Quantifying the Knowledge in GNNs for Reliable Distillation into MLPs

Lirong Wu [1] **Haitao Lin** [1] **Yufei Huang** [1] **Stan Z. Li** [1]

Abstract

To bridge the gaps between topology-aware Graph Neural Networks (GNNs) and inference-efficient Multi-Layer Perceptron (MLPs), GLNN (Zhang et al., 2021) proposes to distill knowledge from a well-trained teacher GNN into a student MLP. Despite their great progress, comparatively little work has been done to explore *the reliability of different knowledge points (nodes) in GNNs, especially their roles played during distillation.* In this paper, we first quantify the knowledge reliability in GNN by measuring the invariance of their information entropy to noise perturbations, from which we observe that different knowledge points *(1)* show different distillation speeds (*temporally*); *(2)* are differentially distributed in the graph (*spatially*). To achieve reliable distillation, we propose an effective approach, namely *Knowledge-inspired Reliable Distillation* (KRD), that models the probability of each node being an informative and reliable knowledge point, based on which we sample a set of additional reliable knowledge points as supervision for training student MLPs. Extensive experiments show that KRD improves over the vanilla MLPs by 12.62% and outperforms its corresponding teacher GNNs by 2.16% averaged over 7 datasets and 3 GNN architectures. Codes are publicly available at: https://github.com/LirongWu/RKD.

1. Introduction

Recent years have witnessed the great success of Graph Neural Networks (GNNs) (Hamilton et al., 2017; Kipf & Welling, 2016; Veličković et al., 2017; Liu et al., 2020; Wu et al., 2020; Zhou et al., 2020; Wu et al., 2021b;a) in handling graph-related tasks. Despite their great *academic success*, Multi-Layer Perceptrons (MLPs) remain the primary workhorse for practical *industrial applications*. One reason for such academic-industrial gap is the neighborhood-fetching latency incurred by data dependency in GNNs (Jia et al., 2020; Zhang et al., 2021), which makes it hard to deploy for latency-sensitive applications. Conversely, Multi-Layer Perceptrons (MLPs) involve no data dependence between data pairs and infer much faster than GNNs, but their performance is less competitive. Motivated by these complementary strengths and weaknesses, one solution to reduce their gaps is to perform GNN-to-MLP knowledge distillation (Yang et al., 2021; Zhang et al., 2021; Gou et al., 2021), which extracts the knowledge from a well-trained teacher GNN and then distills the knowledge into a student MLP.

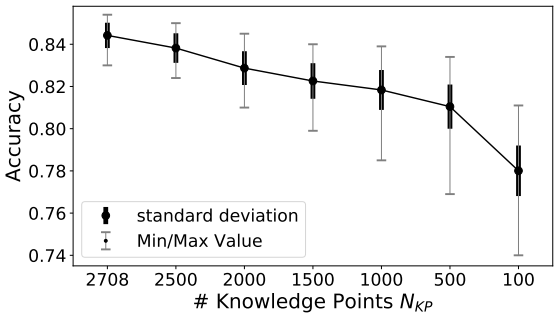


Figure 1. Mean, standard deviation, and minimum/maximum classification accuracy of student MLPs trained with different combinations of (randomly sampled) GNN knowledge points on Cora.

Despite the great progress, most previous works have simply treated all knowledge points (nodes) in GNNs as equally important, and few efforts are made to explore *the reliability of different knowledge points in GNNs and the diversity of the roles they play in the distillation process.* From the motivational experiment in Fig. 1, we can make two important observations about knowledge points: ***(1) More is better:*** the performance of distilled MLPs can be improved as the number of knowledge points N_{KP} increases; and ***(2) Reliable is better:*** the performance variances (e.g., standard deviation and best/worst performance gap) of different knowledge combinations are enlarged as N_{KP} decreases. The above two observations suggest that different knowledge points may play different roles in the distillation process and that distilled MLPs can consistently benefit from *more reliable* knowledge points, while those uninformative and unreliable knowledge points may contribute little to the distillation.

[1]AI Lab, Research Center for Industries of the Future, Westlake University, Hangzhou, China. Correspondence to: Stan Z. Li <stan.zq.li@westlake.edu.cn>, Lirong Wu <wulirong@westlake.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Present Work. In this paper, we identify a potential *under-confidence* problem for GNN-to-MLP distillation, i.e., the distilled MLPs may not be able to make predictions as confidently as teacher GNNs. Furthermore, we conduct extensive theoretical and experimental analysis on this problem and find that it is mainly caused by the lack of reliable supervision from teacher GNNs. To provide more supervision for reliable distillation into student MLPs, we propose to quantify the knowledge in GNNs by measuring the invariance of their information entropy to noise perturbations, from which we find that different knowledge points *(1)* show different distillation speeds (*temporally*); *(2)* are differentially distributed in the graph (*spatially*). Finally, we propose an effective approach, namely *Knowledge-inspired Reliable Distillation* (KRD), for filtering out unreliable knowledge points and making full use of those with informative knowledge. The proposed KRD framework models the probability of each node being an information-reliable knowledge point, based on which we sample a set of additional reliable knowledge points as supervision for training student MLPs.

