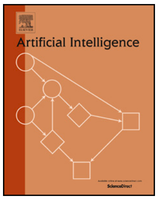
Artificial Intelligence 348 (2025) 104396

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



MATE: Masked optimal transport with dynamic selection for partial label graph learning

Yiyang Gu [a], Binqi Chen [a], Zihao Chen [b], Ziyue Qiao [c], Xiao Luo [d,*], Junyu Luo [a], Zhiping Xiao [e,*], Wei Ju [a], Ming Zhang [a,*]

[a] School of Computer Science, National Key Laboratory for Multimedia Information Processing, PKU-Anker LLM Lab, Peking University, Beijing, China
[b] School of Mathematical Sciences, Peking University, Beijing, China
[c] School of Computing and Information Technology, Great Bay University, Dongguan, China
[d] Department of Computer Science, University of California, Los Angeles, USA
[e] Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, USA

ARTICLE INFO

Keywords:
Graph classification
Partial label learning
Graph neural networks
Masked optimal transport
Partial energy discrepancy

ABSTRACT

This paper investigates the problem of partial label graph learning, in which every graph is associated with a set of candidate labels. Previous methods for weakly supervised graph classification often provide pseudo-labels for graph samples that could be overconfident and biased towards the dominant classes, thus resulting in substantial error accumulation. In this paper, we introduce a new framework named Masked Optimal Transport with Dynamic Selection (MATE) for partial label graph learning, which improves the quality of graph assignments from the perspectives of class balancing and uncertainty mining. In particular, our MATE masks probabilities out of candidate sets and then adopts optimal transport to optimize the assignments without class biases. This design is based on the assumption that the true label distribution is class-balanced or nearly balanced, which is common in various training datasets and real-world scenarios. To further reduce potential noise, we propose a novel scoring metric termed partial energy discrepancy (PED) to evaluate the uncertainty of assignments, and then introduce a dynamic selection strategy that modifies the sample-specific thresholds via momentum updating. Finally, these samples are divided into three levels, i.e., confident, less-confident, and unconfident and each group is trained separately in our collaborative optimization framework. Extensive experiments on various benchmarks demonstrate the superiority of our MATE compared to various state-of-the-art baselines.

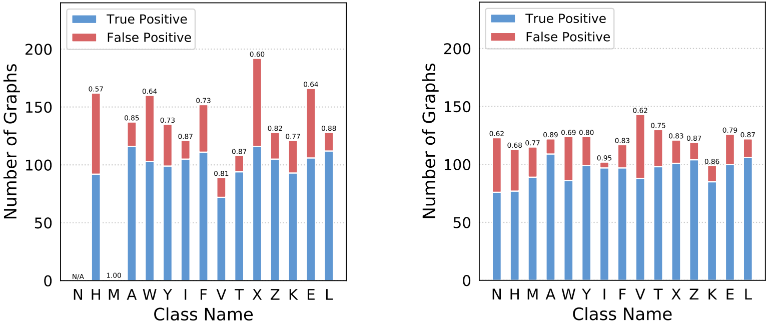
1. Introduction

Graphs are a pervasive form of data representation in the real world such as social networks [13,59,12] and biochemical networks [63,55]. To manage a variety of scenarios, graph neural networks (GNNs) [29,19,67,66,62,58] have been developed to summarize graph-structured data into graph-level representations, which can benefit applications such as protein activation forecasting. Typically, these methods follow the paradigm of neighborhood aggregation to update node representations, which are

[*] Corresponding authors.
E-mail addresses: yiyanggu@pku.edu.cn (Y. Gu), xiaoluo@cs.ucla.edu (X. Luo), patxiao@uw.edu (Z. Xiao), mzhang_cs@pku.edu.cn (M. Zhang).

https://doi.org/10.1016/j.artint.2025.104396
Received 6 November 2023; Received in revised form 28 June 2025; Accepted 21 July 2025

Available online 29 July 2025



(a) Vanilla Pseudo-Labeling

(b) Our Pseudo-Labeling

Fig. 1. Number of graphs classified into various classes in Letter-High dataset with different pseudo-labeling strategies. It can be observed that learning with vanilla biased pseudo labels is prone to get imbalanced classification under label ambiguity (a), thus we propose masked optimal transport for more balanced and accurate pseudo-labeling (b).

summarized by graph pooling operators [15,1,34]. In the end, the resulting graph-level embeddings are passed to a classifier to produce the final prediction.

However, since they contain abundant parameters, GNNs are all data-hungry, which requires a large number of labeled graph samples for optimization. In fact, collecting precisely labeled graph samples is prohibitively costly. For example, we need complicated density functional theory (DFT) [7,2] calculations and costly experiments to obtain the characteristics of chemical compounds. A feasible solution is to utilize automatic annotation tools [16], which could introduce extensive label ambiguity. Towards this end, this work explores an under-explored yet practical task of partial label graph learning, in which every graph sample is associated with a set of candidate labels.

Unlike standard supervised graph learning, partial label graph learning requires label ambiguity to be resolved. While there has been research on partial label learning in Euclidean data like images [14,42], the investigation of complicated non-Euclidean graph data is still limited. Lately, a number of weakly supervised graph classification methods have been proposed [26,40], which usually combine pseudo-labeling strategy [64,36,33] with GNNs. They leverage GNN models to generate pseudo-labels for unlabeled graphs, and then expand the dataset with them. However, pseudo-labeling could result in an overconfident observation [6], which would result in error accumulation in subsequent optimization, and the complex structure of graphs would further increase the difficulties of obtaining clean samples. Worse still, the pseudo-labeling strategy would be biased towards simpler semantic information, which could cause optimization to overlook partial classes (see Fig. 1). In this work, we focus on the setting where the distribution of the ground truth labels is class-balanced or approximately balanced, a common condition in many real-world scenarios and various training datasets [32,54]. Therefore, the class balance of the pseudo-labeling strategy is crucial to mitigating bias and improving the quality of label assignments.

To this end, we propose a novel approach named Masked Optimal Transport with Dynamic Selection (MATE) for partial label graph learning. At a high level, we take a look at the precision of graph assignments in pseudo-labeling schemes and then improve the quality from two fundamental aspects, i.e., class balancing and uncertainty mining. Specifically, we apply a mask to the network output to eliminate probabilities out of candidate sets and then design an optimal transport objective, which is maximized using the Sinkhorn-Knopp algorithm [30,4] to improve the class balance of graph assignments. In addition, we provide a novel uncertainty scoring function called partial energy discrepancy (PED), which takes into account both the sharpness of the label distribution and the sufficiency degree of the optimization. Then, a dynamic selection strategy is adopted which modifies the sample-specific thresholds via the process of momentum updating. To further enhance the robustness of pseudo-labeling, we integrate both weak and strong augmented views for each graph sample and divide the whole dataset into three distinct groups, namely confident, less-confident, and unconfident groups. These groups are trained individually based on their respective uncertainty ratings, and then summarized into a collaborative optimization framework. We also increase the discriminability of graph representations using learning to cluster to mitigate the issue of partial samples consistently obtaining unconfident assignments under inadequate supervision. The proposed MATE has been extensively evaluated on a range of benchmarks, and the results consistently demonstrate its superiority over numerous state-of-the-art baselines. The key contributions of this work are as follows:

- We study an under-explored yet practical graph partial label learning problem and explore the potential limitations of graph assignments under label ambiguity.
- To enhance the quality of pseudo-labeling, we not only formulate an optimal transport objective for the class balance of graph assignments, but also introduce partial energy discrepancy for uncertainty measurement and a dynamic selection strategy using historical scores.
- Comprehensive experiments across multiple graph benchmark datasets demonstrate the proposed MATE's superiority over various state-of-the-art methods.

Table 1
Overview of main notations.

Notation	Description
G	A graph
$\mathcal{V}$	The collection of nodes in a graph
$\mathcal{E}$	The collection of edges in a graph
C	The number of graph classes
Y_i	The candidate label set for the graph G_i
$\mathcal{D}$	The training set
$\hat{\boldsymbol{p}}$	The predicted label distribution
$\boldsymbol{Q}$	The refined assignments through masked optimal transport
B	The size of a mini-batch
$\mathbf{1}_B$	The column vectors of all ones with length of B
$\mathbf{1}_C$	The column vectors of all ones with length of C
PED	The proposed partial energy discrepancy for uncertainty measure
$\tau_i^{\mathrm{PED}}(t)$	the dynamic threshold for the graph G_i on the t^{th} epoch
$\hat{G}_i^{(w)}$	The weakly augmented view of the graph G_i
$\hat{G}_i^{(s)}$	The strongly augmented view of the graph G_i
$\mathcal{C}$	The confident group
$\mathcal{W}$	The less-confident group
$\mathcal{U}$	The unconfident group

2. Related work

2.1. Graph neural networks

In recent years, graph neural networks (GNNs) [29,19,37,44,24,66,62] have shown significant achievements within the domain of graph-based machine learning including graph classification. Typically, these approaches use GNNs to produce graph-level representations, which are then utilized by downstream classifiers in an end-to-end manner. Typically, these approaches use the message passing mechanism [17] to update node representations, which are then summarized using a global pooling operator [1,34] for graph-level representations. Recently, Graph Transplant [44] has been proposed to improve the performance of GNNs for graph-level classification by learning the node saliency for GNNs and conducting the Mixup-like graph augmentation. There are also several Transformer-based architectures for effective graph representation learning [61,3,5]. Despite the encouraging results, these methods often depend on a considerable number of accurate graph labels, which are prohibitively costly in practical scenarios [26]. Towards this end, our study aims to investigate an under-explored but realistic challenge of partial label graph learning, in which every graph is associated with a set of candidate labels.

2.2. Partial label learning

The objective of partial label learning is to address scenarios where every training instance is linked to a set of candidate labels [25,23]. Early approaches attempt to assign equal importance to all potential labels during the optimization based on uniform assumption [25,8,18]. Another area of study is the use of label disambiguation techniques [68,14,42,52], which aim to mitigate the influence of noisy labels. In recent years, a variety of self-supervised learning techniques including contrastive learning and consistency regularization, have been introduced to enhance representation learning. For example, PiCO [51] acquires prototypes for each category in the hidden space, which helps the disambiguation of labels based on contrastive learning. Although these approaches have been effective in Euclidean data, the problem on complicated non-Euclidean graph data remains unexplored. A recent attempt is DEER [53], which leverages distribution divergence and soft label correction but may still over-represent dominant classes when generating pseudo-labels, especially under high uncertainty. Towards this end, we propose MATE, which enforces class-balanced assignments via masked optimal transport and dynamically partitions samples with partial energy discrepancy, thereby achieving more reliable label disambiguation for partial label graph learning.

3. Methodology

3.1. Notations and problem definition

Notations. A graph is denoted by $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. $\boldsymbol{x}_v \in \mathbb{R}^F$ denotes the attribute vector of node v in which F is the attribute dimension. $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the adjacency matrix of G. We summarize the main notations in this paper and their corresponding brief descriptions in Table 1. Then we formulate the problem of partial label graph learning.

Definition 1 *(Partial label graph learning).* We assume that we have a training set $\mathcal{D} = \{G_i, Y_i\}_{i=1}^n$, in which every sample G_i is associated with a candidate set Y_i. Y_i is a subset of the whole label space $\mathcal{Y} = \{1, 2, \cdots, C\}$. The true label $y_i \in Y_i$ is the one among the candidate set, which we cannot get access to during training. The aim is to learn a model for accurate graph classification under label ambiguity.

