PSEUDO-EDGE: SEMI-SUPERVISED LINK PREDICTION WITH GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pseudo-labeling (Lee et al., 2013) is one of the powerful Semi-Supervised Learning (SSL) approaches, which generates confident pseudo-labels of unlabeled data and leverages them for training. Recently, pseudo-labeling has been further extended to Graph Neural networks (GNNs) to address the data sparsity problem due to the nature of graph-structured data. Despite their success in the graph domain, they have been mainly designed for node-level tasks by utilizing node-level algorithms (e.g., Label Propagation) for pseudo-labeling, which can not be directly applied to the link prediction task. Besides, existing works for link prediction only use given edges as positively-labeled data, and there have been no attempts to leverage non-visible edges for training a model in a semi-supervised manner. To address these limitations, we revisit the link prediction task in a semi-supervised fashion and propose a novel pseudo-labeling framework, *Pseudo-Edge*, that generates qualified pseudo-labels in consideration of graph structures and harnesses them for the link prediction. Specifically, our framework constructs distance-based potential edge candidates and carefully selects pseudo-labels through our relation-aware pseudo-labels generation, which reflects the comparative superiority of each unlabeled edge over its local neighborhoods in graphs. Also, we propose uncertainty-aware pseudo-labels generation that can effectively filter out over-confident samples when the model overfits to specific graph structures. Extensive experiments show that our method achieved remarkable performance across five link prediction benchmark datasets and GNN architectures, compared to state-of-the-art GNN-based semi/self-supervised models.

1 INTRODUCTION

Deep neural networks (DNNs) have shown impressive performance across diverse domains (He et al., 2015; Silver et al., 2016; Mathis et al., 2018). However, this success usually requires large-scale labeled datasets, which are typically expensive to obtain. To overcome this challenge, Semi-Supervised Learning (SSL) (Chapelle et al., 2009) has been widely adopted in learning algorithms to utilize the vast amount of unlabeled data, given a small mount of labeled data. (Berthelot et al., 2019; Sohn et al., 2020; Lee et al., 2013; Tarvainen & Valpola, 2017; Zhu & Ghahramani, 2002) Pseudo-labeling (Lee et al., 2013), one of the powerful SSL approaches, generates pseudo-labels of unlabeled data by using a model trained on the labeled dataset and leverages these pseudo-labels for training. Recently, pseudo-labeling has been actively studied in the direction of combining with other SSL approaches (Xie et al., 2020a; Sohn et al., 2020; Pham et al., 2021) and utilizing advanced thresholding techniques (Zhang et al., 2021; Wang et al., 2022) in computer vision.

The application of pseudo-labeling has been further extended to graph neural networks (GNNs) to solve the scarcity of labeled data. As most graph datasets possess incompleteness, i.e. invisible ground-truth nodes/edges, the performance significantly decreases when the network prediction relies on sparsely labeled graph. To handle this issue, several works (Li et al., 2018; Sun et al., 2020; Wang et al., 2021; Dong et al., 2021) employ pseudo-labeling to leverage unlabeled nodes for training GNNs and achieve promising performance on the node classification task. However, they have focused only on the node classification not the link prediction task. Specifically, their pseudo-labeling methods are designed to utilize node-level algorithms (e.g., label propagation, random walk, k-means) for pseudo-labels selection, which can not be directly applied to the link prediction task.

In addition, existing works for link prediction only use given edges as positively-labeled data, and there is no attempt to leverage non-visible edges for training a model in a semi-supervised manner.

In this work, we revisit the link prediction task in a semi-supervised scenario and propose a novel pseudo-labeling framework, *Pseudo-Edge*, for semi-supervised link prediction by leveraging unlabeled edges. To the best of our knowledge, this is the first approach that utilizes unlabeled edges as pseudo-labels for link prediction. Our *Pseudo-Edge* generates qualified pseudo-labels of unseen edges considering graph structures and harnesses the pseudo-labels for training GNNs for link prediction. Specifically, *Pseudo-Edge* first constructs distance-based potential edge candidates and carefully selects pseudo-labels through our proposed *relation*-aware pseudo-labels generation that reflects the comparative superiority of each unlabeled edge over its local neighborhoods in graphs. Additionally, motivated from Rizve et al. (2021), we propose *uncertainty*-aware pseudo-labels generation considering uncertainty in graph structures, which can effectively filter out over-confident samples when the model overfits to specific graph structures. Our comprehensive experiments show that *Pseudo-Edge* achieves strong performance on three standard GNNs over six benchmark datasets for link prediction compared to state-of-the-art GNN-based semi/self-supervised learning models.

Our **contributions** are as follows:

- We revisit the link prediction task with GNNs in a semi-supervised scenario and propose a new training paradigm for link prediction leveraging unlabeled edges for link prediction.
- We propose a novel GNN-based pseudo- labeling framework for link prediction, *Pseudo-Edge*, that generates qualified pseudo-labels of unseen edges considering graph structures and harnesses the pseudo-labels for training GNNs for link prediction.
- We validate the effectiveness of the *Pseudo-Edge* with extensive experimental results that our *Pseudo-Edge* achieves strong performance on three standard GNNs over five graph datasets for link prediction compared to state-of-the-art approaches for semi-supervised and self-supervised learning.