Our main contributions can be summarized as follows:

- We are the first to identify a potential *under-confidence* problem for GNN-to-MLP distillation, and more importantly, we described in detail what it represents, how it arises, what impact it has, and how to deal with it.
- We propose a perturbation invariance-based metric to quantify the reliability of knowledge in GNNs and analyze the roles played by different knowledge nodes *temporally* and *spatially* in the distillation process.
- We propose a *Knowledge-inspired Reliable Distillation* (KRD) framework based on the quantified GNN knowledge to make full use of those reliable knowledge points as additional supervision for training MLPs.

2. Related Work

GNN-to-GNN Knowledge Distillation. Despite the great progress, most existing GNNs share the de facto design that relies on message passing to aggregate features from neighborhoods, which may be one major source of latency in GNN inference. To address this problem, there are previous works that attempt to distill knowledge from large teacher GNNs to smaller student GNNs, termed as GNN-to-GNN distillation (Lassance et al., 2020; Zhang et al., 2020a; Ren et al., 2021; Joshi et al., 2021; Wu et al., 2022a;b). For example, the student model in RDD (Zhang et al., 2020b) and TinyGNN (Yan et al., 2020) is a GNN with fewer parameters but not necessarily fewer layers than the teacher GNN. Besides, LSP (Yang et al., 2020b) transfers the topological structure (rather than feature) knowledge from a pre-trained teacher GNN to a shallower student GNN. In addition, GNN-SD (Chen et al., 2020) directly distills knowledge across different GNN layers, mainly aiming to solve the over-smoothing problem but with unobvious performance improvement at shallow layers. Moreover, FreeKD (Feng et al., 2022) studies a free-direction knowledge distillation architecture, with the purpose of dynamically exchanging knowledge between two shallower GNNs. Note that both teacher and student models in the above works are GNNs, making it still suffer from neighborhood-fetching latency.

GNN-to-MLP Knowledge Distillation. To enjoy the topology awareness of GNNs and inference-efficient of MLPs, the other branch of graph knowledge distillation is to directly distill from teacher GNNs to lightweight student MLPs, termed as GNN-to-MLP distillation. For example, CPF (Yang et al., 2021) *directly* improves the performance of student MLPs by adopting deeper/wider network architectures and incorporating label propagation in MLPs, both of which burden the inference latency. Instead, GLNN (Zhang et al., 2021) distills knowledge from teacher GNNs to vanilla MLPs without other computing-consuming operations; while the performance of their distilled MLPs can be *indirectly* improved by employing more powerful GNNs, they still cannot match GNN-to-GNN distillation in terms of classification performance. To further improve GLNN, RKD-MLP (Anonymous, 2023) adopts a meta-policy to filter out unreliable soft labels, but this is essentially a down-sampling-style strategy that will further reduce the already limited supervision. In contrast, this paper aims to provide more reliable supervision for training student MLPs, which can be considered as an up-sampling-style strategy.