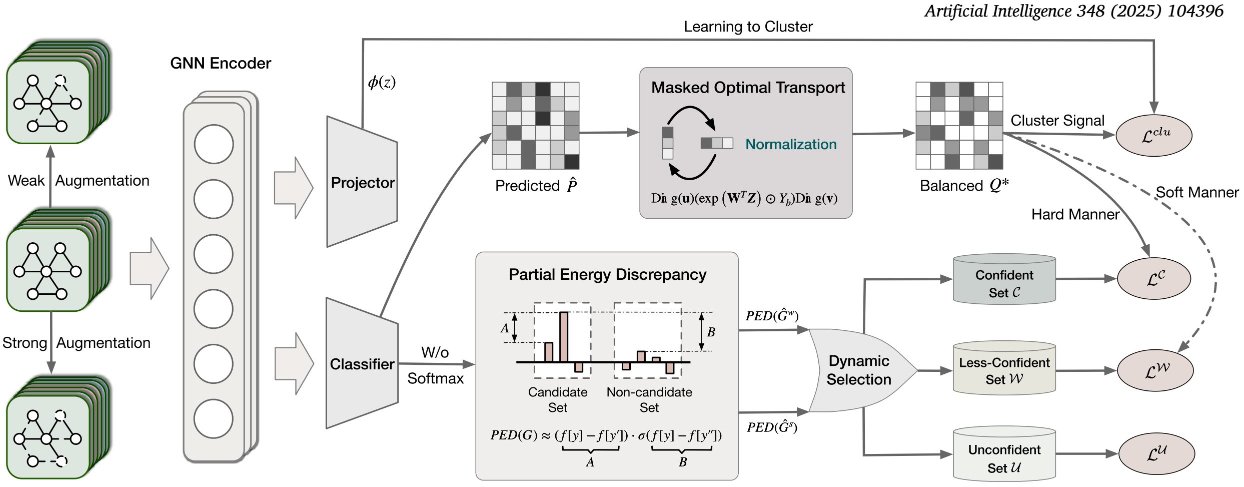

Learning to Cluster
GNN Encoder
$\phi(z)$
Weak Augmentation
Strong Augmentation
Projector
Classifier
W/o Softmax
Predicted $\hat{P}$
Masked Optimal Transport
Normalization
Diag(u)(exp $(\mathbf{W}^T\mathbf{Z}) \odot Y_b$)Diag(v)
Balanced Q^*
Cluster Signal
$\mathcal{L}^{clu}$
Soft Manner
Hard Manner
Partial Energy Discrepancy
A
B
Candidate Set
Non-candidate Set
$PED(G) \approx \underbrace{(f[y]-f[y'])}_{A} \cdot \sigma\underbrace{(f[y]-f[y''])}_{B}$
$PED(\hat{G}^w)$
$PED(\hat{G}^s)$
Dynamic Selection
Confident Set $\mathcal{C}$
Less-Confident Set $\mathcal{W}$
Unconfident Set $\mathcal{U}$
$\mathcal{L}^{\mathcal{C}}$
$\mathcal{L}^{\mathcal{W}}$
$\mathcal{L}^{\mathcal{U}}$

Fig. 2. Overview of the proposed framework MATE. MATE generates two different augmented views for each graph and feeds them into the GNN-based encoder. We utilize the masked optimal transport to optimize the assignments for class balance. Partial energy discrepancy is also introduced to select confidence samples, which generates confident, less-confident, and unconfident groups for separate training strategies. Finally, we enhance the discriminative ability of MATE by learning to cluster.

3.2. Framework overview

This work explores the problem of partial label graph learning and proposes a novel approach named MATE to solve the problem. Specifically, to generate unbiased predictions, we adopt a GNN-based encoder to capture graph embeddings and then solve an optimal transport objective with a mask based on candidate sets. Moreover, we introduce a new uncertainty metric called partial energy discrepancy and a dynamic selection strategy that adjusts the sample-specific thresholds via momentum updating. To improve the robustness of the model, both strongly and weakly augmented views of each graph are generated, which help partition the dataset into the confident, less-confident, and unconfident sets. These sets are then trained separately based on their confidence in a collaborative optimization framework. An overview of the proposed MATE can be found in Fig. 2 and we would then elaborate on each crucial component of our MATE in the following parts.

3.3. GNN-based encoder

To generate the graph-level representations, we adopt graph neural networks (GNNs), which can extract structural information using the message passing mechanism [17]. We represent the embedding of $v \in G$ at layer l as $\boldsymbol{h}_v^{(l)}$. The updating rule is written as:

$$\boldsymbol{h}_v^{(l)} = \mathrm{COM}\left(\boldsymbol{h}_v^{(l-1)}, \mathrm{AGG}\left(\left\{\boldsymbol{h}_u^{(l-1)}\right\}_{u\in\mathcal{N}(v)}\right)\right), \tag{1}$$

in which $\mathcal{N}(v)$ represents the neighbors of v. AGG and COM represent the aggregation and combination functions, respectively. After stacking L layers, we adopt a readout function to aggregate all node embeddings in the final layer as below:

$$\boldsymbol{z} = f_{enc}(G) = \sum_{v\in G} \boldsymbol{h}_v^{(L)}. \tag{2}$$

In the end, an MLP-based classifier ϕ_{cla} with the softmax activation function is adopted to connect graph representation with label distributions as:

$$\hat{\boldsymbol{p}} = \phi_{cla}(\boldsymbol{z}) = \mathrm{Softmax}(\mathbf{W}_2 \cdot \mathrm{ReLU}(\mathbf{W}_1\boldsymbol{z} + \mathbf{b}_1) + \mathbf{b}_2), \tag{3}$$

where $\hat{\boldsymbol{p}} \in [0,1]^C$ denotes the predicted probability distribution, $\mathbf{W}_1$ and $\mathbf{W}_2$ denote the parameter matrices for the two linear layers, $\mathbf{b}_1$ and $\mathbf{b}_2$ serve as the associated bias parameters, and ReLU and Softmax are the nonlinear activation operations.

3.4. Masked optimal transport for balanced pseudo-labeling

Pseudo-labeling [39,26,40,41] is a popular technique in weakly supervised graph classification, which utilizes the trained model to generate predicted labels and then expand the training data. However, these approaches will generate biased pseudo-labels toward dominant classes [27]. In this paper, "bias" refers to the systematic over-representation of dominant classes when assigning pseudo-labels due to the reliance on high-confidence but potentially skewed predictions. This often occurs when models favor semantically easier classes or dominant patterns in the dataset, leading to an overestimation of their prevalence. Note that we consider the condition where the ground truth label distribution is class-balanced or approximately class-balanced in this paper, which is common in real-world scenarios [32,54]. To tackle this, we propose masked optimal transport, which can obtain class-balanced assignments for graph samples under the label ambiguity, closer to the ground truth label distribution.

Specifically, for a mini-batch containing B graphs, we stack the predicted distributions as a matrix $\hat{\boldsymbol{P}} \in [0,1]^{C\times B}$, where $\hat{\boldsymbol{P}}_{ij} = \hat{p}(y=i|G_j) = \phi_{cla}\left(f_{enc}\left(G_j\right)\right)[i]$, and define a matrix to indicate the refined graph assignments $\boldsymbol{Q} \in \mathbb{R}^{C\times B}$ with each element $\boldsymbol{Q}_{ij} = q(y=i|G_j)$, which denotes the refined probability of assigning label i to graph G_j, and is computed at the mini-batch level via the proposed masked optimal transport mechanism. The goal is to align the refined graph assignments $\boldsymbol{Q}$ with the original predicted distributions $\hat{\boldsymbol{P}}$ in a way that minimizes the transportation cost while respecting the constraints of the class balance. The transportation problem can be formulated as:

$$\min_{\boldsymbol{Q}} \left\langle -\hat{\boldsymbol{P}}, \boldsymbol{Q} \right\rangle \quad \text{s.t.} \quad \boldsymbol{Q}\mathbf{1}_B = \frac{1}{C}\mathbf{1}_C, \quad \boldsymbol{Q}^T\mathbf{1}_C = \frac{1}{B}\mathbf{1}_B, \tag{4}$$

where $\langle\cdot,\cdot\rangle$ denotes the (Frobenius) inner product. $\mathbf{1}_B$ and $\mathbf{1}_C$ represent the column vectors of all ones, where the length of the vector corresponds to B and C, respectively. $-\hat{\boldsymbol{P}}$ can be regarded as the cost matrix of the transportation problem. The first term encourages the similarity between $\boldsymbol{Q}$ and the original predicted distributions $\hat{\boldsymbol{P}}$. The second term ensures that the distribution of $\boldsymbol{Q}$ across classes is uniform, encouraging class balance in the refined graph assignments. The third term enforces each column of $\boldsymbol{Q}$ to be a scaled probability distribution. To address the computational challenges of solving the standard linear programming problem, a smoothed version of the optimization problem is introduced by adding a negative entropy regularization term to the objective. Then we can introduce Lagrange multipliers to help solve the problem, which leads to the following modified objective function to be maximized:

$$\Gamma = \left\langle \hat{\boldsymbol{P}}, \boldsymbol{Q} \right\rangle + \lambda\mathcal{H}(\boldsymbol{Q}) + < \boldsymbol{t}, \boldsymbol{Q}\mathbf{1}_B - \frac{1}{C}\mathbf{1}_C > + \left\langle \boldsymbol{g}, \boldsymbol{Q}^\top\mathbf{1}_C - \frac{1}{B}\mathbf{1}_B \right\rangle, \tag{5}$$

where $\mathcal{H}(\boldsymbol{Q}) = -\sum_{ij}\boldsymbol{Q}_{ij}\log\boldsymbol{Q}_{ij}$ is the entropy of the matrix, and λ is a coefficient for the entropy regularization term. In computation, we set $\lambda = 2$. $\boldsymbol{t} \in \mathbb{R}^B$ and $\boldsymbol{g} \in \mathbb{R}^C$ denote two Lagrange multipliers. By calculating the gradient of the objective, we can have:

$$\frac{\partial\Gamma}{\partial\left(\boldsymbol{Q}_{ij}\right)} = \hat{\boldsymbol{P}}_{ij} - \lambda\left(\log(\boldsymbol{Q})_{ij} + 1\right) + \boldsymbol{t}_i + \boldsymbol{g}_j. \tag{6}$$

Then, we can get the solution of Eq. (5) as:

$$\mathbf{Q}^* = \mathrm{Diag}(\mathbf{u})\exp\left(\frac{\hat{\boldsymbol{P}}}{\lambda}\right)\mathrm{Diag}(\mathbf{v}), \tag{7}$$

in which $\mathbf{u} = \exp\left(\frac{1}{2} + \frac{g}{\lambda}\right)$, $\mathbf{v} = \exp\left(\frac{1}{2} + \frac{t}{\lambda}\right)$, $\mathrm{Diag}(\cdot)$ denotes the diagonal matrix constructed from a vector, and $\exp(\cdot)$ denotes the element-wise exponential of a matrix. Moreover, note that in our case, when the label is out of the candidate set, the probability of corresponding assignments should be zero. In other words, let $\boldsymbol{Y}^B \in \mathbb{R}^{C\times B}$ denote the stacked matrix indicating the candidate set, i.e., $\boldsymbol{Y}^B_{ij} = 1$ when $i \in Y_j$, and we have:

$$\left\langle \boldsymbol{Q}, \mathbf{1} - \boldsymbol{Y}^B \right\rangle = 0. \tag{8}$$

To achieve this, we revise Eq. (7) into the following equation:

$$\mathbf{Q}^* = \mathrm{Diag}(\mathbf{u})\left(\exp\left(\frac{\hat{\boldsymbol{P}}}{\lambda}\right)\odot\boldsymbol{Y}^B\right)\mathrm{Diag}(\mathbf{v}), \tag{9}$$

where $\mathbf{u} = \exp\left(\frac{1}{2} + \frac{g}{\lambda}\right)$, $\mathbf{v} = \exp\left(\frac{1}{2} + \frac{t}{\lambda}\right)$, and $\odot$ is the Hadamard product to mask the elements out of the candidate set. Then, denoting $\exp\left(\frac{\hat{\boldsymbol{P}}}{\lambda}\right)\odot\boldsymbol{Y}^B =: \boldsymbol{K}$, we can use the Sinkhorn-Knopp algorithm [48,30,9,49] to approach the optimal multipliers iteratively, i.e. $(\mathbf{u},\mathbf{v}) \leftarrow (\mathbf{1}_C \;./\; \boldsymbol{K}\mathbf{v}, \mathbf{1}_B \;./\; \boldsymbol{K}^\top\mathbf{u})$, where ./ denotes element-wise division. With the proposed masked optimal transport, we can generate more balanced and accurate assignments for graphs to capture the true label distribution better and benefit subsequent optimization. It can lead to smoother training objectives and improved optimization with more stable gradients and better generalization by preventing overfitting to dominant classes.