2 Related Works

2.1 Pseudo-Labeling

Pseudo-labeling has been widely adopted to address semi-supervised learning in a supervised manner by assigning unlabeled data with predicted labels. Lee et al. (2013) treat a class which has the highest predicted probability on each unlabeled sample as a pseudo-label and Rosenberg et al. (2005) use this same strategy only when the prediction confidence is higher than the fixed threshold to prevent error propagation by wrongly assigned pseudo-labels. Iscen et al. (2019) assign pseudo-labels based on a nearest-neighbor graph in the feature space. MixMatch (Berthelot et al., 2019), UDA (Xie et al., 2020a), and NoisyStudent (Xie et al., 2020b) produce pseudo-labels boosting model robustness with consistency regularization methods such as MixUp (Zhang et al., 2018) and adversarial perturbations (Miyato et al., 2018). ReMixMatch (Berthelot et al., 2020), FixMatch (Sohn et al., 2020), and Meta Pseudo-Labels (Pham et al., 2021) utilize the predictions on weak augmentations of input samples as pseudo-labels for strong augmentations of the same samples. FlexMatch (Zhang et al., 2021), FreeMatch (Wang et al., 2022), and Curriculum Labeling (Cascante-Bonilla et al., 2021) employ adaptive thresholding depending on the learning status of models to tackle the limitation of the fixed threshold of pseudo-labels. In this work, we also present a novel adaptive thresholding method that considers the relation between data instances of graph domains.

2.2 Pseudo-Labeling in Graph Neural Networks

Pseudo-labeling has also attracted considerable attention in GNNs which suffer from severe data sparsity problem due to the nature of graph structure data. Li et al. (2018) select nearest-neighbor unlabeled nodes of labeled nodes using a random walk model and assign the unlabeled nodes with the labeled nodes' class. M3S (Sun et al., 2020) assigns pseudo-labels for unlabeled data by using the predictions of GNN models and DeepCluster (Caron et al., 2018). PTA (Dong et al., 2021) uses the Label Propagation algorithm to propagate the known labels along the graph to generate pseudo-labels for the unlabeled nodes. CaGCN (Wang et al., 2021) applies the confidence calibration to the predictions to produce more reliable pseudo-labels. However, most pseudo-labeling studies for

GNNs have focused only on node classification rather than link prediction. On the other hand, we propose a pseudo-labeling framework for link prediction by leveraging unlabeled edges as pseudo-labels for training GNNs.

3 PRELIMINARIES

3.1 PSEUDO-LABELING FOR SEMI-SUPERVISED LEARNING

Pseudo-labeling (Lee et al., 2013) is one of the most widely adopted methods in semi-supervised learning, which generates pseudo-labels of unlabeled data by using a model trained on the labeled dataset and leverages the pseudo-labels for training. Specifically, suppose we have an unlabeled dataset $\mathcal{U} = \{x^{(i)}\}_{i=1}^{U}$ and a model f_θ trained on a labeled dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $x^{(i)}$ is the input instance and $y^{(i)} = [y_1^{(i)}, ..., y_C^{(i)}] \subseteq \{0, 1\}^C$ is the corresponding label with C classes. Then, unlabeled instances $\tilde{x}^{(i)}$ whose prediction scores for a certain class are above the predefined threshold τ are *pseudo-labeled* as $\tilde{y}^{(i)}$ and added to the labeled dataset $\mathcal{D}$ as follows:

$$\begin{aligned}\mathcal{D} &= \mathcal{D} \cup \tilde{\mathcal{D}}, \\ \tilde{\mathcal{D}} &= \{(\tilde{x}^{(i)}, \tilde{y}^{(i)}) \mid \tilde{y}_c^{(i)} = \mathbb{1}[\Phi(\tilde{x}^{(i)})_c > \tau]\}_{i=1}^{\tilde{N}}\end{aligned} \tag{1}$$

After the pseudo-labels generation, the new dataset $\mathcal{D}$ combined with the additional pseudo-labeled dataset $\tilde{\mathcal{D}}$ is leveraged to train a model Φ in a supervised manner.

3.2 LINK PREDICTION WITH GRAPH NEURAL NETWORKS

Let us define a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} = \{v_1, ..., v_N\}$ and a set of observed edges $\mathcal{E} = \{e_{ij} = (v_i, v_j) | v_i, v_j \in \mathcal{V}\}$, where each node v in $\mathcal{G}$ has its own feature vector $\mathbf{x}_v \in \mathbb{R}^F$. Given a graph $\mathcal{G}$ and features vectors $\{\mathbf{x}_i\}_{i=1}^{N}$, Graph Neural Networks(GNNs) learn representations of each node v via an iterative aggregation of hidden representations of its neighboring nodes in the previous layer. The hidden representation of the node v in the l-th GNN layer is formulated as below:

$$h_v^{(l+1)} = \phi\big(h_v^{(l)},\, \psi(\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\})\big), \tag{2}$$

where ϕ combines the previous representation of v and its neighborhoods, ψ denotes an aggregation function, and $h_v^0 = \mathbf{x}_v$. For link prediction, node representations $h^{(L)}$ after L GNN layers are used to predict the existence of each link (v_i, v_j) using a link predictor f_θ, e.g., MLP, as below:

$$P(A_{ij} = 1) = \sigma(f_\theta(h_i^{(L)}, h_j^{(L)})), \tag{3}$$

where σ is a non-linear activation function that maps the prediction score to the range of $[0, 1]$. For simplicity, we denote our model that consists of GNNs and f_θ as Φ and link prediction scores of a link (v_i, v_j) from Φ as $\Phi(v_i, v_j)$.

4 PSEUDO-EDGE: PSEUDO-LABELING FOR LINK PREDICTION

The goal of our method is to revisit the link prediction task with GNNs in a semi-supervised scenario and harness a large amount of unlabeled data for link prediction. To be specific, whereas existing methods for link prediction use only given edges as positively-labeled data, we regard unseen edges as unlabeled data and propose a novel GNN-based pseudo-labeling framework for link prediction, *Pseudo-Edge*, that generates qualified pseudo-labels of unseen edges considering graph-structures and harnesses the pseudo-labels for training GNNs for link prediction.

4.1 DISTANCE-BASED CANDIDATES CONSTRUCTION

In order to generate pseudo-labels, we need to compute prediction scores for whole potential unlabeled edges (i.e., all node pairs except given ground-truth edges). This requires substantial computational costs, making it infeasible to compute on large graphs. A naive approach is to construct a set of potential edge candidates $\mathcal{U}^{rand}$ by randomly selecting node pairs from unlabeled edges.