3. Preliminaries

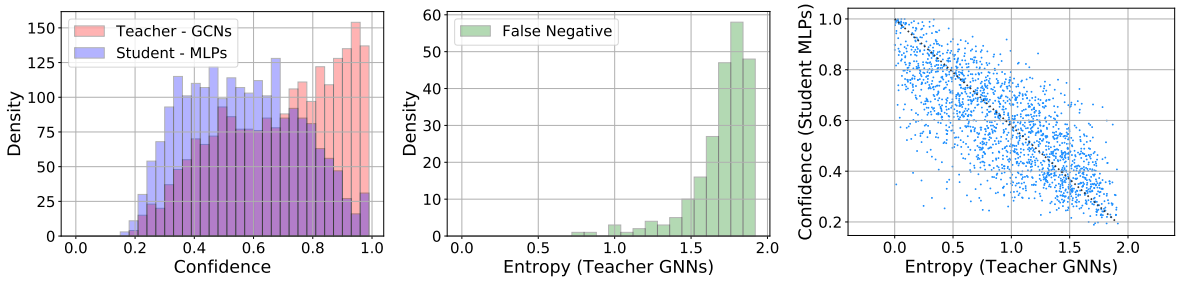
Notions and Problem Statement. Let $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ be a graph with the node set $\mathcal{V}$ and edge set $\mathcal{E}$, where $\mathcal{V}$ is the set of N nodes with features $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{N\times d}$. The graph structure is denoted by an adjacency matrix $\mathbf{A} \in [0,1]^{N\times N}$ with $\mathbf{A}_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ and $\mathbf{A}_{i,j} = 0$ if $e_{i,j} \notin \mathcal{E}$. Considering a semi-supervised node classification task where only a subset of node $\mathcal{V}_L$ with labels $\mathcal{Y}_L$ are known, we denote the labeled set as $\mathcal{D}_L = (\mathcal{V}_L, \mathcal{Y}_L)$ and unlabeled set as $\mathcal{D}_U = (\mathcal{V}_U, \mathcal{Y}_U)$, where $\mathcal{V}_U = \mathcal{V} \backslash \mathcal{V}_L$. The node classification aims to learn a mapping $\Phi : \mathcal{V} \rightarrow \mathcal{Y}$ so that it can be used to infer the ground-truth label $y_i \in \mathcal{Y}_U$.

Graph Neural Networks (GNNs). A general GNN framework consists of two key operations for each node v_i: (1) AGGREGATE: aggregating messages from neighborhood $\mathcal{N}_i$; (2) UPDATE: updating node representations. For an L-layer GNN, the formulation of the l-th layer is as

$$\begin{aligned} \mathbf{m}_i^{(l)} &= \mathrm{AGGREGATE}^{(l)}\left(\left\{\mathbf{h}_j^{(l-1)} : v_j \in \mathcal{N}_i\right\}\right) \\ \mathbf{h}_i^{(l)} &= \mathrm{UPDATE}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \mathbf{m}_i^{(l)}\right) \end{aligned} \tag{1}$$

where $1 \leq l \leq L$, $\mathbf{h}_i^{(0)} = \mathbf{x}_i$ is the input node feature, and $\mathbf{h}_i^{(l)}$ is the node representation of node v_i in the l-th layer.

Multi-Layer Perceptrons (MLPs). To achieve efficient



(a) Confidence Distribution (b) Histogram of *False Negative* Samples (c) GNN Entropy *vs.* MLP Confidence

Figure 2. ***(a)*** Histograms of the confidence distributions of teacher GCNs and students MLP for those correct predictions on the `Cora` dataset. ***(b)*** Distribution of *"False Negative"* samples w.r.t the information entropy of teacher's predictions on the `Cora` dataset. ***(c)*** Scatter curve of confidence (student MLP) and information entropy (teacher GCN) for those *"True Positive"* samples on the `Cora` dataset.

inference, the vanilla MLPs are used as the student model by default in this paper. For a L-layer MLP, the l-th layer is composed of a linear transformation, an activation function $\mathrm{ReLu}(\cdot)$, and a dropout function $\mathrm{Dropout}(\cdot)$, as follows

$$\mathbf{z}_i^{(l)} = \mathrm{Dropout}\left(\mathrm{ReLu}\left(\mathbf{z}_i^{(l-1)}\mathbf{W}^{(l-1)}\right)\right) \tag{2}$$

where $\mathbf{z}_i^{(0)} = \mathbf{x}_i$ is the input feature, and $\{\mathbf{W}^{(l)}\}_{l=0}^{L-1}$ are weight matrices with the hidden dimension F. In this paper, the network architecture of MLPs, such as the layer number L and layer size F, is set the same as that of teacher GNNs.

GNN-to-MLP Knowledge Distillation. The knowledge distillation is first introduced in (Hinton et al., 2015) to mainly handle image data. However, recent works on GNN-to-MLP distillation (Yang et al., 2021; Zhang et al., 2021) extend it to the graph domain by imposing KL-divergence constraint $\mathcal{D}_{KL}(\cdot,\cdot)$ between the label distributions generated by teacher GNNs and student MLPs, as follows

$$\mathcal{L}_{\mathrm{KD}} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathcal{D}_{KL}\left(\sigma\left(\mathbf{z}_i^{(L)}\right), \sigma\left(\mathbf{h}_i^{(L)}\right)\right) \tag{3}$$

where $\sigma(\cdot) = \mathrm{softmax}(\cdot)$, and all nodes (knowledge points) in the set $\mathcal{V}$ are indiscriminately used as supervisions.