We provide a theoretical analysis to demonstrate the effectiveness of our proposed masked optimal transport (OT). Since the masked OT framework encourages $\boldsymbol{Q}$ to satisfy the transportation polytope constraints (i.e., $\boldsymbol{Q}\mathbf{1}_B = \frac{1}{C}\mathbf{1}_C$ and $\boldsymbol{Q}^\top\mathbf{1}_C = \frac{1}{B}\mathbf{1}_B$), the refined graph assignments $\boldsymbol{Q}$ are more class-balanced and less biased toward any specific class naturally. We further prove that there exists a lower bound on the similarity between $\boldsymbol{Q}$ and the prior uniform label distributions derived from the candidate label sets.

Theorem 3.1. *Let $\boldsymbol{X} = \frac{1}{B}\hat{\boldsymbol{P}}$ be the normalized version of $\hat{\boldsymbol{P}}$. Let $\boldsymbol{L} \in [0,1]^{C\times B}$ denote the prior uniform label distributions, where $\boldsymbol{L}_{ij} = \begin{cases} 1/|Y_j|, & \text{if} \quad i \in Y_j, \\ 0, & \text{otherwise.} \end{cases}$ Further suppose that $\|\hat{\boldsymbol{P}} - \boldsymbol{L}\|_{\max} < \varepsilon$ for some $\varepsilon > 0$. Then after our masked optimal transport, we have*

$$\langle\boldsymbol{X},\boldsymbol{L}\rangle + C_{\boldsymbol{X}} \leq \langle\boldsymbol{Q},\boldsymbol{L}\rangle + 3\varepsilon,$$

where $C_{\boldsymbol{X}} = \sum_{i=1}^{C}\left(\frac{1}{C} - \frac{1}{B}\sum_{j=1}^{B}\hat{\boldsymbol{P}}_{ij}\right)\left(\sum_{j=1}^{B}\boldsymbol{X}_{ij}\right)$.

Remark. Using the similarity between $\hat{P}$ and the prior uniform label distributions $\boldsymbol{L}$ as a reference, there exists a lower bound on the similarity between Q and $\boldsymbol{L}$, indicating that Q does not deviate too much from the prior uniform label distributions. This claim is further supported by our empirical results in Fig. 1 and Fig. 4, which show that the refined pseudo-labels Q exhibit a significantly more balanced class distribution compared to the biased predictions $\hat{P}$. On the other hand, given that Q approximately satisfies the transportation polytope constraint of $\boldsymbol{Q}\mathbf{1}_B = \frac{1}{C}\mathbf{1}_C$ and $\boldsymbol{Q}^\top\mathbf{1}_C = \frac{1}{B}\mathbf{1}_B$, it is more class-balanced. We consider the condition where the true label distribution is either class-balanced or approximately class-balanced, which is common in various real-world applications and has been used in many existing works [32,54]. The masked OT can prevent refined assignments from concentrating too much on certain classes or deviating too much from the observable label distributions with ambiguity, allowing Q to be closer to the true label distribution than the original predicted distribution $\hat{P}$. It is worth noting that the above analysis relies on the assumption that the true label distribution is class-balanced or approximately class-balanced. While this holds in many cases, extending the masked OT framework to handle class-imbalanced distributions would be an interesting direction for future research. Proof of Theorem 3.1 can be found in the Appendix A.

3.5. Dynamic selection with partial energy discrepancy

Even when we provide the enhanced assignments for graph samples, they still could contain a certain amount of noise. To tackle this, we need to identify samples with low uncertainty to prevent potential error accumulation. Note that energy [38] has been adopted to measure the uncertainty in out-of-distribution detection [21,22]. In light of this, we put forward a novel uncertainty measure named partial energy discrepancy (PED) and a dynamic selection strategy to select reliable graph samples.

In particular, we first recall the definition of energy:

$$\mathrm{E}\left(G_i\right) = -\log \sum_{c=1}^{C} e^{\boldsymbol{f}_i[c]}, \tag{10}$$

where $\boldsymbol{f}_i$ is the output before the softmax activation and $\boldsymbol{f}_i[c]$ calculates the logit for class c. In our case, confident graphs should contain the following two crucial characteristics: (1) Sharp Distributions. This requires the energy discrepancy to be large after subtracting the maximum logit. (2) Sufficient Training. This requires a significant energy discrepancy between candidate sets and non-candidate sets. From the analysis, our PED is formulated as:

$$\mathrm{PED}(G) = A \cdot \sigma(B), \tag{11}$$

$$A = \log \sum_{c\in Y_i} e^{\boldsymbol{f}_i[c]} - \log\left[\sum_{c\in Y_i} e^{\boldsymbol{f}_i[c]} - \max_{c'\in Y_i}\left\{e^{\boldsymbol{f}_i[c']}\right\}\right], \tag{12}$$

$$B = \log \sum_{c\in Y_i} e^{\boldsymbol{f}_i[c]} - \log \sum_{c''\notin Y_i} e^{\boldsymbol{f}_i[c'']}, \tag{13}$$

where σ is the sigmoid function. By combining negative energy discrepancy after excluding the highest logit value (i.e., term A), and negative energy discrepancy between candidate sets and non-candidate sets (i.e., term B), we can identify well-trained samples with confident assignments. Further, for G_i, assume that y_i is the optimal label in the candidate set, y_i' is the sub-optimal label in the candidate set, y_i'' is the optimal label in the non-candidate set. We have:

$$-\mathrm{E}(G_i) = \log e^{\boldsymbol{f}_i[y_i]} + \log(\sum_{c=1}^{C} e^{\boldsymbol{f}_i[c]-\boldsymbol{f}_i[y_i]}) \approx \boldsymbol{f}_i[y_i]. \tag{14}$$

Therefore, we can approximate the PED score as:

$$\mathrm{PED}(G_i) \approx (\boldsymbol{f}_i[y_i] - \boldsymbol{f}_i[y_i']) \cdot \sigma(\boldsymbol{f}_i[y_i] - \boldsymbol{f}_i[y_i'']). \tag{15}$$

Typically, a higher PED score indicates a more confident sample. Previous approaches usually establish a fixed threshold and choose examples with scores above it, which ignore the varied difficulty of learning different graphs. Instead, we use a dynamic selection strategy [36], where the threshold is updated using the historical PED scores. In formulation, the threshold for G_i is written as:

$$\tau_i^{\mathrm{PED}}(t) = \beta\tau_i^{\mathrm{PED}}(t-1) + (1-\beta)\,\mathrm{PED}(G_i), \tag{16}$$

in which t denotes the epoch index and β is set to 0.99 for robust momentum updating following [20].

3.6. Collaborative optimization from weak and strong augmented views

Given the dynamic selection strategy, we need to partition the whole dataset into different groups and create unique optimization algorithms for each group. To further increase the robustness of the optimization process, we incorporate both weak and strong augmented views for every graph and then compare their PED scores with thresholds to construct distinct groups, namely the confident, less-confident, and unconfident groups. In the end, we provide a unified framework for collaborative optimization where each group is trained in a separate fashion and then enhanced by learning to cluster.

To be specific, we first introduce four popular graph augmentation strategies [60] as follows: (1) Edge perturbation, which involves inserting or removing partial edges from graphs; (2) Node dropping, which randomly gets rid of partial nodes along with their associated edges; (3) Attribute masking, which involves selecting partial nodes and masks their attributes; (4) Subgraph, which selects a subgraph from the full graph using random walks. Here, we adopt two augmentation ratios, i.e., $\rho_w < \rho_s$ for generating weakly and strongly augmented views, $\hat{G}_i^w$ and $\hat{G}_i^s$, respectively. Then, their PED scores can be recorded as $\mathrm{PED}(\hat{G}_i^w)$ and $\mathrm{PED}(\hat{G}_i^s)$. We consider samples with both scores above the threshold as confident and consider samples with only one of the scores over the threshold as less confident. The left samples are regarded as unconfident. In formulation, the confident group $\mathcal{C}$, less-confident group $\mathcal{W}$ and unconfident group $\mathcal{U}$ are defined as:

$$\mathcal{C} = \left\{G_i \mid \mathrm{PED}(\hat{G}_i^w) > \tau_i^{\mathrm{PED}}(t)\right\} \cap \left\{G_i \mid \mathrm{PED}(\hat{G}_i^s) > \tau_i^{\mathrm{PED}}(t)\right\}, \tag{17}$$

$$\mathcal{W} = \left\{G_i \mid \mathrm{PED}(\hat{G}_i^w) > \tau_i^{\mathrm{PED}}(t)\right\} \cup \left\{G_i \mid \mathrm{PED}(\hat{G}_i^s) > \tau_i^{\mathrm{PED}}(t)\right\} - \mathcal{C}, \tag{18}$$

$$\mathcal{U} = \mathcal{D} - \mathcal{C} - \mathcal{W}. \tag{19}$$

Finally, we combine three groups into a collaborative optimization framework. First, we utilize hard graph assignments, i.e., one-hot pseudo-labels of samples in $\mathcal{C}$ to supervise the optimization in a hard manner. Formally,

$$\mathcal{L}^{\mathcal{C}} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|} ||\hat{\boldsymbol{p}}_i - \boldsymbol{y}_i^{hard}||_2^2, \tag{20}$$

where $\boldsymbol{y}_i^{hard}[c] = \begin{cases} 1, & \text{if } c = \arg\max_{c'} \boldsymbol{q}_i[c'] \\ 0, & \text{otherwise} \end{cases}$ represents the one-hot encoded pseudo-label, and $\boldsymbol{q}_i$ is the i^{th} column of $\boldsymbol{Q}$. Second, we utilize the soft graph assignments to supervise the optimization on $\mathcal{W}$ in a soft manner since they are less confident:

$$\mathcal{L}^{\mathcal{W}} = \frac{1}{|\mathcal{W}|}\sum_{i=1}^{|\mathcal{W}|} ||\hat{\boldsymbol{p}}_i - \boldsymbol{y}_i^{soft}||_2^2, \tag{21}$$

where $\boldsymbol{y}_i^{soft} = \boldsymbol{q}_i$ is the i^{th} column of $\boldsymbol{Q}$ derived from masked optimal transport. Finally, for these unconfident samples, we merely utilize the candidate set to train the model:

$$\mathcal{L}^{\mathcal{U}} = -\frac{1}{|\mathcal{U}|}\sum_{i=1}^{|\mathcal{U}|}\sum_{c\notin Y_i} \log(1 - \hat{\boldsymbol{p}}_i[c]), \tag{22}$$

which forces the likelihood of the non-candidate set to be as small as possible for more sufficient training. The final supervised objective is obtained by summarizing these terms:

$$\mathcal{L}^{sup} = \mathcal{L}^{\mathcal{C}} + \mathcal{L}^{\mathcal{W}} + \mathcal{L}^{\mathcal{U}}. \tag{23}$$

