagations) matrix. Moreover, s will not be quite large, since only a small proportion of unlabeled examples are incorporated into the curriculum per propagation. Therefore, $\Sigma_{\mathcal{L},\mathcal{L}}^{-1}$ can be updated efficiently.

References

[1] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. San Rafael, CA, USA: Morgan & Claypool Pub., 2009.

[2] K. C. A. Kumar and C. De Vleeschouwer, "Discriminative label propagation for multi-object tracking with sporadic appearance features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 2000–2007.

[3] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3166–3173.

[4] J. Ugander and L. Backstrom, "Balanced label propagation for partitioning massive graphs," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, Rome, Italy, Feb. 2013, pp. 507–516.

[5] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.

[6] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2002, pp. 945–952.

[7] X.-M. Wu, Z. Li, A. M. So, J. Wright, and S.-F. Chang, "Learning with partially absorbing random walks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 3077–3085.

[8] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2003, pp. 321–328.

[9] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.

[10] B. Wang, Z. Tu, and J. K. Tsotsos, "Dynamic label propagation for semi-supervised multi-class multi-label classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 425–432.

[11] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 679–686.

[12] Y. Fujiwara and G. Irie, "Efficient label propagation," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 784–792.

[13] C. Gong, D. Tao, K. Fu, and J. Yang, "Fick's law assisted propagation for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2148–2162, Sep. 2015.

[14] M. Karasuyama and H. Mamitsuka, "Multiple graph label propagation by sparse integration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1999–2012, Dec. 2013.

[15] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[16] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. A. K. Suykens, "Multiclass semisupervised learning based upon kernel spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 720–733, Apr. 2015.

[17] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, Jul. 1993.

[18] F. Khan, B. Mutlu, and X. Zhu, "How do humans teach: On curriculum learning and teaching dimension," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 1449–1457.

[19] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 41–48.

[20] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2010, pp. 1189–1197.

[21] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. G. Hauptmann, "Self-paced learning with diversity," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2078–2086.

[22] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Austin, TX, USA, Jan. 2015, pp. 2078–2086.

[23] A. Shinohara and S. Miyano, "Teachability in computational learning," *New Generat. Comput.*, vol. 8, no. 4, pp. 337–347, Feb. 1991.

[24] F. J. Balbach and T. Zeugmann, "Teaching randomized learners," in *Proc. Annu. Conf. Learn. Theory*, Pittsburgh, PA, USA, Jun. 2006, pp. 229–243.

[25] A. Singla, I. Bogunovic, G. Bartók, A. Karbasi, and A. Krause, "Near-optimally teaching the crowd to classify," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 154–162.

[26] S. Zilles, S. Lange, R. Holte, and M. Zinkevich, "Models of cooperative teaching and learning," *J. Mach. Learn. Res.*, vol. 12, pp. 349–384, Feb. 2011.

[27] X. Zhu, "Machine teaching for Bayesian learners in the exponential family," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 1905–1913.

[28] K. R. Patil, X. Zhu, L. Kopeć, and B. C. Love, "Optimal teaching for limited-capacity human learners," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2465–2473.

[29] X. Zhu, "Machine teaching: An inverse problem to machine learning and an approach toward optimal education," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Austin, TX, USA, Jan. 2015, pp. 2078–2086.

[30] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Austin, TX, USA, Jan. 2015, pp. 1–7.

[31] X. Zhu, J. D. Lafferty, and Z. Ghahramani, "Semi-supervised learning: From Gaussian fields to Gaussian processes," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-03-175, Aug. 2003.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.

[33] H. Qiu and E. R. Hancock, "Clustering and embedding using commute times," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1873–1890, Nov. 2007.

[34] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. San Francisco, CA, USA: Academic, 2014.

[35] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Math. Program.*, vol. 142, no. 1, pp. 397–434, Dec. 2013.

[36] P.-A. Absil and J. Malick, "Projection-like retractions on matrix manifolds," *SIAM J. Optim.*, vol. 22, no. 1, pp. 135–158, Jan. 2012.

[37] R. Fletcher, "On the Barzilai–Borwein method," in *Optimization and Control With Applications*. New York, NY, USA: Springer, 2005, pp. 235–256.

[38] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, "Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints," *SIAM J. Optim.*, vol. 6, no. 3, pp. 674–703, Jul. 1996.

[39] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, Aug. 2003, pp. 912–919.

[40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012.

[41] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[42] W. W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, no. 2, pp. 221–239, Jun. 1989.

[43] M. Karasuyama and H. Mamitsuka, "Manifold-based similarity adaptation for label propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 1547–1555.

[44] J. R. Taylor and W. Thompson, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. Bristol, PA, USA: IOP Pub., 1998.

[45] J. Wang, T. Jebara, and S.-F. Chang, "Graph transduction via alternating minimization," in *Proc. Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 1144–1151.

[46] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

Chen Gong received the bachelor's degree from the East China University of Science and Technology (ECUST), Shanghai, China, in 2010. He is currently pursuing the Ph.D. degrees at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University (SJTU), and the Centre for Quantum Computation & Intelligent Systems, University of Technology Sydney (UTS), under the supervision of Prof. J. Yang and Prof. D. Tao. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published 23 technical papers at prominent journals and conferences, such as the IEEE T-NNLS, T-IP. IEEE T-CYB, CVPR, AAAI, ICME, and so on.

Dacheng Tao (F'15) is currently a Professor of Computer Science with the Centre for Quantum Computation & Intelligent Systems, and the Faculty of Engineering and Information Technology at the University of Technology Sydney. He mainly applies statistics and mathematics to data analytics problems and his research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as the IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM'07, the Best Student Paper Award in the IEEE ICDM'13, and the 2014 ICDM 10-Year Highest-Impact Paper Award. He received the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award, and the UTS Vice-Chancellor's Medal for Exceptional Research. He is a fellow of the IEEE, OSA, IAPR, and SPIE.

Wei Liu (M'14) received the Ph.D. degree from Columbia University, New York, NY, USA, in 2012.

He has been a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, since 2012. Currently, he is a Research Staff in Didi Research, Beijing, China. His current research interests include machine learning, big data analytics, computer vision, pattern recognition, and information retrieval.

Dr. Liu was a recipient of the 2013 Jury Award for Best Thesis of Columbia University.

Liu Liu received the B.Sc. degree from the Hebei University of Technology, Tianjin, China, in 2011, and the M.Eng. degree from Beihang University, Beijing, China, in 2014. She is currently pursuing the Ph.D. degree with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia.

Her current research interests include machine learning and optimization.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University, Hamburg, Germany, in 1994.

He is currently a Professor with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., the National Science Foundation and the 863 National High Tech Plan), had published one book in German, and has authored over 200 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.