4. Methodology

4.1. What Gets in the Way of Better Distillation?

Potential Under-confident Problem. The GNN-to-MLP distillation can be achieved by directly optimizing the objective function $\mathcal{L}_{\mathrm{KD}}$ defined in Eq. (3). However, such a straightforward distillation completely ignores the differences between knowledge points in GNNs and may suffer from a potential under-confident problem, i.e., the distilled MLP may fail to make predictions as confidently as teacher GNNs. To illustrate this problem, we report in Fig. 2(a) the confidences of teacher GCNs and student MLPs for those correct predictions by the UMAP (McInnes et al., 2018) algorithm on the Cora dataset. It can be seen that there exists a significant distribution shift between the confidence distribution of teacher GCNs and student MLPs, which confirms the existence of the under-confident problem. The direct hazard of such an under-confident problem is that it may push those samples located near the class boundaries into incorrect predictions, as shown in Fig. 3(a) and Fig. 3(c), which hinders the performance of student MLPs.

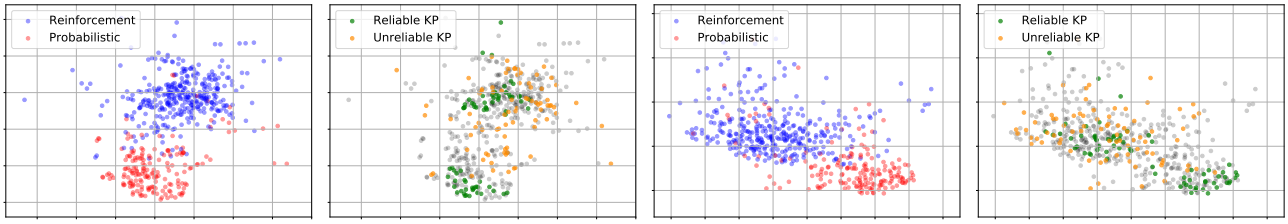
To go deeper into the under-confident problem and explore what exactly stands in the way of better GNN-to-MLP distillation, we conducted extensive theoretical and experimental analysis and found that one of the main causes could be due to the lack of reliable supervision from teacher GNNs.

Theoretical Analysis. The main strength of teacher GNNs over student MLPs is their excellent topology-awareness capability, which is mainly enabled by message passing. There have been a number of works exploring the roles of message passing in GNNs. For example, (Yang et al., 2020a) have proved that message passing (architecture design) in GNNs is equivalent to performing Laplacian smoothing (supervision design) on node embeddings in MLPs. In essence, message-passing-based GNNs implicitly take the objective of Dirichlet energy minimization (Belkin & Niyogi, 2001) as graph-based regularization, which is defined as follows

$$\mathcal{L}_{reg} = \mathrm{Tr}\left(\mathbf{Y}^\top \Delta \mathbf{Y}\right) = \sum_i \sum_{j\in\mathcal{N}_i} \left\| \frac{\mathbf{Y}_i}{\sqrt{d_i}} - \frac{\mathbf{Y}_j}{\sqrt{d_j}} \right\|_2^2 \tag{4}$$

where $\Delta = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized Laplacian operator, $\mathbf{D}$ is the degree matrix with $\mathbf{D}_{i,i} = d_i = \sum_j \mathbf{A}_{i,j}$, and $\mathbf{Y} = \mathrm{softmax}\left(\mathbf{H}^{(L)}\right)$ is the label distribution matrix.

Apart from the supervision of cross-entropy on the labeled set, message passing in GNNs implicitly provides a special kind of ***self-supervision***, which imposes regularization constraints on the label distributions between neighboring nodes. We conjecture that it is exactly such additional self-supervision that enables GNNs to make highly confident predictions. In contrast, student MLPs are trained in a way that cannot capture the fine-grained dependencies between neighboring nodes; instead, they only learn the overall contextual information about their neighborhood from teacher GNNs, resulting in undesirable under-confident predictions.



(a) Visualizations in GNNs (b) Spatial Distribution in GNNs (c) Visualizations in MLPs (d) Spatial Distribution in MLPs

Figure 3. ***(a)(c)*** Visualizations of the embeddings of teacher GNNs and student MLPs for two classes on `Cora`. ***(b)(d)*** Spatial distribution of knowledge points with the reliability ranked in the top 20% and bottom 10%, which are marked in green and orange, respectively.