Learning to Cluster. However, there could be the risk that partial samples would always be unreliable. To handle this, we use cluster learning, which seeks to bring samples from the same cluster closer while driving those from different clusters further apart [65]. To be more precise, pseudo-labels of all these samples are derived using balanced assignments:

$$\hat{y}_i = argmax_{c=1}^{C} \boldsymbol{q}_i[c]. \tag{24}$$

Then, for each graph sample G_i, we collect the samples with the same semantics as:

$$\Pi(i) = \{j | \hat{y}_i = \hat{y}_j\}. \tag{25}$$

The objective function can be written as:

$$\mathcal{L}^{clu} = -\sum_{i=1}^{N} \log \frac{\sum_{j\in\Pi(i)} \exp\left(\phi\left(\boldsymbol{z}_i\right)\cdot\phi\left(\boldsymbol{z}_j\right)/\tau\right)}{\sum_{j'\notin\Pi(i)} \exp\left(\phi\left(\boldsymbol{z}_i\right)\cdot\phi\left(\boldsymbol{z}_{j'}\right)/\tau\right)}, \tag{26}$$

where $\phi(\cdot)$ is an MLP-based projector to transform graph representations to the embedding space and τ refers to a temperature hyperparameter fixed at 0.07 following [20]. In Eq. (26), the numerator summarizes the distance of intra-cluster sample pairs and the denominator summarizes the distance of inter-cluster sample pairs. In this way, we promote the discriminability of graph representations in the embedding space and therefore enhance the classification performance.

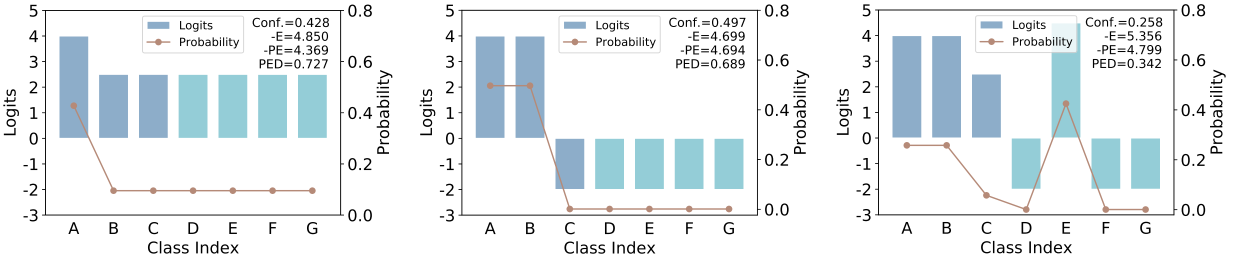
In a nutshell, the final loss objective is summarized into:

$$\mathcal{L} = \mathcal{L}^{sup} + \eta\mathcal{L}^{clu}, \tag{27}$$

where η is a parameter to balance two losses. The complete procedure of our MATE framework is outlined in Algorithm 1.

Algorithm 1 Optimization Algorithm of MATE.

```
Input: Training Set $\mathcal{D} = \{G_i, Y_i\}_{i=1}^{n}$ and hyper-parameters
Output: Trained GNN encoder $f_{enc}$ and MLP classifier $\phi_{cla}$
Initialize the parameters and warm up the model;
while not done do
    Sample a mini-batch from $\mathcal{D}$ and generate weakly and strongly augmented views for each graph;
    Compute class-balanced $\mathbf{Q}^*$ with masked optimal transport using Eq. (9);
    Compute PED($\hat{G}_i^w$) and PED($\hat{G}_i^s$) using Eq. (11);
    Compute the clustering loss $\mathcal{L}^{clu}$ using Eq. (26);
    Dynamically select three sets using Eq. (17)-(19);
    Compute losses for three sets using Eq. (20)-(22);
    Update the parameters by gradient descent to minimize $\mathcal{L}$ using Eq. (27);
    Update PED thresholds by momentum using Eq. (16)
end while
```



(a) Sample I. (b) Sample II. (c) Sample III.

Fig. 3. Distributions of logits and probability of three samples with various scoring functions for uncertainty measurement, including confidence (Conf.), negative Energy (-E), negative Partial Energy (-PE), and Partial Energy Discrepancy (PED). The logits in the candidate set are shown in dark blue, while those in the non-candidate set are shown in light blue. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.7. Further analysis of partial energy discrepancy

To further analyze the effectiveness of our designed scoring function, Partial Energy Discrepancy (PED), we compare it with several common scoring functions for uncertainty measurement, i.e. confidence, Energy and its variant Partial Energy (PE) for partially labeled data. Formally,

- **Confidence**, which corresponds to the maximum probability in the candidate set:

$$\mathrm{Conf}(G_i) = \max_{c \in Y_i} \left\{ \hat{p}_i[c] \right\}. \tag{28}$$

- **Negative Energy**, which computes the log-sum-exp of all the logits,

$$-\mathrm{E}\left(G_i\right) = \log \sum_{c=1}^{C} e^{f_i[c]}. \tag{29}$$

- **Negative Partial Energy**, which computes the log-sum-exp of the logits in the candidate set,

$$-\mathrm{PE}\left(G_i\right) = \log \sum_{c \in Y_i} e^{f_i[c]}. \tag{30}$$

- **Partial Energy Discrepancy**, which combines the negative energy discrepancy after removing the maximum logit (term A), and the negative energy discrepancy between candidate sets and non-candidate sets (term B),

$$\mathrm{PED}(G_i) = A \cdot \sigma(B), \tag{31}$$

$$A = \log \sum_{c \in Y_i} e^{f_i[c]} - \log \left[\sum_{c \in Y_i} e^{f_i[c]} - \max_{c' \in Y_i} \left\{ e^{f_i[c']} \right\} \right], \tag{32}$$

$$B = \log \sum_{c \in Y_i} e^{f_i[c]} - \log \sum_{c'' \notin Y_i} e^{f_i[c'']}. \tag{33}$$

Our proposed partial energy discrepancy can capture the distribution information both inside and outside the candidate set, which cannot be captured by the other three scoring functions sufficiently. We take the predicted distributions of three typical instances as examples to intuitively illustrate this. As we can observe from Fig. 3, the uncertainty of three samples becomes higher and higher from (I) to (III). However, the confidence scoring function gives the sample (II) the highest score, since it ignores the distributions except the maximum value. Besides, the negative Energy and negative partial Energy evaluate sample (III) as the one with the highest score mistakenly, because they only consider the value of the corresponding logits, but cannot capture the important statistical characteristics of predicted distributions, like the sharp degree of the distributions. While our PED score can sort the uncertainty of three samples correctly, i.e. $\mathrm{PED}(G_{\mathrm{I}}) > \mathrm{PED}(G_{\mathrm{II}}) > \mathrm{PED}(G_{\mathrm{III}})$, where the higher PED score corresponds to the lower uncertainty. This owes to its ability to not only capture the distribution characteristics within the candidate set for estimating the sharp degree of distributions, but also models the energy discrepancy across the candidate set and the non-candidate set for evaluating the sufficient degree of training.

3.8. Computational complexity analysis

With N as the number of graphs, $|\mathcal{E}|$ as the average number of edges in the graphs, C as the number of graph classes, B as the batch size, GraphSAGE [19] has a computational complexity of $O(|\mathcal{E}|)$ for each graph. Since the Sinkhorn-Knopp algorithm converges quickly with a few iterations, the time complexity of masked optimal transport is $O(BC)$ for each batch. The computational complexity of PED and supervised losses are also $O(BC)$ for each batch. Besides, computing the clustering loss for a batch takes $O(B^2)$ time. Collectively, the overall computational complexity of MATE is $O(N(|\mathcal{E}| + B + C))$.

4. Experiments

4.1. Experimental setup

4.1.1. Datasets

The assessment of our proposed framework MATE is conducted across five widely-used graph benchmark datasets: ENZYMES [47], Letter-High [46], CIFAR10 [11], COIL-DEL [46], and COLORS-3 [31]:

- **ENZYMES** [47], sourced from the BRENDA enzyme database, comprises a collection of 600 tertiary structures of proteins. It encompasses a diverse array of protein classes, each categorized into one of the six top-level Enzyme Commission (EC) classes.
- **Letter-High** [46] encompasses a set of 15 uppercase letters (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z) expressed as intricate graphs. Each individual letter undergoes a transformation into a prototypical graph, wherein lines are transformed into undirected edges, and the terminal points of these lines are transformed into nodes.
- **CIFAR10** [11], deriving its origin from a kindred vision dataset, is fashioned by abstracting super-pixels from images. These discrete super-pixels are subsequently assembled as nodes, with the resulting structure forming a k-nearest neighbors (kNN) graph that captures the interrelationships intrinsic to these extracted super-pixels.
- **COIL-DEL** [46]. Derived from a vision dataset, COIL-DEL takes shape through the employment of Harris corner detection combined with Delaunay Triangulation techniques. The generated triangulated structure is transmuted into an undirected graph topology, in which the nodes mirror the extremities of lines, and the edges encode the interlinkages between these lines.
- COLORS-3 [31] features a large number of graphs designed for controlled graph reasoning tasks. Each graph contains nodes with distinct colors (red, green, blue) encoded by one-hot features, and the task involves counting the number of nodes with a specific color.

We construct candidate label sets by randomly flipping each negative label into a false positive with probability q following [53]. A larger q increases the size of the candidate set and the level of label ambiguity. In our experiments, we set $q \in \{0.1, 0.3, 0.5\}$ for most datasets, and $q \in \{0.02, 0.05, 0.1\}$ for COIL-DEL, considering its larger label space.

4.1.2. Baselines

We compare our MATE with four groups of competitive baselines: graph convolution methods, graph pooling methods, graph augmentation method, graph contrastive learning method, and weakly supervised graph learning method.

Graph Convolution Methods. Our comparative evaluation includes four distinct varieties of graph convolutional layers, each characterized by its unique attributes:

- **Graph Convolutional Network (GCN)** [29] employs the local first-order approximation of Chebyshev polynomials to enhance the efficiency of spectral graph convolution.
- **Graph Attention Network (GAT)** [50] is distinguished by its incorporation of the attention mechanism, enabling adaptive focus on neighboring nodes in the process of message integration.
- **Graph Isomorphic Network (GIN)** [56] hinges on the utilization of multi-layer perceptrons (MLPs) to fit injective aggregation operators, which is theoretically underpinned by a potent discriminative ability equivalent to the Weisfeiler-Lehman (WL) graph isomorphism test.
- **GraphSAGE** [19]. In order to better scale to large graphs, GraphSAGE samples a local surrounding for every node and then gathers meaningful information from it.