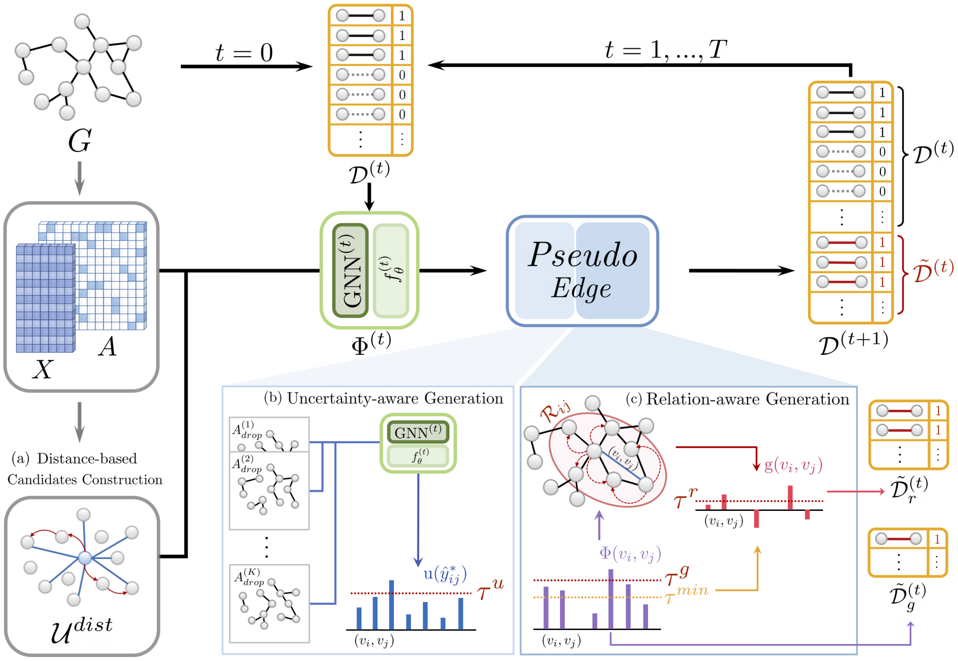


Figure 1: Overview of our *Pseduo-Edge* framework. *Pseudo-Edge* first (a) constructs a distance-based potential edge candidates $\mathcal{U}^{dist}$ for pseudo-labeling. For each iteration t, the GNN model $\Phi^{(t)}$ trained on the labeled dataset $\mathcal{D}^{(t)}$ predicts confidence scores of $\mathcal{U}^{dist}$, which are fed into the *Pseudo-Edge* framework. Then, in (b) uncertainty-aware pseudo-labels generation, $\mathcal{U}^{dist}$ is filtered to $\mathcal{U}^{u}$ whose uncertainty scores u$(\hat{y}^*)$, measured through K perturbed graphs, do not exceed τ^u. In (c) relation-aware pseudo-labels generation, among $\mathcal{U}^u$, *Pseudo-Edge* generates $\tilde{\mathcal{D}}_r^{(t)}$ whose relative differences g(v_i, v_j) are higher than τ^r and confidence scores $\Phi^{(t)}(v_i, v_j)$ are higher than τ^{min}. Also, *Pseudo-Edge* generates $\tilde{\mathcal{D}}_g^{(t)}$ whose $\Phi^{(t)}(v_i, v_j)$ are higher than τ^g. Finally, $\mathcal{D}^{(t+1)}$ is updated by adding pseudo-labels $\tilde{\mathcal{D}}^{(t)}$ composed of $\tilde{\mathcal{D}}_r^{(t)}$ and $\tilde{\mathcal{D}}_g^{(t)}$.

However, this can lead to the selection of node pairs that are far from each other, which constitutes candidates that are unlikely to be edges. Instead, we construct a set of candidates $\mathcal{U}^{dist}$ whose node pairs are apart from each other in the range of $(1, k)$ distance as follows:

$$\mathcal{U}^{dist} = \{(v_i, v_j) | 1 < d(v_i, v_j) < k, v_i, v_j \in \mathcal{V}\}, \tag{4}$$

where $d(v_i, v_j)$ is a length of the shortest path between v_i and v_j. For instance, if k equals 4, our proposed candidate set $\mathcal{U}^{dist}$ contains node pairs that are 2 and 3-hop distant from each other. This candidate formulation gives a chance to preserve more node pairs that are likely to be ground-truth edges and effectively reduce the search space for pseudo-labeling.

4.2 Relation-aware pseudo-labels Generation

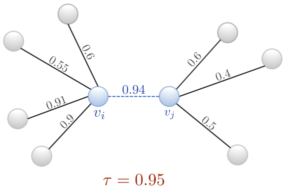


Figure 2: An example of a plausible candidate (v_i, v_j) where fixed threshold τ alone can not capture.

Recent works for pseudo-labeling (Zhang et al., 2021; Wang et al., 2022) have been proposed to generate high-quality pseudo-labels with advanced thresholding techniques according to various factors (e.g., classes, learning status) beyond a fixed threshold. However, there is no study on pseudo-labeling with advanced thresholds that considers the characteristic, connection between data samples, of the graph domain. Specifically, Fig.2 describes a circumstance that is elusive to generate high-quality pseudo-labels with only a fixed threshold. Let us denote the fixed threshold as τ. An edge candidate (v_i, v_j) in the figure obtains the confidence score $\Phi(v_i, v_j) = 0.94$, shown in blue. Although the score is slightly lower than $\tau = 0.95$, it shows the highest confidence among local edges. Nevertheless, τ alone is not capable of reflecting underlying connections between samples (edges) in the graph. Thus, relying exclusively on absolute threshold may fail to capture the relative confidence of artificial edges to corresponding nearby edges in local regions.