Experimental Analysis. To see why the (distilled) student MLPs tend to make low-confidence predictions, we conducted an in-depth statistical analysis on two types of special samples. *(1)* The distribution of *"False Negative"* samples (predicted correctly by GNNs but incorrectly by MLPs) w.r.t the information entropy of teacher's predictions is reported in Fig. 2(b), from which we observe that most of the *"False Negative"* samples are distributed in the region of higher entropy. *(2)* For those *"True Positive"* samples (predicted correctly by both GNNs and MLPs), the scatter of confidence and information entropy from student MLPs and teacher GNNs is plotted in Fig. 2(c), which shows that GNN knowledge with high uncertainty (low reliability) may undermine the capability of student MLPs to make sufficiently confident predictions. Based on these two observations, it is reasonable to hypothesize that one cause of the under-confident problem suffered by student MLPs is be the lack of sufficiently reliable supervision from teacher GNNs.

4.2. How to Quantify the Knowledge in GNNs?

Based on the above experimental and theoretical analysis, a key issue in GNN-to-MLP distillation may be to provide ***more and reliable*** supervision for training student MLPs. Next, we first describe how to quantify the reliability of knowledge in GNNs, and then propose how to sample more reliable supervision through a knowledge-inspired manner.

Knowledge Quantification. Given a graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and a pre-trained teacher GNN $f_\theta(\cdot,\cdot)$, we propose to quantify the reliability of a knowledge point (node) $v_i \in \mathcal{V}$ in GNNs by measuring the invariance of its information entropy to noise perturbations, which is defined as follows

$$\rho_i = \frac{1}{\delta^2} \mathop{\mathbb{E}}_{\mathbf{X}' \sim \mathcal{N}(\mathbf{X}, \mathbf{\Sigma}(\delta))} \left[\left\| \mathcal{H}(\mathbf{Y}'_i) - \mathcal{H}(\mathbf{Y}_i) \right\|^2 \right], \quad \text{where} \quad \mathbf{Y}' = f_\theta(\mathbf{A}, \mathbf{X}') \quad \text{and} \quad \mathbf{Y} = f_\theta(\mathbf{A}, \mathbf{X}) \tag{5}$$

where δ is the variance of Gaussian noise and $\mathcal{H}(\cdot)$ denote the information entropy. The smaller the metric ρ_i is, the higher the reliability of knowledge point v_i is. The quantification of GNN knowledge defined in Eq. (5) has the following three strengths: ***(1)*** It measures the robustness of knowledge in teacher GNNs to noise perturbations, and thus more truly reflects the reliability of different knowledge points, which is very important for reliable distillation. ***(2)*** The message passing is what makes GNNs special over MLPs, so the key to quantify GNN knowledge is to measure its topology-awareness capability. Compared with node-wise information entropy, Eq. (5) not only reflects the node uncertainty, but also takes into account the contextual information from the neighborhood. ***(3)*** As will be analyzed next, the knowledge quantified by Eq. (5) shows the roles played by different knowledge points *spatially* and *temporally*.

Spatial Distribution of Knowledge Points. To explore the spatial distribution of different knowledge points in the graph, we first visualize the embeddings of teacher GNNs and student MLPs in Fig. 3(a) and Fig. 3(c), and then we mark the knowledge points with the reliability ranked in the top 20% and bottom 10% as green and orange in Fig. 3(b) and Fig. 3(d). To make it clearer, we only report the results for two classes on the Cora dataset; more visualizations can be found in **Appendix C**. We find that different knowledge points are differentially distributed in the graph, where most reliable knowledge points are distributed around the class centers regardless of being in teacher GNNs or student MLPs, while those unreliable ones are distributed at the class boundaries. The spatial distribution of knowledge points explains well why most of the *False Negatice* samples are located in regions with high uncertainty in Fig. 2(c).

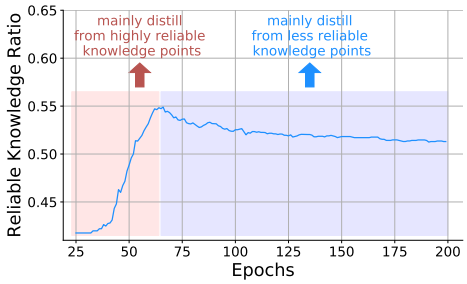


Figure 4. Percentage of highly reliable knowledge points on `Cora` to show the distillation speeds of different knowledge points.

Temporal Distribution of Knowledge Points. To see the distillation speed of different knowledge points, we explore which knowledge points the student MLPs will be fitted to first during the training process. We considered those knowledge points that are correctly predicted by student MLPs

and ranked in the top 50% of reliability, among which we calculate the percentage of points with the top 20% of reliability in Fig. 4. It can be seen that student MLPs will quickly fit to those highly reliable knowledge points first as the training proceeds, and then gradually learn from those relatively less reliable knowledge points. This indicates that different knowledge points play different roles in the distillation process, which inspires us to sample some reliable knowledge points from teacher GNNs *in a dynamic manner* to provide more additional supervision for training MLPs.