Graph Pooling Methods. We embrace four well-known hierarchical graph pooling techniques as comparisons:

- **TopK Pooling (TopKPool)** [15] involves evaluating the significance of each node through learning a projection vector, and then constructs a coarsened graph by selecting the most significant subset of nodes from the original graph through the TopK operator.
- **Self-Attention Graph Pooling (SAGPool)** [35] capitalizes on self-attention scores generated through graph convolution. This strategy captures both node attributes and the overarching graph structure.
- **Edge Pooling (EdgePool)** [10] harnesses the principle of edge contraction to achieve a sparsified and localized coarsening transformation.
- **Adaptive Structure Aware Pooling (ASAP)** [45] ingeniously integrates TopK-based coarsening approaches with cluster-based coarsening approaches through localized clustering of h-hop neighborhood.

All the graph pooling methods are equipped with GraphSAGE as the backbone convolutional layers, and are set with a pooling ratio of 0.6 during the assessment process.

Graph Augmentation Method. A recent graph augmentation method Graph Transplant [44] is compared.

- **Graph Transplant** [44] is a novel Mixup-inspired graph augmentation approach for effective graph classification. Its functionality involves recognizing sub-structures as composite units and utilizing node saliency information to discern significant subgraphs and dynamically assign labels.

Graph Contrastive Learning Method. We also compare our proposed MATE with a popular graph contrastive learning method, GraphCL [60].

- **GraphCL** [60] is a graph contrastive learning method that learns graph embeddings by contrasting augmented views of the graph. It has demonstrated superior performance on graph-based tasks by capturing the graph's structural and semantic characteristics through a contrastive loss.

Weakly Supervised Graph Learning Method. We further compare our proposed MATE with a recent weakly supervised graph learning method, Twin Graph Neural Network [26].

- **Twin Graph Neural Network (TGNN)** [26] is a joint learning framework for semi-supervised graph classification. It involves a message-passing module and a graph kernel module for exploring graph structural information from complementary perspectives, and makes full use of unlabeled data by maximizing consistency between similarity distributions from two modules.

These baseline approaches are trained by a widely adopted cross-entropy objective to maximize the log-likelihood of the candidate labels for weak supervision. Besides, TGNN is jointly optimized by this weak-supervision objective and its own unsupervised consistency regularization, since partially labeled data can be exploited as not only weakly annotated data but also unlabeled data from two aspects. Note that there is a shortage of partial label graph learning approaches.

4.1.3. *Evaluation protocol and implementation details*

We partition the training/validation/test sets in an 80%:5%:15% proportion for most datasets. For CIFAR10, we utilize the conventional split of 45,000 training, 5,000 validation, and 10,000 test graphs as in [11]. We report results using the mean classification accuracy and the standard deviation over five runs. Our MATE utilizes a two-layer GraphSAGE with a latent dimension of 512 as the GNN Encoder. It's optimized using the Adam optimizer [28] with a starting learning rate of 0.001 and a mini-batch size of 128. The graph augmentation ratios ρ_w and ρ_s for weak and strong views are configured to 0.1 and 0.3, respectively. To get initial graph representations and predicted distributions, we perform a 20-epoch warm-up phase, where it only employs a common graph contrastive loss as GraphCL [60] and utilizes prior uniform label distributions on the candidate sets as the target. We implement our proposed algorithm MATE with PyTorch.

4.2. *Experimental results*

We present the classification accuracy of our MATE compared with strong baseline methods in Table 2, Table 3 and Table 4. From the results, we have the following observations:

- Overall, our MATE reliably surpasses other baselines by a significant margin across all datasets. In particular, MATE averagely outperforms the closest competitor on ENZYMES with 8% and COIL-DEL with 15%, demonstrating the impressive ability of our framework for partial label graph labeling.
- When the number of graph categories is large, such as COIL-DEL with 100 categories, the improvement brought from our MATE is more significant, owing to the balanced and unbiased pseudo-labels generated by our proposed masked optimal transport mechanism.
- By leveraging the mixup augmentation, the recent Graph Transplant outperforms other graph convolution methods and graph pooling methods in most cases. Besides, GraphSAGE performs better among graph convolution methods, suggesting that its architecture is more robust to ambiguous and noisy labels.

Table 2
Comparative results over five graph benchmarks (Part 1/3). Best in bold, second-best underlined.

Dataset	ENZYMES			Letter-High		
Methods	$q = 0.1$	$q = 0.3$	$q = 0.5$	$q = 0.1$	$q = 0.3$	$q = 0.5$
GCN	61.33 ± 2.85	48.44 ± 2.06	40.22 ± 2.93	50.09 ± 0.70	44.00 ± 1.08	35.94 ± 1.82
GAT	58.22 ± 3.03	49.11 ± 2.93	34.67 ± 3.87	73.39 ± 1.41	61.33 ± 3.48	53.04 ± 3.06
GIN	59.78 ± 4.58	47.11 ± 4.59	34.22 ± 1.78	55.83 ± 4.28	50.43 ± 1.92	35.59 ± 3.75
GraphSAGE	60.89 ± 1.09	47.33 ± 3.03	39.33 ± 3.11	78.20 ± 1.17	70.96 ± 1.48	60.35 ± 1.83
TopKPool	53.11 ± 4.12	44.22 ± 2.76	36.00 ± 4.80	67.07 ± 1.60	55.25 ± 2.74	43.83 ± 5.21
SAGPool	56.89 ± 5.37	46.67 ± 2.53	37.11 ± 5.00	67.42 ± 1.91	55.71 ± 4.71	39.30 ± 5.49
EdgePool	58.67 ± 2.67	51.11 ± 3.06	33.33 ± 1.99	70.49 ± 3.29	64.17 ± 2.44	55.36 ± 2.16
ASAP	60.89 ± 2.67	44.44 ± 3.06	31.56 ± 3.34	71.25 ± 1.44	65.04 ± 1.22	52.75 ± 4.41
TGNN	62.44 ± 3.01	53.33 ± 3.51	42.22 ± 4.39	78.55 ± 0.78	70.43 ± 0.97	59.83 ± 1.32
Graph Transplant	61.56 ± 2.86	51.78 ± 2.39	43.78 ± 3.41	80.75 ± 0.60	74.84 ± 1.44	66.78 ± 1.86
GraphCL	61.78 ± 1.51	54.22 ± 5.14	39.78 ± 5.09	78.43 ± 0.85	72.00 ± 2.01	62.49 ± 2.00
MATE (Ours)	64.67 ± 3.25	59.11 ± 2.67	48.89 ± 1.86	83.71 ± 1.61	81.39 ± 0.70	75.48 ± 2.80
Improvement	**3.57%**	**9.02%**	**11.67%**	**3.67%**	**8.75%**	**13.03%**

Table 3
Comparative results over five graph benchmarks (Part 2/3). Best in bold, second-best underlined. OOM denotes out-of-memory.

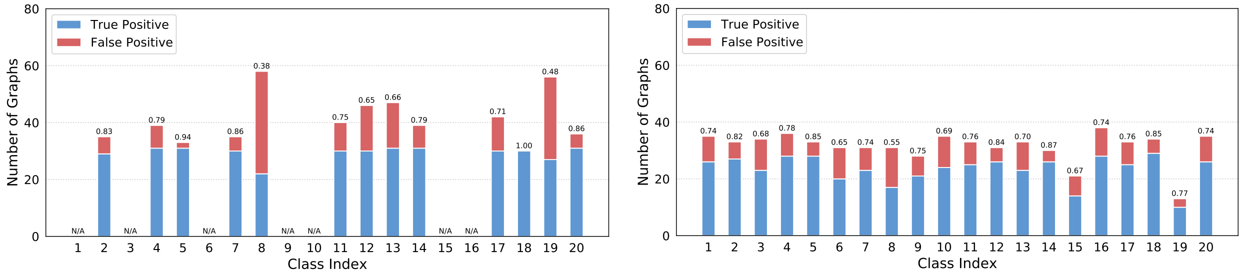
Dataset	CIFAR10			COIL-DEL		
Methods	$q = 0.1$	$q = 0.3$	$q = 0.5$	$q = 0.02$	$q = 0.05$	$q = 0.1$
GCN	47.18 ± 1.09	43.68 ± 0.68	41.35 ± 0.65	60.77 ± 1.71	50.43 ± 1.07	41.63 ± 1.74
GAT	57.56 ± 0.65	52.93 ± 1.22	48.54 ± 0.46	69.11 ± 2.86	59.77 ± 1.97	46.63 ± 1.54
GIN	47.29 ± 0.61	43.91 ± 0.45	41.24 ± 0.52	55.94 ± 1.69	46.23 ± 0.88	37.29 ± 1.04
GraphSAGE	57.22 ± 0.67	51.92 ± 0.26	47.44 ± 0.83	71.40 ± 2.15	58.91 ± 1.92	49.23 ± 1.90
TopKPool	55.26 ± 0.85	48.97 ± 1.24	42.87 ± 1.31	55.80 ± 4.86	44.83 ± 2.19	34.63 ± 2.08
SAGPool	54.23 ± 0.53	50.01 ± 0.68	45.16 ± 0.36	52.94 ± 2.59	41.89 ± 4.28	30.17 ± 1.85
EdgePool	55.09 ± 0.61	50.17 ± 0.64	45.90 ± 0.44	68.74 ± 1.85	56.74 ± 3.98	45.89 ± 1.30
ASAP	54.56 ± 0.66	50.10 ± 0.63	44.81 ± 1.57	59.03 ± 3.09	46.20 ± 4.08	34.94 ± 3.02
TGNN	OOM	OOM	OOM	70.49 ± 0.87	62.28 ± 1.05	50.20 ± 1.17
Graph Transplant	56.87 ± 1.28	53.79 ± 1.11	48.95 ± 1.47	80.09 ± 0.75	66.57 ± 1.60	57.11 ± 1.03
GraphCL	57.62 ± 0.56	53.57 ± 0.87	48.10 ± 0.61	78.83 ± 1.06	69.94 ± 2.31	60.17 ± 2.96
MATE (Ours)	59.17 ± 0.21	55.92 ± 0.38	53.31 ± 0.55	87.11 ± 0.91	81.12 ± 1.54	72.51 ± 1.84
Improvement	**2.69%**	**3.96%**	**8.91%**	**8.77%**	**15.99%**	**20.51%**

Table 4
Comparative results over five graph benchmarks (Part 3/3). Best in bold, second-best underlined.

Dataset	COLORS-3		
Methods	$q = 0.1$	$q = 0.3$	$q = 0.5$
GCN	90.34 ± 0.06	74.87 ± 0.25	60.67 ± 1.64
GAT	89.33 ± 1.42	71.83 ± 0.22	62.56 ± 3.54
GIN	63.01 ± 1.45	48.17 ± 0.44	41.00 ± 2.75
GraphSAGE	91.70 ± 2.18	71.24 ± 3.00	56.63 ± 5.81
TopKPool	82.35 ± 1.36	56.69 ± 3.58	33.49 ± 1.74
SAGPool	76.99 ± 4.39	59.91 ± 4.14	24.62 ± 0.32
EdgePool	87.47 ± 0.41	76.96 ± 0.13	62.31 ± 1.23
ASAP	77.84 ± 1.26	70.11 ± 0.54	62.47 ± 0.98
TGNN	93.16 ± 1.55	75.84 ± 1.81	63.95 ± 2.18
Graph Transplant	85.48 ± 0.89	74.66 ± 1.30	62.72 ± 2.37
GraphCL	92.71 ± 1.61	72.89 ± 1.24	61.55 ± 1.01
MATE (Ours)	96.44 ± 2.84	88.91 ± 2.11	69.02 ± 1.87
Improvement	**3.52%**	**17.23%**	**7.93%**

Table 5
Ablation study on Letter-High and COIL-DEL. MOT corresponds to masked optimal transport.