To address the limitation above, we propose a relation-aware threshold τ^r which reflects the comparative superiority of each unlabeled edge over its local region in graphs, which enables delicate generation of pseudo-labels along with fixed threshold. In order to compute the relative difference of each unlabeled edge from respective local region, we first define the local region $\mathcal{R}_{ij}$ of an unlabeled edge (v_i, v_j) based on a pair of two nodes, v_i, v_j, and their neighbors, $\mathcal{N}_{v_i}, \mathcal{N}_{v_j}$ as follows:

$$\mathcal{R}_{ij} = \mathcal{R}_{ij}^{(1)} \cup \mathcal{R}_{ij}^{(2)}, \tag{5}$$

$$\begin{aligned} \mathcal{R}_{ij}^{(1)} &= \{\hat{e} | \hat{e} \in \{v_i\} \times \mathcal{N}_{v_j} \cup \{v_j\} \times \mathcal{N}_{v_i}\}, \\ \mathcal{R}_{ij}^{(2)} &= \{\hat{e} | \hat{e} \in \mathcal{N}_{v_i} \times \mathcal{N}_{v_j}\}, \end{aligned} \tag{6}$$

where $\times$ denotes the Cartesian product of two sets. In other words, $R_{ij}^{(1)}$ involves pairs between each node in (v_i, v_j) and neighbors of the other node while $R_{ij}^{(2)}$ contains pairs between neighbors of each node. Then the local region score of (v_i, v_j), denoted as $\mathrm{s}(\mathcal{R}_{ij})$, is formulated as the linear combination between the average score of edges in $\mathcal{R}_{ij}^{(1)}$ and that in $\mathcal{R}_{ij}^{(2)}$:

$$\mathrm{s}(\mathcal{R}_{ij}) = \alpha \cdot \mathbb{E}_{\hat{e}_{mn} \in \mathcal{R}_{ij}^{(1)}}[\Phi(v_m, v_n)] + (1 - \alpha) \cdot \mathbb{E}_{\hat{e}_{mn} \in \mathcal{R}_{ij}^{(2)}}[\Phi(v_m, v_n)], \tag{7}$$

where α is a hyperparameter controlling the impact of 1-hop local region over 2-hop region. Then, we calculate the relative difference $\mathrm{g}(v_i, v_j)$ of prediction scores between (v_i, v_j) and $\mathcal{R}_{ij}$ as

$$\mathrm{g}(v_i, v_j) = \Phi(v_i, v_j) - \mathrm{s}(\mathcal{R}_{ij}). \tag{8}$$

Now, the relative threshold τ^r is defined by the top-N_r largest relative difference $\mathrm{g}(v_i, v_j)$ among the given unlabeled candidates $\mathcal{U}^{dist}$ as

$$\tau^r = \mathrm{Top}(\{\mathrm{g}(v_i, v_j) | (v_i, v_j) \in \mathcal{U}^{dist}\}, N_r), \tag{9}$$

where $\mathrm{Top}(\cdot, N_r)$ returns the N_r-th largest value from a given set of values. Further, since samples with large relative differences but low confidence scores rather give noise, we generate pseudo-labels $\tilde{D}_r$ for each sample $\tilde{x} = (v_i, v_j)$ whose confidence score is higher than the minimum fixed threshold τ^{min} and the relative difference is larger than τ^r as follows:

$$\tilde{\mathcal{D}}_r = \{(\tilde{x}^{(i)}, \tilde{y}^{(i)}) \mid \mathrm{g}(\tilde{x}^{(i)}) > \tau^r \wedge \Phi(\tilde{x}^{(i)}) > \tau^{min}\}_{i=1}^{N_r} \tag{10}$$

In addition, we generate pseudo-labels $\tilde{D}_g$ for samples whose confidence scores are higher than the N_g-th largest confidence scores as

$$\tilde{\mathcal{D}}_g = \{(\tilde{x}^{(i)}, \tilde{y}^{(i)}) \mid \Phi(\tilde{x}^{(i)}) > \tau^g\}_{i=1}^{N_g}, \tag{11}$$

where τ^g is fixed global threshold that is the N_g-th highest confidence score among scores of candidates. Finally, we combine the two pseudo-labels $\tilde{D}_r$ and $\tilde{D}_g$ to generate the final pseudo-labels $\tilde{D}$ by adjusting the number of $\tilde{D}_r$ and $\tilde{D}_g$ to be β% and (1-β%) of the total number as $\tilde{\mathcal{D}} = \tilde{\mathcal{D}}_r \cup \tilde{\mathcal{D}}_g$.

4.3 Uncertainty-Aware pseudo-labels Generation

Pseudo-labeling methods are mainly designed to select unlabeled samples with high confidence scores as pseudo-labels. Even though the selection based on high confidence scores can be effective for capturing informative samples, there exists another issue with the selection of *over-confident* samples. This also applies to our GNN-based pseudo-labeling, especially when the model overfits to specific graph structures and thereby assigns high prediction scores to only samples with the specific structures. Such ill-calibrated networks may yield noisy pseudo-labeled samples, which incurs poor generalization and unstable learning.