4.3. Knowledge-inspired Reliable Distillation

In this subsection, we first model the probability of each node being an informative and reliable knowledge point based on the knowledge quantification defined by Eq. (5). Next, we propose a knowledge-based sampling strategy to make full use of those reliable knowledge points as additional supervision for more reliable distillation into MLPs. A high-level overview of the proposed *Knowledge-inspired Reliable Distillation* (KRD) framework is shown in Fig. 5.

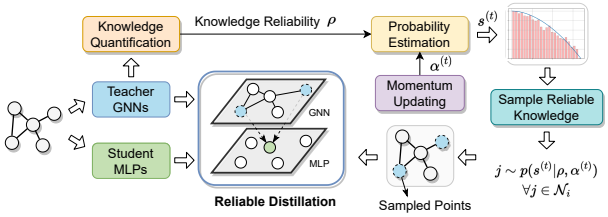


Figure 5. A high-level overview of the proposed KRD framework.

Sampling Probability Modeling. We aim to estimate the sampling probability of a knowledge point based on its quantified reliability. As shown in Fig. 6, we plot the histograms of *"True Positive"* sample density w.r.t the reliability metric ρ on two datasets (see **Appendix D** for more results), where the density has been min/max normalized. We model the sampling probability s_i of node v_i based on the metric ρ_i by a learnable power distribution (with power α), as follows:

$$p(s_i \mid \rho_i, \alpha) = 1 - (\frac{\rho_i}{\rho_M})^{\alpha}, \quad \forall v_i \in \mathcal{V} \tag{6}$$

where $\rho_M = \operatorname{argmax}_j \rho_j$. When the ground-truth labels are available, an optimal power α_{opt} can be directly fitted from histograms. However, the ground-truth labels are often unknown in practice, so we propose to combine the student MLPs $g_{\psi^{(t)}}(\cdot)$ with the pre-trained teacher GNNs $f_{\theta_{pre}}(\cdot)$ to model $p\left(\alpha^{(t)} \mid f_{\theta_{pre}}(\mathbf{A}, \mathbf{X}), g_{\psi^{(t)}}(\mathbf{A}, \mathbf{X})\right)$ at t-th epoch, which can be implementated by the following four steps: (1) initializing the power $\alpha^{(0)} = 1.0$; (2) constructing a histogram of sample density (predicted to be the same by both teacher GNNs and student MLPs) w.r.t the knowledge reliability metric ρ; (3) inferring a new power $\alpha_{new}^{(t)}$ by fitting the histogram; (4) updating power $\alpha^{(t-1)}$ in a dynamic momentum manner, which can be formulated as follows

$$\alpha^{(t)} \leftarrow \eta\alpha^{(t-1)} + (1-\eta) * \alpha_{new}^{(t)} \tag{7}$$

where η is the momentum updating rate. We provide the fitted curves with *fixed* and *learnable* powers in Fig. 6, which shows that the fitted distributions of learnable powers are more in line with the histogram. Moreover, we also include the results of fitting by Gaussian and exponential distributions as comparisons, but it shows that they do not work better. A quantitative comparison of different distribution fitting schemes has been provided in Table. 2, and the fitted results on more datasets are available in **Appendix D**.

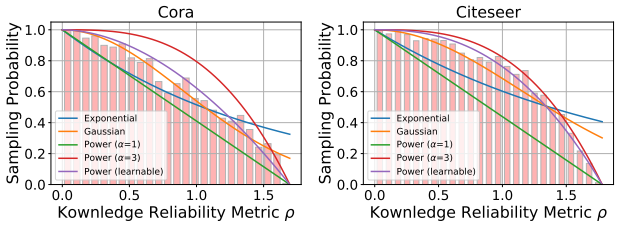


Figure 6. Histograms of *"True Positive"* sample density w.r.t the reliability metric ρ, as well as five distribution fitting schemes for modeling the sampling probability on the Cora and Citeseer datasets, where the density has been min/max normalized.