Ablation	Letter-High		COIL-DEL	
	$q = 0.1$	$q = 0.3$	$q = 0.02$	$q = 0.05$
MATE w/o MOT	82.38 ± 1.49	80.17 ± 0.60	86.29 ± 0.98	79.74 ± 1.62
MATE w/o PED	81.62 ± 1.58	79.36 ± 2.11	83.69 ± 1.56	78.26 ± 1.60
MATE w Fixed Thres.	82.84 ± 1.84	80.58 ± 0.81	86.43 ± 1.01	79.29 ± 2.18
MATE w/o $\mathcal{L}^{C}$	82.90 ± 1.30	80.29 ± 1.20	85.91 ± 1.57	79.89 ± 1.24
MATE w/o $\mathcal{L}^{\mathcal{W}}$	81.74 ± 2.19	80.46 ± 2.47	86.00 ± 1.44	80.17 ± 1.81
MATE w/o $\mathcal{L}^{\mathcal{U}}$	76.87 ± 2.54	71.30 ± 1.07	71.26 ± 2.32	56.57 ± 1.30
MATE w/o $\mathcal{L}_{clu}$	81.04 ± 1.51	79.88 ± 2.07	82.43 ± 1.85	75.31 ± 1.78
MATE (Full Model)	83.71 ± 1.61	81.39 ± 0.70	87.11 ± 0.91	81.12 ± 1.54



(a) Using vanilla pseudo-labeling.

(b) Using our pseudo-labeling.

Fig. 4. Number of graphs classified into the first 20 classes in COIL-DEL dataset ($q = 0.1$) using different pseudo-labeling strategies during training. (a) Vanilla pseudo-labeling leads to imbalanced assignments, with certain dominant classes overrepresented. (b) Our masked optimal transport approach yields a more class-balanced distribution of pseudo-labels.

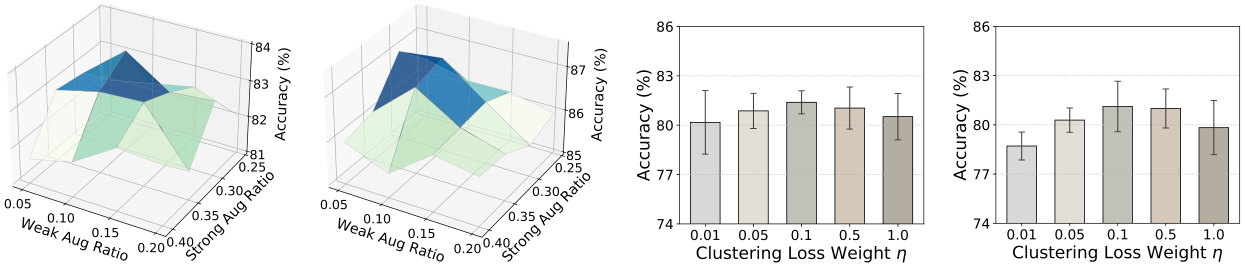
- The performance of all the methods drops when the degree of label ambiguity q increases, since the supervising signals become weaker and noisier. However, the accuracy of our MATE decreases more slowly with q growing, showing its remarkable robustness to label ambiguity. We deem that it benefits from PED-based uncertainty measurement against noisy pseudo labels.

4.3. Ablation study

Effect of Masked Optimal Transplant. We examine the effectiveness of the masked optimal transplant (MOT) mechanism by replacing the balance-optimized $\boldsymbol{Q}$ with the original prediction $\hat{\boldsymbol{P}}$ as the pseudo labels (i.e. MATE w/o MOT). As shown in Table 5, the model performance declines when the prediction lacks the optimization via the MOT, indicating more class-balanced pseudo-labels obtained from the MOT benefit to graph classification under the label ambiguity. Fig. 1 and Fig. 4 further illustrate the effectiveness of our pseudo-labeling strategy using masked optimal transport. Vanilla pseudo-labeling means assigning pseudo-labels by selecting the category assigned the greatest predicted likelihood for each instance, while our pseudo-labeling mechanism utilizes masked optimal transport to refine assignments for graph samples under the label ambiguity. As shown in the figures, our method generates significantly more class-balanced pseudo-label distributions during training, compared to the biased assignments generated by vanilla pseudo-labeling. This highlights the effectiveness of removing class biases through our proposed refinement strategy, and supports the claim in Section 3.4 that our masked optimal transport leads to more class-balanced pseudo-labels, aligning better with the uniform prior.

Effect of Dynamic Selection with PED. To justify the benefits of dynamic selection with partial energy discrepancy, we compare MATE with two variants: 1) MATE w/o PED, which utilizes a simple confidence score $\max_{c \in Y_i} \left\{ \hat{\boldsymbol{p}}_i[c] \right\}$ to replace the PED score; 2) MATE w fixed thres., which removes momentum updating of the PED threshold. As shown in Table 5, our proposed PED score outperforms the simple confidence score, since the latter ignores the distributions other than the maximum value, thus cannot capture distribution information inside and outside the candidate set sufficiently. The dynamic threshold further improves the performance due to modeling the diverse optimization difficulty of different graphs.

Effect of Sets with Different Levels of Confidence. We evaluate the effect of different sets by removing the loss for each set respectively. From Table 5, we can see that $\mathcal{L}^{C}$ is beneficial since it can sharpen the predicted distributions of confident samples, while $\mathcal{L}^{\mathcal{W}}$ also improves the performance by leveraging the optimized pseudo labels to supervise samples at decision boundaries in a soft manner. Particularly, the non-candidate loss $\mathcal{L}^{\mathcal{U}}$ for the unconfident set is crucial to our proposed framework MATE owing to concentrating the predicted distributions of insufficiently trained samples into the candidate set.



(a) Letter-High ($q = 0.1$) (b) COIL-DEL ($q = 0.02$) (c) Letter-High ($q = 0.3$) (d) COIL-DEL ($q = 0.05$)

Fig. 5. (a)(b) Analysis of graph augmentation ratios for weak and strong views on Letter-High and COIL-DEL. (c)(d) Analysis of balance weight η for the clustering loss on Letter-High and COIL-DEL.

Table 6
Comparative results on CIFAR10 with hierarchical label noise. Best in bold, second-best underlined. q reflects the degree of label ambiguity.

Dataset	CIFAR10		
Methods	$q = 0.2$	$q = 0.5$	$q = 0.8$
GCN	47.27 ± 0.28	43.17 ± 0.52	34.25 ± 1.47
GAT	57.01 ± 0.81	51.25 ± 0.78	41.05 ± 0.29
GIN	47.26 ± 0.33	42.72 ± 0.62	33.55 ± 0.74
GraphSAGE	56.93 ± 0.34	49.73 ± 0.54	39.16 ± 0.90
TopKPool	51.38 ± 1.61	45.43 ± 1.78	37.32 ± 0.99
SAGPool	50.38 ± 0.90	45.68 ± 0.57	37.94 ± 1.12
EdgePool	55.42 ± 0.51	49.48 ± 0.76	39.36 ± 0.63
ASAP	53.33 ± 0.25	48.21 ± 0.49	38.40 ± 0.53
TGNN	OOM	OOM	OOM
Graph Transplant	56.21 ± 1.20	49.93 ± 2.50	41.99 ± 0.79
GraphCL	57.25 ± 0.79	50.18 ± 0.54	39.17 ± 0.48
MATE (Ours)	58.76 ± 0.94	55.14 ± 0.54	48.54 ± 0.86
Improvement	**2.64%**	**7.59%**	**15.60%**

Effect of Learning to Cluster. As we can see in Table 5, the model performance declines obviously when the clustering loss $\mathcal{L}_{clu}$ is removed (i.e. MATE w/o $\mathcal{L}_{clu}$), demonstrating the advantage of learning discriminative graph representations via clustering guided by balanced pseudo labels.

4.4. *Hyper-parameter study*

Analysis of Augmentation Strength of Weak and Strong Views. We first investigate the effect of graph augmentation ratios ρ_w and ρ_s for weak and strong views. As shown in Fig. 5(a,b), the accuracy achieves the peak when $\rho_w = 0.1$ and $\rho_s = 0.3$ on both datasets, since the moderate difference of augmentation strength between two views facilitates exploring diverse semantics of graphs. Besides, both too-small ratios and too-large ratios hurt the performance, because the former reduces the discriminative ability of the model, while the latter distorts the original semantics of graphs.

Analysis of Clustering Loss Weight. Next, we analyze the influence of the balance weight η for the clustering loss $\mathcal{L}_{clu}$. It can be observed from Fig. 5(c,d) that the model performance improves when η increases from 0.01 to 0.1, while declining slightly when η exceeds 0.1. It demonstrates that appropriately increasing the contribution of clustering is advantageous to partial label graph learning, since the clustering loss can encourage more discriminative representations and serve as a regularization term against label noise.

4.5. *Further study on semantic similarity between labels*

In practical scenarios, the semantic associations among labels often lead to noisy candidate labels that are more closely connected to the ground-truth label compared to non-candidate labels. This characteristic increases the difficulty of distinguishing between these semantically related labels, posing a significant challenge for partial label graph learning algorithms.

To better capture real-world label ambiguity, we utilize the super-class and sub-class relationships in the CIFAR10 dataset to generate label candidate sets, where super-class 1 (vehicles) includes automobile, ship, airplane, and truck (but not pickup truck), and super-class 2 (animals) includes cat, dog, frog, deer, bird, and horse. In this setting, only the negative labels under the same super-class can be flipped with a probability q to enter the candidate label set. Therefore, the label candidates for each sample are

Table 7
Comparative results on two graph datasets with competitive label noise. Best in bold, second-best underlined. Avg.#CL denotes the average number of candidate labels.