In order to alleviate this issue, we present uncertainty-aware pseudo-labels generation considering uncertainty in graph structures. Specifically, to measure each sample's uncertainty from the graph structure, we first randomly drop out edges in A for K iteration with a probability of p_u. That is, assuming $A'^{(k)}$ as an adjacency matrix with k-th randomly selected edges, the k-th perturbed adjacency matrix $A^{(k)}_{drop}$ is expressed as $A^{(k)}_{drop} = A - A'^{(k)}$. Then we derive the prediction outputs of a given sample $\hat{y}^*_{ij}$ with respect to $A^{(1)}_{drop}, ..., A^{(k)}_{drop}$ and an input feature matrix X:

$$\begin{gathered} \hat{y}^*_{ij} = \{f_\theta(z_i, z_j) | z_i, z_j \in Z^{(k)})\}^K_{k=1}, \\ Z^{(k)} = \text{GNN}(A^{(k)}_{drop}, X) \end{gathered} \tag{12}$$

Finally, the uncertainty of the sample $\text{u}(\hat{y}^*_{ij})$ is measured as the variance of prediction scores derived from K perturbed graph structures:

$$\text{u}(\hat{y}^*_{ij}) = \sum_{k=1}^{K} \frac{(\mathbb{E}(\hat{y}^*_{ij}) - \hat{y}^{*(k)}_{ij})^2}{K} \tag{13}$$

Here, we measure the uncertainty of every candidates and filter out links which possess higher uncertainty than a threshold τ^u at the beginning of pseudo-labeling procedure:

$$\mathcal{U}^u = \{(i, j) | \text{u}(\hat{y}^*_{ij}) < \tau^u\} \tag{14}$$

Throughout this process, the network is able to produce more reliable candidates which satisfy sufficient certainty as well as high confidence. Such pseudo-labeled edges offer appropriate guidance to the model, hence increase robustness to potential calibration error.

Algorithm 1 The Process of pseudo-label generation in *Pseudo-Edge*

```
  Input $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\mathcal{D}^{(0)} = \{(x^{(i)}, y^{(i)})\}^N_{i=1}$, $\Phi^{(0)}$
  Output $\mathcal{D}^{(T+1)}$, $\Phi^{(T+1)}$
Construct distance-based edge candidates $\mathcal{U}^{dist}$;
for $t = 0 \dots T$ do
    Train $\Phi^{(t)}$ on $\mathcal{D}^{(t)}$;
    Compute prediction scores of $\mathcal{U}^{dist}$ from the trained $\Phi^{(t)}$;
    $\mathcal{U}^u \leftarrow \{(v_i, v_j) | \text{u}(\hat{y}^*_{ij}) < \tau^u, (v_i, v_j) \in \mathcal{U}^{dist}\}$;
    $\tilde{\mathcal{D}}^{(t)}_g = \{(\tilde{x}^{(i)}, \tilde{y}^{(i)}) \mid \Phi(\tilde{x}^{(i)}) > \tau^g\}^{N_g}_{i=1}$;
    Construct $\mathcal{R}_{ij}$ for $\hat{e}_{ij} \in \{(v_i, v_j) \mid (v_i, v_j) \in \mathcal{U}^u \wedge \Phi^{(t)}(v_i, v_j) < \tau^g\}$
    $\tilde{\mathcal{D}}^{(t)}_r = \{(\tilde{x}^{(i)}, \tilde{y}^{(i)}) \mid \text{g}(\tilde{x}^{(i)}) > \tau^r \wedge \Phi^{(t)}(\tilde{x}^{(i)}) > \tau^{min}\}^{N_r}_{i=1}$;
    $\tilde{\mathcal{D}}^{(t)} \leftarrow \tilde{\mathcal{D}}^{(t)}_g \cup \tilde{\mathcal{D}}^{(t)}_r$;
    $\mathcal{D}^{(t+1)} \leftarrow \mathcal{D}^{(t)} \cup \tilde{\mathcal{D}}^{(t)}$;
    Initialize a new model $\Phi^{(t+1)}$;
end for
Train $\Phi^{(T+1)}$ on $\mathcal{D}^{(T+1)}$;
```

Overall Framework Our proposed pseudo-labeling framework, *Pseudo-Edge*, is organized by three stages: 1) Candidates construction for pseudo-labeling, 2) Relation-aware pseudo-labels generation, 3) Uncertainty-aware pseudo-labels generation. Fig.1 illustrates the proposed framework. Unlabeled candidates $\mathcal{U}^{dist}$ are first constructed in Eq. (4). For each iteration t, we train the model $\Phi^{(t)}$ on the labeled dataset $\mathcal{D}^{(t)}$ and candidates $\mathcal{U}^{dist}$ are fed into the trained model $\Phi^{(t)}$ to generate prediction scores for each candidate. Then we measure uncertainties of candidates in Eq. (13) and filter out samples whose the uncertainty $\text{u}(\hat{y}^*_{ij})$ is higher than τ^u. Based on these filtered candidates, our *Pseudo-Edge* generates two pseudo-labels, $\tilde{\mathcal{D}}^{(t)}_r$ and $\tilde{\mathcal{D}}^{(t)}_g$, and generate a pseudo-label dataset $\tilde{\mathcal{D}}^{(t)}$ by combining them. After the t-th generation process, we train a new model, $\Phi^{(t+1)}$ on a new labeled dataset $\mathcal{D}^{(t+1)}$ composed of $\tilde{\mathcal{D}}^{(t)}$ and $\mathcal{D}^{(t)}$. Finally, after total generation processes, a final model $\Phi^{(T+1)}$ is trained on $\mathcal{D}^{(T+1)}$ and used for evaluation. The complete training algorithm is described in Alg.1.

5 Experiments

In this section, we evaluate the effectiveness of our proposed *Pseudo-Edge* against state-of-the-art GNN-based semi/self-supervised models on link prediction benchmark datasets.

5.1 Experimental Setup

Datasets. We evaluate our proposed method on five benchmark datasets: two Open Graph Benchmark(OGB)(Hu et al., 2020) datasets (OGBL-COLLAB, OGBL-DDI) for link prediction and three citation networks (Bojchevski & Günnemann, 2018) (Cora, Citeseer, Pubmed). For the three citation networks, we randomly select 5% and 10% of the given edges and the same numbers of disconnected node pairs as validation and test samples. For OGB datasets, we follow the official train/validation/test dataset splits (Hu et al., 2020). Detailed data statistics are provided in Tab.1.