Knowledge-based Sampling. Next, we describe how to sample a set of reliable knowledge points as additional supervision for training student MLPs. Given any target node v_i, we first sample some highly reliable knowledge points $v_j \in \mathcal{N}_i$ from its neighborhood according to the sampling probability $p(s_j \mid \rho_i, \alpha^{(t)})$. Then, we take sampled knowledge points as multiple teachers and distill their knowledge into student MLPs as additional supervision through a *multi-teacher distillation* objective, which is defined as follows

$$\mathcal{L}_{\text{KRD}} = \mathop{\mathbb{E}}_{i} \mathop{\mathbb{E}}_{\substack{j \in \mathcal{N}_i \\ j \sim p(s_j \mid \rho_i, \alpha^{(t)})}} \mathcal{D}_{KL}\left(\sigma(\mathbf{z}_j^{(L)}/\tau), \sigma(\mathbf{h}_i^{(L)}/\tau)\right) \tag{8}$$

where τ is the distillation temperature coefficient.

4.4. Training Strategy

The pseudo-code of the KRD framework is summarized in Algorithm 1. To achieve GNN-to-MLP knowledge distillation, we first pre-train the teacher GNNs with the classification loss $\mathcal{L}_{\text{label}} = \frac{1}{|\mathcal{V}_L|} \sum_{i \in \mathcal{V}_L} \text{CE}\left(y_i, \sigma(\mathbf{h}_i^{(L)})\right)$, where $\text{CE}(\cdot)$ denotes the cross-entropy loss. Finally, the total objective function to distill reliable knowledge from the teacher GNNs into the student MLPs is defined as follows

$$\mathcal{L}_{\text{total}} = \frac{\lambda}{|\mathcal{V}_L|} \sum_{i \in \mathcal{V}_L} \mathcal{H}\big(y_i, \sigma(\mathbf{z}_i^{(L)})\big) + (1-\lambda)\big(\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{KRD}}\big)$$

where λ is the weight to balance the influence of the classification loss and knowledge distillation losses.

4.5. Time Complexity Analysis

It is noteworthy that the main computational burden introduced in this paper comes from additional reliable supervision as defined in Eq. (8). However, we sample reliable

knowledge points *in the neighborhood* instead of the entire set of nodes $\mathcal{V}$, which has reduced the time complexity from $\mathcal{O}(|\mathcal{V}^2|F)$ to less than $\mathcal{O}(|\mathcal{E}|F)$. The training time complexity of the KRD framework mainly comes from two parts: (1) GNN training $\mathcal{O}(|\mathcal{V}|dF + |\mathcal{E}|F)$ and (2) knowledge distillation $\mathcal{O}(|\mathcal{E}|F)$, where d and F are the dimensions of input and hidden spaces. The total time complexity $\mathcal{O}(|\mathcal{V}|dF+|\mathcal{E}|F)$ is linear w.r.t the number of nodes $|\mathcal{V}|$ and edges $|\mathcal{E}|$, which is in the same order as GCNs and GLNN.

Algorithm 1 Algorithm for KRD framework (Transductive)

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, Node Features: $\mathbf{X}$, # Epoch: E.
Output: Predicted labels $\mathcal{Y}_U$, MLP parameters $\{\mathbf{W}^l\}_{l=0}^{L-1}$.

```
Randomly initialize the parameters of GNNs and MLPs.
Pre-train the teacher GNNs until convergence by L_label.
Quantify the reliability of knowledge points by Eq. (5).
for t ∈ {1, 2, ···, E + 1} do
    Calculate the node representations from teacher GNNs and student MLPs by Eq. (1) and Eq. (2);
    Estimate the sampling probability by Eq. (6);
    Sample reliable knowledge points and calculate the multi-teacher distillation loss L_KRD by Eq. (8);
    Calculate the total loss L_total and update the parameters of student MLPs {W^l}_{l=0}^{L-1} by back propagation.
    Momentum updating the power α^(t) by Eq. (7).
end for
Predicted labels y_i ∈ Y_U for those unlabeled nodes V_U.
return Predicted labels Y_U and Parameters {W^l}_{l=0}^{L-1}.
```

5. Experiments

In this section, we evaluate KRD on seven real-world datasets by answering the following six questions. **Q1**: How effective is KRD in the transductive and inductive settings? Is KRD applicable to different teacher GNNs? **Q2:** How does KRD compare to other leading baselines on graph knowledge distillation? **Q3:** What happens if we model the sampling probability using other distribution functions? **Q4:** How does KRD perform by applying other heuristic knowledge sampling approach? **Q5:** Can KRD improve the predictive confidence of distilled MLPs? **Q6:** How do the two key hyperparameters λ and η influence the performance of KRD?

Dataset. The effectiveness of the KRD framework is evaluated on seven real-world datasets, including Cora (Sen et al., 2008), Citeseer (Giles et al., 1998), Pubmed (McCallum et al., 2000), Coauthor-CS, Coauthor-Physics, Amazon-Photo (Shchur et al., 2018), and ogbn-arxiv (Hu et al., 2020). A statistical overview of datasets is placed in **Appendix A**. Besides, each set of experiments is run five times with different random seeds, and the average accuracy and standard deviation are reported. Due to space limitations, we defer the implementation details and hyperparameter settings for each dataset to **Appendix C** and supplementary materials.