Dataset	Letter-High			CIFAR10		
Methods	Avg.#CL = 3	Avg.#CL = 4	Avg.#CL = 5	Avg.#CL = 3	Avg.#CL = 4	Avg.#CL = 5
GCN	50.84 ± 1.12	46.78 ± 1.40	40.58 ± 2.34	44.84 ± 0.60	42.66 ± 0.64	38.53 ± 0.68
GAT	68.29 ± 1.25	59.83 ± 3.54	56.46 ± 1.08	54.00 ± 0.38	50.46±0.46	44.06 ± 0.90
GIN	58.67 ± 2.41	51.94 ± 0.87	45.62 ± 3.80	45.69 ± 0.42	42.99 ± 0.71	37.95 ± 0.80
GraphSAGE	73.28±1.28	66.14±3.58	60.52 ± 1.40	52.77 ± 0.96	49.44 ± 0.48	42.94 ± 1.06
TopKPool	64.64 ± 1.57	57.04 ± 2.79	49.74 ± 5.89	49.27 ± 1.23	45.83 ± 0.61	41.80 ± 0.71
SAGPool	67.42±1.77	64.35 ± 1.71	56.12 ± 2.05	48.33 ± 0.56	45.98 ± 1.31	42.53 ± 0.70
EdgePool	70.14 ± 1.54	63.77 ± 3.56	57.97 ± 2.37	52.18 ± 0.73	47.49 ± 0.89	43.58 ± 0.76
ASAP	68.29 ± 2.15	63.42 ± 2.10	54.49 ± 3.41	51.19 ± 0.52	47.25 ± 0.38	43.95 ± 0.92
TGNN	73.56 ± 1.92	68.35 ± 1.73	63.13 ± 1.28	OOM	OOM	OOM
Graph Transplant	78.96 ± 1.30	73.85 ± 1.69	69.45 ± 1.21	53.71 ± 1.67	51.64 ± 0.82	49.96 ± 0.75
GraphCL	74.61 ± 1.06	66.09 ± 2.80	62.55 ± 4.26	53.97 ± 0.54	50.10 ± 0.63	45.23 ± 0.71
MATE (Ours)	81.80 ± 1.81	79.19 ± 1.56	74.66 ± 1.61	55.92±**0.45**	54.82 ± 0.36	52.77 ± 0.28
Improvement	**3.60%**	**7.23%**	**7.50%**	**3.56%**	**6.16%**	**5.62%**

Table 8
Effect of various graph augmentation strategies.

Augmentation Strategies	ENZYMES $q = 0.1$	Letter-High $q = 0.1$	COIL-DEL $q = 0.02$	Avg.
Edge Perturbation	65.56 ± 4.22	79.65 ± 1.60	85.03 ± 0.59	76.75
Node Dropping	63.11 ± 2.67	79.48 ± 2.73	86.29 ± 1.02	76.29
Attribute Masking	63.78 ± 4.59	81.74 ± 2.11	85.97 ± 1.00	77.16
Subgraph	58.89 ± 3.85	83.48 ± 1.63	73.06 ± 1.77	71.81
Random Strategy	64.67 ± 3.25	83.71 ± 1.61	87.11 ± 0.91	**78.50**

semantically related. For example, within the "animals" super-class, subclasses like "dog," "cat," and "horse" form the candidate set, avoiding irrelevant labels from the "vehicles" category. This setup better captures the nature of label uncertainty that arises in real-world applications. The results are shown in Table 6. From the results, we can find that our MATE outperforms all the competitive baselines in this more realistic setting, demonstrating the superiority of our MATE for partial label graph learning with hierarchical label noise.

For datasets without a known hierarchical label structure, or in scenarios where we aim to explore intrinsic label correlations beyond predefined super-class and sub-class relationships, we adopt a competitive label noise setting following [57]. Specifically, a graph neural network is trained on the clean dataset to predict category probabilities for each instance, and negative labels with Top-K predicted probabilities are selected as noisy candidates randomly. We conduct experiments on the Letter-High and CIFAR10 datasets, and set K to 6. The results are shown in Table 7, where the average number of candidate labels (Avg.#CL) reflects the degree of label noise. We can observe from the results that MATE consistently outperforms baseline methods, showcasing its robustness and adaptability in handling semantic similarity in the noisy candidate labels, which is more realistic and challenging.

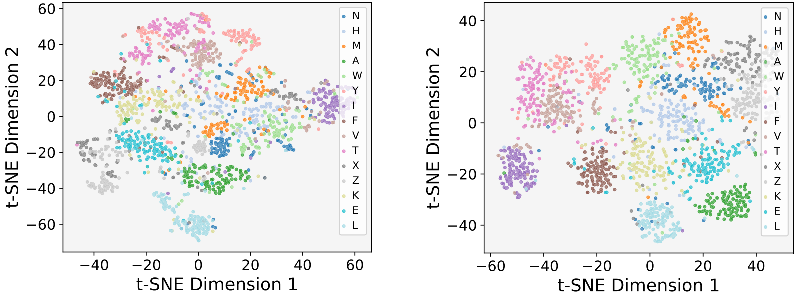
4.6. Effect of graph augmentation strategies

We look into the effect of various graph augmentation strategies including edge perturbation, node dropping, attribute masking, subgraph, and random selecting from these four strategies. As shown in Table 8, different graph augmentation strategies contribute differently to varied datasets. For instance, the subgraph strategy is great for Letter-High, but performs poorly on ENZYMES and COIL-DEL datasets. This inspires us that developing learnable and adaptive graph augmentation strategies may further improve the capability and robustness of the model. Besides, the random strategy which randomly selects one of the four strategies for each graph, shows stable and superior performance across varied datasets.

4.7. Visualization and case study

Moreover, we look into the discriminative ability of our proposed MATE by visualizing the learned graph representations with t-SNE [43]. Fig. 6 shows the results of our MATE compared to the best baseline Graph Transplant. As we can see, even in cases where the degree of label ambiguity is high ($q = 0.5$), MATE exhibits relatively clear class boundaries and fewer samples scattered to other classes, compared to Graph Transplant. It indicates that our MATE generates more discriminative representations and suffers less from label ambiguity, which explains its superior performance on partial label graph learning.

We further analyze the effectiveness of our proposed MATE through case studies on the COIL-DEL dataset. As shown in Fig. 7, four examples are presented with predictions from our MATE and the competitive baseline GraphCL. While GraphCL fails to classify



(a) Graph Transplant

(b) MATE (Ours)

Fig. 6. t-SNE visualization of learned graph representations for Letter-High ($q = 0.5$).

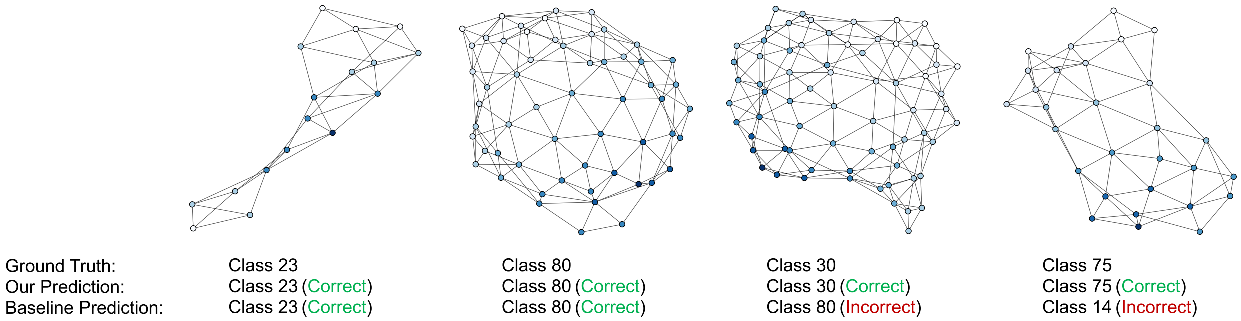


Fig. 7. Visualization of four graph samples with our predictions and baseline predictions on COIL-DEL.

two cases correctly, our MATE achieves accurate predictions for all four cases. As we can see, the third graph structurally resembles the second one. This likely causes GraphCL to misclassify them to the same category, whereas MATE successfully distinguishes subtle differences and provides correct predictions. These results highlight the capability and robustness of our MATE in handling challenging graph-structured data under label noise and ambiguity.

5. Conclusion

This paper studies a practical yet underexplored challenge of partial label graph learning, and introduces a new approach called MATE to tackle it. The core of the proposed MATE is to improve the quality of graph assignments from the perspectives of class balancing and uncertainty mining. On the one hand, we formulate an optimal transport objective for the class balance of graph assignments. On the other hand, we introduce partial energy discrepancy for uncertainty measurement and a dynamic selection strategy using historical scores. We also utilize learning to cluster for the discriminability of graph representations. Experimental results across multiple real-world graph benchmarks validate the advantages of our framework MATE over a wide range of methods. For future work, we plan to apply our framework to more visual and natural language tasks and scenarios, such as text classification and tag-assisted classification.

Limitation

The benefits of masked optimal transport (OT) and class-balanced assignments depend on the ground truth label distribution. The masked OT approach is most effective when the ground truth label distribution is class-balanced or approximately so, which is common in various training datasets and real-world applications [32,54]. We deem that extending the masked OT framework to accommodate class-imbalanced ground truth distributions is an interesting direction for future work.

CRediT authorship contribution statement

Yiyang Gu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Binqi Chen:** Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Zihao Chen:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Ziyue Qiao:** Writing – review & editing, Formal analysis, Conceptualization. **Xiao Luo:** Writing – original draft, Methodology, Investigation, Conceptualization. **Junyu Luo:** Methodology, Investigation, Conceptualization. **Zhiping Xiao:** Writing – review & editing, Visualization, Conceptualization. **Wei Ju:** Methodology, Investigation, Data curation. **Ming Zhang:** Conceptualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This paper is partially supported by the National Natural Science Foundation of China with Grant Number 62276002.

Appendix A. Proof of theorem

Proof of Theorem 3.1. Let $\hat{\boldsymbol{X}}$ be the projection of $\boldsymbol{X}$ onto $\prod(\frac{1}{C}\mathbf{1}_C, \frac{1}{B}\mathbf{1}_B)$, i.e., $\hat{\boldsymbol{X}}$ is the solution of

$$\begin{aligned}&\min_{\hat{\boldsymbol{X}}} \quad \frac{1}{2}\left\|\hat{\boldsymbol{X}} - \boldsymbol{X}\right\|_F^2\\ &\text{s.t.} \quad \hat{\boldsymbol{X}}\mathbf{1}_B = \frac{1}{C}\mathbf{1}_C, \quad \hat{\boldsymbol{X}}^T\mathbf{1}_C = \frac{1}{B}\mathbf{1}_B.\end{aligned} \tag{34}$$

Using the Lagrange multiplier, we know that

$$\hat{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{\alpha}\mathbf{1}_B^T, \tag{35}$$

where $\alpha = (\alpha_1, \dots, \alpha_C)^T$ and $\alpha_i = \frac{1}{B}\left(\frac{1}{B}\sum_{j=1}^B \hat{\boldsymbol{P}}_{ij} - \frac{1}{C}\right)$.