Dataset	#Nodes	#Edges	Avg. degree	Split ratio
OGBL-COLLAB	235,868	1,285,465	8.2	92/4/4
OGBL-DDI	4,267	1,334,889	500.5	80/10/10
CORA	19,793	107,816	4.49	85/5/10
CITESEER	4,230	9,074	1.77	85/5/10
PUBMED	19,717	75,352	3.15	85/5/10

Table 1: Statistics and evaluation metrics of link prediction datasets.

Baselines. We evaluate *Pseudo-Edge* on three standard GNNs: GCN(Kipf & Welling, 2016), GraphSAGE(i.e. SAGE)(Hamilton et al., 2017), and JKNet(Xu et al., 2018). Since GNN-based semi/self-supervised learning models for link prediction have not been studied, we compare the performance of our method against state-of-the-arts GNN-based semi/self-supervised learning baselines for node classification: MVGRL(Hassani & Khasahmadi, 2020), GRAND(Feng et al., 2020), and LAGNN(Liu et al., 2022). MVGRL trains GNNs in a self-supervised fashion by maximizing mutual information between representations of the same node, each derived from different structural views. GRAND trains GNNs in a semi-supervsied manner by leveraging consistency regularization to optimize the prediction consistency of unlabeled nodes across different data augmentations. Lastly, LAGNN improves GNN representations within self-supervised framework by augmenting node features via generative model pretrained of neighborhood features distribution.

Evaluation. For all experiments, we evaluate the link prediction performance of models based on two ranking evaluation metrics, Hits@20 and Hits@50, in (Hu et al., 2020). Specifically, each model ranks positive test edges against negative test edges and computes the ratio of positive test edges ranked at K-th place or above (Hits@K). We report averaged test performance with their standard deviation over 10 runs with different seeds (from 0 to 9).

Implementation details. We implemented GNNs and our method using PyTorch (Paszke et al., 2019), PyTorch Geometric (Fey & Lenssen, 2019), and OGB (Hu et al., 2020). For each baseline, we used the implementation in the official github repository for each method. We set the number of GNN layers to 3 and the hidden dimenstion to 256 for all GNNs. In our method, pseudo-labels are selected from a set of potential edge candidates, $\mathcal{U}^{dist}$, up to twice as many as the original training data through up to five iterations. Note that $\mathcal{U}^{dist}$ does not contain positive and negative edges in the validation/test samples. We choose hyperparameters based on the best validation performance. The experiments are conducted on a RTX 3090 (24GB).

5.2 Results on Link Prediction

Table2 shows the link prediction performance of our method and baselines on three standard GNNs and five benchmark datasets. Since LAGNN generates node features conditioned on original input features of center nodes, LAGNN can not be evaluated on OGBL-DDI owing to the absence of input features in OGBL-DDI. As shown in Table 2, our *Pseudo-Edge* consistently achieved state-of-the-arts performance across four datasets.

Compared to vanilla GNNs, *Pseudo-Edge* shows significant improvements over all GNNs on OGBL-COLLAB, OGBL-DDI, Pubmed, and Cora. Especially, *Pseudo-Edge* remarkably improved

Dataset		OGBL-COLLAB		OGBL-DDI		PUBMED		CORA		CITESEER	
GNNs	**Methods**	Hits@20	Hits@50	Hits@20	Hits@50	Hits@20	Hits@50	Hits@20	Hits@50	Hits@20	Hits@50
GCN	Vanilla	41.76 ± 1.10	48.56 ± 0.72	56.71 ± 7.80	74.99 ± 3.13	31.18 ± 2.34	48.81 ± 1.30	23.06 ± 1.20	33.70 ± 1.21	59.44 ± 2.59	73.31 ± 1.09
	MVGRL	27.14 ± 2.36	36.63 ± 1.77	19.78 ± 2.89	32.20 ± 4.18	28.84 ± 1.41	41.50 ± 1.98	21.23 ± 1.12	30.42 ± 1.24	52.62 ± 2.31	67.27 ± 1.97
	GRAND	40.09 ± 1.63	47.95 ± 0.93	59.38 ± 4.83	73.12 ± 4.38	35.28 ± 1.70	49.29 ± 1.45	23.49 ± 1.11	33.13 ± 1.48	61.60 ± 2.02	**74.47 ± 0.59**
	LAGNN	44.42 ± 0.68	51.08 ± 0.42	N/A	N/A	34.21 ± 1.53	52.55 ± 1.87	23.58 ± 1.76	36.02 ± 1.54	**63.01 ± 2.11**	74.20 ± 1.78
	Ours	**46.56 ± 0.96**	**53.06 ± 0.42**	**69.27 ± 7.78**	**82.79 ± 3.10**	**37.67 ± 1.06**	**55.78 ± 0.95**	**25.57 ± 0.78**	**38.54 ± 0.90**	61.23 ± 2.91	73.66 ± 1.42
SAGE	Vanilla	42.22 ± 1.02	50.33 ± 0.58	30.07 ± 9.78	65.38 ± 12.50	23.78 ± 1.74	36.22 ± 1.43	25.36 ± 1.18	38.36 ± 2.51	43.52 ± 1.89	49.86 ± 0.37
	MVGRL	25.23 ± 1.91	35.30 ± 1.29	21.62 ± 3.96	34.91 ± 3.56	23.81 ± 1.92	36.01 ± 1.45	23.08 ± 2.40	33.07 ± 2.33	28.12 ± 3.29	43.35 ± 3.22
	GRAND	42.05 ± 1.51	50.20 ± 0.79	32.52 ± 10.60	73.00 ± 9.36	22.88 ± 1.40	37.67 ± 1.31	26.25 ± 2.39	39.79 ± 2.39	43.78 ± 0.94	49.64 ± 0.53
	LAGNN	44.54 ± 0.63	51.52 ± 0.49	N/A	N/A	25.02 ± 1.81	41.44 ± 1.57	25.23 ± 2.17	39.38 ± 1.14	45.40 ± 0.85	49.99 ± 0.48
	Ours	**46.41 ± 0.90**	**53.40 ± 0.39**	**41.93 ± 14.04**	**87.16 ± 1.11**	**27.39 ± 2.48**	**44.22 ± 1.27**	**26.55 ± 1.99**	**39.94 ± 1.73**	**45.56 ± 1.01**	**50.82 ± 0.25**
JKNet	Vanilla	43.75 ± 0.87	51.08 ± 0.86	39.88 ± 11.22	72.53 ± 4.08	29.12 ± 1.91	44.79 ± 1.12	20.73 ± 1.17	30.18 ± 0.83	52.51 ± 3.77	69.33 ± 1.03
	MVGRL	30.32 ± 1.22	38.98 ± 0.95	12.67 ± 6.30	21.81 ± 7.08	28.54 ± 1.82	42.86 ± 2.37	20.74 ± 1.15	30.73 ± 1.50	52.71 ± 1.96	68.43 ± 1.09
	GRAND	44.42 ± 1.09	51.58 ± 0.98	46.07 ± 9.39	73.94 ± 4.96	29.73 ± 1.57	43.59 ± 1.74	20.08 ± 1.75	29.43 ± 1.38	52.68 ± 3.19	69.57 ± 1.03
	LAGNN	44.96 ± 1.08	51.50 ± 0.60	N/A	N/A	30.33 ± 1.95	46.30 ± 1.04	20.28 ± 1.32	30.30 ± 0.67	**54.09 ± 2.08**	**70.74 ± 1.4**
	Ours	**47.40 ± 1.05**	**54.26 ± 1.05**	**69.33 ± 10.65**	**88.05 ± 1.08**	**33.78 ± 1.39**	**47.54 ± 0.86**	**23.21 ± 1.44**	**35.11 ± 1.31**	52.40 ± 2.65	67.79 ± 0.69

Table 2: Link prediction performances (Hits@20 and Hits@50) of our *Pseudo-Edge* and baselines across three GNN architectures and 5 benchmark datasets. Each number is the average performance for 10 random initialization of the experiments.

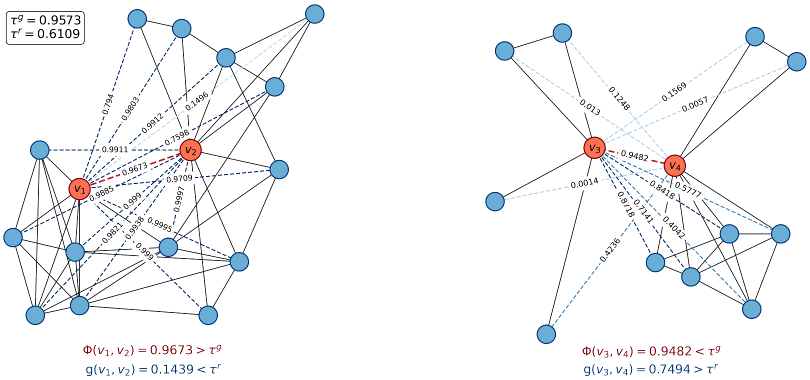
Module			OGBL-COLLAB		OGBL-DDI		PUBMED		CORA		CITESEER	
$\mathcal{U}^{dist}$	τ^r	τ^u	Hits@20	**Hits@50**	**Hits@20**	Hits@50	Hits@20	**Hits@50**	Hits@20	**Hits@50**	Hits@20	**Hits@50**
			41.00 ± 1.00	49.97 ± 0.49	29.24 ± 10.66	85.77 ± 1.52	22.03 ± 1.49	34.59 ± 1.33	24.04 ± 1.89	35.26 ± 1.65	40.30 ± 1.79	49.54 ± 0.47
✓			41.40 ± 1.04	50.40 ± 0.85	34.68 ± 8.12	84.69 ± 3.85	26.48 ± 1.46	41.75 ± 1.55	24.81 ± 1.46	38.95 ± 1.19	44.11 ± 1.01	50.23 ± 0.55
✓	✓		42.17 ± 1.15	51.52 ± 0.86	39.50 ± 13.26	87.15 ± 1.51	26.48 ± 2.61	**45.17 ± 1.25**	26.03 ± 2.06	**40.32 ± 1.67**	45.44 ± 0.94	50.43 ± 1.09
✓	✓	✓	**46.41 ± 0.90**	**53.40 ± 0.39**	**41.93 ± 14.04**	**87.16 ± 1.11**	**27.39 ± 2.48**	44.22 ± 1.27	**26.55 ± 1.99**	39.94 ± 1.73	**45.56 ± 1.01**	**50.82 ± 0.25**

Table 3: Ablation study of each module in *Pseudo-Edge* with SAGE on four datasets.

the link prediction performance (Hits@20) of GNNs on OGBL-DDI by 12.56%, 11.86%, 29.45% for GCN, SAGE, and JKNet, respectively. Furthermore, *Pseudo-Edge* consistently outperforms naive GNNs on other datasets by $3.07 \sim 4.5\%$, $2.75 \sim 8\%$, $1.58 \sim 4.93\%$ on OGBL-COLLAB, Pubmed, and Cora. This shows the effectiveness of leveraging unlabeled edges for pseudo-labels for GNN-based link prediction. Compared to GNN-based semi/self-supervised learning models, our *Pseudo-Edge* consistently outperforms all baselines on 4 out of 5 datasets, where the improvements over the best baseline are $1.88 \sim 2.68\%$, $9.41 \sim 23.26\%$, $1.24 \sim 3.23\%$, and $0.15 \sim 4.38\%$ on OGBL-COLLAB, OGBL-DDI, Pubmed, and Cora. Interestingly, MVGRL and GRAND shows poor or marginal performances compared to vanila GNNs. This implies that existing *node-level* methods for semi/self supervised learning have a difficulty in directly benefiting for link prediction and there is a need to design semi-supervised learning for link prediction.