Next, let $\hat{\boldsymbol{Q}}$ be a solution of the Kantorovitch's problem

$$\max_{Q \in \prod(\frac{1}{C}\mathbf{1}_C, \frac{1}{B}\mathbf{1}_B)} \sum_{i,j} \boldsymbol{Q}_{ij}\hat{\boldsymbol{P}}_{ij}. \tag{36}$$

Then, we have

$$\left\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{P}} \right\rangle \leq \left\langle \hat{\boldsymbol{Q}}, \hat{\boldsymbol{P}} \right\rangle. \tag{37}$$

From direct computation, we know that

$$= \left\langle \hat{\boldsymbol{X}} - \boldsymbol{X}, \hat{\boldsymbol{P}} \right\rangle = C_{\boldsymbol{X}}. \tag{38}$$

Using standard OT theory, we know that as $\lambda \to \infty$ in Eq. (5), $\boldsymbol{Q}$ converges to the solution of problem (36) with maximal entropy, $\boldsymbol{Q}_\infty$. Therefore, for large enough λ, we know that

$$\|\boldsymbol{Q} - \boldsymbol{Q}_\infty\|_F \leq \varepsilon / \left\|\hat{\boldsymbol{P}}\right\|_F. \tag{39}$$

Using Cauchy's inequality, we have

$$\left|\left\langle \boldsymbol{Q} - \boldsymbol{Q}_\infty, \hat{\boldsymbol{P}} \right\rangle\right| \leq \varepsilon. \tag{40}$$

Combining Eq. (37)-(40) and take $\hat{\boldsymbol{Q}} = \boldsymbol{Q}_\infty$ in Eq. (37), we get

$$\left\langle \boldsymbol{X}, \hat{\boldsymbol{P}} \right\rangle + C_{\boldsymbol{X}} \leq \left\langle \boldsymbol{Q}, \hat{\boldsymbol{P}} \right\rangle + \varepsilon. \tag{41}$$

Note that every element of $\boldsymbol{X}$ and $\boldsymbol{Q}$ belongs to $[0, 1]$, so we have

$$\begin{aligned}\left\langle \boldsymbol{X} - \boldsymbol{Q}, \hat{\boldsymbol{P}} - \boldsymbol{L} \right\rangle &= \sum_{i,j} \left(\boldsymbol{X}_{ij} - \boldsymbol{Q}_{ij}\right)\left(\hat{\boldsymbol{P}}_{ij} - l_{ij}\right)\\ &\leq \varepsilon \sum_{i,j} |\boldsymbol{X}_{ij} - \boldsymbol{Q}_{ij}| \leq 2\varepsilon.\end{aligned} \tag{42}$$

Hence,

$$\langle \boldsymbol{X}, \boldsymbol{L} \rangle + C_{\boldsymbol{X}} \leq \langle \boldsymbol{Q}, \boldsymbol{L} \rangle + 3\varepsilon. \tag{43}$$

Data availability

Data will be made available on request.

References

[1] J. Baek, M. Kang, S.J. Hwang, Accurate learning of graph representations with graph multiset pooling, in: International Conference on Learning Representations, 2020.

[2] A.D. Becke, Perspective: fifty years of density-functional theory in chemical physics, J. Chem. Phys. 140 (2014) 18A301.

[3] D. Cai, W. Lam, Graph transformer for graph-to-sequence learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 7464–7471.

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, in: Proceedings of the Conference on Neural Information Processing Systems, 2020, pp. 9912–9924.

[5] D. Chen, L. O'Bray, K. Borgwardt, Structure-aware transformer for graph representation learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 3469–3489.

[6] L. Chen, Y. Lou, J. He, T. Bai, M. Deng, Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16134–16143.

[7] H. Chermette, Chemical reactivity indexes in density functional theory, J. Comput. Chem. 20 (1999) 129–154.

[8] T. Cour, B. Sapp, B. Taskar, Learning from partial labels, J. Mach. Learn. Res. 12 (2011) 1501–1536.

[9] M. Cuturi, Sinkhorn distances: lightspeed computation of optimal transport, Adv. Neural Inf. Process. Syst. 26 (2013).

[10] F. Diehl, Edge contraction pooling for graph neural networks, arXiv preprint, arXiv:1905.10990, 2019.

[11] V.P. Dwivedi, C.K. Joshi, T. Laurent, Y. Bengio, X. Bresson, Benchmarking graph neural networks, arXiv preprint, arXiv:2003.00982, 2020.

[12] E. Eiben, R. Ganian, T. Hamm, S. Ordyniak, Parameterized complexity of envy-free resource allocation in social networks, Artif. Intell. 315 (2023) 103826.

[13] H. Fang, X. Li, J. Zhang, Integrating social influence modeling and user modeling for trust prediction in signed networks, Artif. Intell. 302 (2022) 103628.

[14] L. Feng, B. An, Partial label learning by semantic difference maximization, in: IJCAI, 2019, pp. 2294–2300.

[15] H. Gao, S. Ji, Graph u-nets, in: International Conference on Machine Learning, PMLR, 2019, pp. 2083–2092.

[16] L. Ge, M. Fang, H. Li, B. Chen, Label correlation for partial label learning, J. Syst. Eng. Electron. 33 (2022) 1043–1051.

[17] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.

[18] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, D. Tao, A regularization approach for instance-based superset label learning, IEEE Trans. Cybern. 48 (2017) 967–978.

[19] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).

[20] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[21] R. He, Z. Han, X. Lu, Y. Yin, Ronf: reliable outlier synthesis under noisy feature space for out-of-distribution detection, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4242–4251.

[22] R. He, Z. Han, X. Lu, Y. Yin, Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14585–14594.

[23] S. He, L. Feng, F. Lv, W. Li, G. Yang, Partial label learning with semantic label representations, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 545–553.

[24] P. Hu, Z. Lin, W. Pan, Q. Yang, X. Peng, Z. Ming, Privacy-preserving graph convolution network for federated item recommendation, Artif. Intell. 324 (2023) 103996.

[25] E. Hüllermeier, J. Beringer, Learning from ambiguously labeled examples, in: Advances in Intelligent Data Analysis VI: 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, in: Proceedings, vol. 6, Springer, 2005, pp. 168–179.

[26] W. Ju, X. Luo, M. Qu, Y. Wang, C. Chen, M. Deng, X.S. Hua, M. Zhang, Tgnn: a joint semi-supervised framework for graph-level classification, arXiv preprint, arXiv:2304.11688, 2023.

[27] N. Karim, M.N. Rizve, N. Rahnavard, A. Mian, M. Shah, Unicon: combating label noise through uniform selection and contrastive learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9676–9686.

[28] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, Conference Track Proceedings, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, 2015.

[29] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the International Conference on Learning Representations, 2017.

[30] P.A. Knight, The Sinkhorn–Knopp algorithm: convergence and applications, SIAM J. Matrix Anal. Appl. 30 (2008) 261–275.

[31] B. Knyazev, G.W. Taylor, M. Amer, Understanding attention and generalization in graph neural networks, Adv. Neural Inf. Process. Syst. 32 (2019).

[32] A. Krizhevsky, et al., Learning multiple layers of features from tiny images, 2009.

[33] Y. Lan, Y. Zhang, Y. Qu, C. Wang, C. Li, J. Cai, Y. Xie, Z. Wu, Weakly supervised 3d segmentation via receptive-driven pseudo label consistency and structural consistency, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 1222–1230.

[34] D. Lee, S. Kim, S. Lee, C. Park, H. Yu, Learnable structural semantic readout for graph classification, in: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 1180–1185.

[35] J. Lee, I. Lee, J. Kang, Self-attention graph pooling, in: International Conference on Machine Learning, PMLR, 2019, pp. 3734–3743.

[36] Y. Li, H. Han, S. Shan, X. Chen, Disc: learning from noisy labels via dynamic instance-specific selection and correction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24070–24079.

[37] Y. Liang, F. Meng, Y. Zhang, Y. Chen, J. Xu, J. Zhou, Emotional conversation generation with heterogeneous graph neural network, Artif. Intell. 308 (2022) 103714.

[38] W. Liu, X. Wang, J.D. Owens, Y. Li, Energy-based out-of-distribution detection, arXiv:2010.03759, 2021.

[39] X. Luo, W. Ju, M. Qu, C. Chen, M. Deng, X.S. Hua, M. Zhang, Dualgraph: improving semi-supervised graph classification via dual contrastive learning, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 699–712.

[40] X. Luo, Y. Zhao, Y. Qin, W. Ju, M. Zhang, Towards semi-supervised universal graph classification, IEEE Trans. Knowl. Data Eng. 36 (2023) 416–428.

[41] C. Zhang, H. Ren, X. He, P2OT: Progressive Partial Optimal Transport for Deep Imbalanced Clustering, in: The Twelfth International Conference on Learning Representations, 2024.

[42] G. Lyu, S. Feng, T. Wang, C. Lang, A self-paced regularization framework for partial-label learning, IEEE Trans. Cybern. 52 (2020) 899–911.

[43] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008).

[44] J. Park, H. Shim, E. Yang, Graph transplant: node saliency-guided graph mixup with local structure preservation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 7966–7974.

[45] E. Ranjan, S. Sanyal, P. Talukdar, Asap: adaptive structure aware pooling for learning hierarchical graph representations, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 5470–5477.

[46] K. Riesen, H. Bunke, Iam graph database repository for graph based pattern recognition and machine learning, in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 2008, pp. 287–297.

[47] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, D. Schomburg, Brenda, the enzyme database: updates and major new developments, Nucleic Acids Res. 32 (2004) D431–D433.

[48] R. Sinkhorn, P. Knopp, Concerning nonnegative matrices and doubly stochastic matrices, Pac. J. Math. 21 (1967) 343–348.

[49] N. Liu, X. Wang, D. Bo, C. Shi, J. Pei, Revisiting graph contrastive learning from the perspective of graph spectrum, Adv. Neural Inf. Process. Syst. 35 (2022) 2972–2983.

[50] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint, arXiv:1710.10903, 2017.

[51] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, J. Zhao, Pico: contrastive label disambiguation for partial label learning, in: Proceedings of the International Conference on Learning Representations, 2022.

[52] W. Wang, M.L. Zhang, Partial label learning with discrimination augmentation, in: Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2022, pp. 1920–1928.

[53] Y. Gu, Z. Chen, Y. Qin, Z. Mao, Z. Xiao, W. Ju, C. Chen, X.S. Hua, Y. Wang, X. Luo, M. Zhang, DEER: Distribution divergence-based graph contrast for partial label learning on graphs, IEEE Trans. Multimed. (2024).

[54] Q. Xie, M.T. Luong, E. Hovy, Q.V. Le, Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.

[55] Z. Xiong, S. Liu, F. Huang, Z. Wang, X. Liu, Z. Zhang, W. Zhang, Multi-relational contrastive learning graph neural network for drug-drug interaction event prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 5339–5347.

[56] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2018.

[57] Y. Yan, Y. Guo, Mutual partial label learning with competitive label noise, in: The Eleventh International Conference on Learning Representations, 2023.

[58] X. Yang, M. Yan, S. Pan, X. Ye, D. Fan, Simple and efficient heterogeneous graph neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 10816–10824.

[59] K. Yao, J. Liang, J. Liang, M. Li, F. Cao, Multi-view graph convolutional networks with attention mechanism, Artif. Intell. 307 (2022) 103708.

[60] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, in: Advances in Neural Information Processing Systems, 2020.

[61] S. Yun, M. Jeong, R. Kim, J. Kang, H.J. Kim, Graph transformer networks, in: Proceedings of the Conference on Neural Information Processing Systems, 2019.

[62] D. Zeng, W. Liu, W. Chen, L. Zhou, M. Zhang, H. Qu, Substructure aware graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 11129–11137.

[63] M. Zeng, F. Zhang, F.X. Wu, Y. Li, J. Wang, M. Li, Protein–protein interaction site prediction through combining local and global features with deep neural networks, Bioinformatics 36 (2020) 1114–1120.

[64] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, T. Shinozaki, Flexmatch: boosting semi-supervised learning with curriculum pseudo labeling, Adv. Neural Inf. Process. Syst. 34 (2021) 18408–18419.

[65] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Adv. Neural Inf. Process. Syst. 33 (2020) 18661–18673.

[66] Q. Zhang, S. Pei, Q. Yang, C. Zhang, N.V. Chawla, X. Zhang, Cross-domain few-shot graph classification with a reinforced task coordinator, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 4893–4901.

[67] T. Zhang, Y. Wang, Z. Cui, C. Zhou, B. Cui, H. Huang, J. Yang, Deep Wasserstein graph discriminant learning for graph classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 10914–10922.

[68] Y. Zhou, J. He, H. Gu, Partial label learning via Gaussian processes, IEEE Trans. Cybern. 47 (2016) 4443–4450.