5.3 QUANTITATIVE ANALYSIS

Here, we conduct an ablation study of our *Pseudo-Edge* to demonstrate the effect of each component in our method: **(1)** Construction of distance-based edge candidates ($\mathcal{U}^{dist}$) **(2)** Relation-aware pseudo-labels generation (τ^r) **(3)** Uncertainty-aware pseudo-labels generation (τ^u). We employ SAGE as our backbone GNN and conduct 10 runs across all datasets. The default setting (the first row in Tab.3) is pseudo-labeling with randomly sampled edge candidates $\mathcal{U}^{rand}$ and a fixed threshold τ^g. First, as shown in Table 3, *Pseudo-Edge* with distance-based candidates $\mathcal{U}^{dist}$ consistently improves the link prediction performance $0.40 \sim 7.16\%$, where largely improves the performance by 7.16% against the randomized strategy in Hits@50 on Pubmed. Second, our relation-aware pseudo-labels generation (τ^r) clearly boosts up the performance compared to the pseudo labeling with the only fixed threshold. In particular, adopting τ^r greatly increases the evaluation result by 4.82% on OGBL-DDI and 3.42% on Pubmed, all on the basis of each standard metrics. Lastly, uncertainty-aware pseudo-labels generation further improves the performance of *Pseudo-*



(a) A local region $\mathcal{R}_{1,2}$ of a ground-truth edge (v_1, v_2) that belongs to category **[1]**

(b) A local region $\mathcal{R}_{3,4}$ of a ground-truth edge (v_3, v_4) that belongs to category **[2]**

Figure 3: Visualization of local regions of test edges(red dashed lines) sampled from different strategies: (a) a test edge (v_1, v_2) selected as a pseudo-label based on a fixed global threshold $\tau^g = 0.9573$, (b) a test edge (v_3, v_4) selected as a pseudo-label based on our proposed relation-aware threshold $\tau^r = 0.6109$. Blue dashed lines denote node pairs $\hat{e}$ in local regions $\mathcal{R}_{ij}$ of each test edge and the intensity is proportional to the prediction score of $\Phi(\hat{e})$. The test edge (v_3, v_4) can not be selected based on τ^g since its prediction score $\Phi(v_3, v_4) = 0.9482$ is slightly below $\tau^g = 0.9573$ whereas our relation-aware pseudo-labeling can capture as the prediction score is relatively higher than node pairs in the local region $\mathcal{R}_{3,4}$, i.e. a large gap $\mathrm{g}(v_3, v_4) = 0.7494$.

Edge, which significantly enhances the performance on OGBL-COLLAB by 4.24% and 1.88% for Hits@20 and Hits@50, respectively.

5.4 Qualitative Analysis

We provide qualitative analysis qualitative to understand why our *relation*-aware pseudo-labels generation is effective. We visualize local regions of test edges, not included in the original graph, selected from different strategies. Specifically, a test edge (v_1, v_2) in Fig.3a is selected as a pseudo-label based on a fixed global threshold $\tau^g = 0.9573$ and a test edge (v_3, v_4) in Fig.3b is selected as a pseudo-label based on our proposed relation-aware threshold $\tau^r = 0.6109$. The visualization results clearly demonstrates our argument in Fig.2. In case of an edge (v_3, v_4), it overlooked during the generation based on τ^g since its prediction score $\Phi(v_3, v_4) = 0.9482$ is slightly below $\tau^g = 0.9573$. However, (v_3, v_4) is selected as a pseudo-label when performing relation-aware pseudo-labeling, as the prediction score is relatively higher than node pairs in the local region $\mathcal{R}_{3,4}$, i.e. a large gap $\mathrm{g}(v_3, v_4) = 0.7494$ that is highly above τ^r, even (v_1, v_2) whose score is not noticeable among $\mathcal{R}_{1,2}$, $\mathrm{g}(v_1, v_2) = 0.1439$, as illustrated in Fig.3a. Thus, our relation-aware pseudo-lables generation enables the overall pseudo-labeling process to consider both absolute high confidence scores and comparative superiority of each sample's confidence scores over its local region.

6 Conclusion

We propose a novel pseudo-labeling framework for link prediction, *Pseudo-Edge*, that hat generates qualified pseudo-labels in consideration of graph structures and harnesses them for the link prediction. *Pseudo-Edge* carefully generates pseudo-labels through our proposed relation-aware pseudo-labels generation reflecting the comparative superiority of each sample over its local neighborhoods in graphs. Also, we propose uncertainty-aware pseudo-labels generation that can effectively filter out over-confident samples when the model overfits to specific graph structures. Our comprehensive experiments show that *Pseudo-Edge* achieves strong performance on three standard GNNs over six benchmark datasets for link prediction compared to state-of-the-art GNN-based semi/self-supervised learning models. One limitation of our method is that we do not directly utilize pseudo-lables by adding them as real edges in the graph for GNNs. We believe future works investigate the effectiveness of utilizing pseudo-lables as ground-truth edges for GNNs.

REFERENCES

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HklkeR4KPB`.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1ZdKJ-0W`.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6912–6920, 2021.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. On the equivalence of decoupled graph convolution network and label propagation. In *Proceedings of the Web Conference 2021*, pp. 3651–3662, 2021.

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126. PMLR, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

Songtao Liu, Rex Ying, Hanze Dong, Lanqing Li, Tingyang Xu, Yu Rong, Peilin Zhao, Junzhou Huang, and Dinghao Wu. Local augmentation for graph neural networks. In *International Conference on Machine Learning*, pp. 14054–14072. PMLR, 2022.

Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11557–11568, 2021.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, volume 1, pp. 29–36. IEEE, 2005.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5892–5899, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34:23768–23779, 2021.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020a.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020b.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.