Baselines. Three basic components in knowledge distillation are (1) teacher model, (2) student model, and (3) distillation objective. As a model-agnostic framework, KRD can be combined with any teacher GNN architecture. In this paper, we consider three types of teacher GNNs, including GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2017). Besides, we adopt pure MLPs (with the same layer number L and size F as teacher GNNs) as the student model for a fair comparison. The focus of this paper is to provide more reliable self-supervision for GNN-to-MLP distillation. Thus, we only take GLNN (Zhang et al., 2021) as an important benchmark to demonstrate the necessity and effectiveness of additional supervision. Besides, we also compare KRD with some state-of-the-art graph distillation baselines in Table. 2, including CPF (Yang et al., 2021), RKD-MLP (Anonymous, 2023), FF-G2M (Wu et al., 2023), RDD (Zhang et al., 2020b), TinyGNN (Yan et al., 2020), LSP (Yang et al., 2020b), etc.

5.1. Classification Performance Comparison (Q1)

The reliable knowledge of three teahcer GNNs is distilled into student MLPs in the transductive and inductive settings. The experimental results on seven datasets are reported in Table. 1, from which we can make three observations: *(1)* Compared to the vanilla MLPs and intuitive KD baseline - GLNN, KRD performs significantly better than them in all cases, regardless of the datasets, teacher GNNs and evaluation settings. For example, KRD outperforms GLNN by 2.03% (GCN), 1.92% (SAGE), and 2.03% (GAT) averaged over seven datasets in the transductive setting, respectively. The superior performance of KRD demonstrates the effectiveness of providing more reliable self-supervision for GNN-to-MLP distillation. *(2)* The performance gain of KRD over GLNN is higher on the large-scale ogbn-arxiv dataset. We speculate that this is because the reliability of different knowledge points probably differ more in large-scale datasets, making those reliable knowledge points play a more important role. *(3)* It can be seen that KRD works much better in the transductive setting than in the inductive one, since there are more node features that can be used for training in the transductive setting, providing more reliable knowledge points to serve as additional self-supervision.

5.2. Comparision with Representative Baselines (Q2)

To answer **Q2**, we compare KRD with several representative graph knowledge distillation baselines, including both GNN-to-GNN and GNN-to-MLP distillation. As can be seen from the results reported in Table 2, KRD outperforms all other GNN-to-MLP baselines by a wide margin. More importantly, we are the first work to demonstrate *the promising potential of distilled MLPs to surpass distilled GNNs.* Even when compared with those state-of-the-art GNN-to-GNN distillation methods, KRD still shows competitive performance, ranking in the top two on 6 out of 7 datasets.

Table 1. Classificatiom accuracy ± std (%) on seven real-world datasets in both transductive and inductive settings, where three different GNN architectures (GCN, GraphSAGE, and GAT) have been considered as the teacher models. The best metrics are marked by **bold**.

Teacher	Student	Cora	Citeseer	Pubmed	Photo	CS	Physics	ogbn-arxiv	*Average*
Transductive Setting									
MLPs	-	$59.58_{\pm 0.97}$	$60.32_{\pm 0.61}$	$73.40_{\pm 0.68}$	$78.65_{\pm 1.68}$	$87.82_{\pm 0.64}$	$88.81_{\pm 1.08}$	$54.63_{\pm 0.84}$	-
GCN	-	$81.70_{\pm 0.96}$	$71.64_{\pm 0.34}$	$79.48_{\pm 0.21}$	$90.63_{\pm 1.53}$	$90.00_{\pm 0.58}$	$92.45_{\pm 0.53}$	$\mathbf{71.20}_{\pm 0.17}$	-
	GLNN	$82.20_{\pm 0.73}$	$71.72_{\pm 0.30}$	$80.16_{\pm 0.20}$	$91.42_{\pm 1.61}$	$92.22_{\pm 0.72}$	$93.11_{\pm 0.39}$	$67.76_{\pm 0.23}$	-
	KRD (ours)	$\mathbf{84.42}_{\pm 0.57}$	$\mathbf{74.86}_{\pm 0.58}$	$\mathbf{81.98}_{\pm 0.41}$	$\mathbf{92.21}_{\pm 1.44}$	$\mathbf{94.08}_{\pm 0.34}$	$\mathbf{94.30}_{\pm 0.46}$	$70.92_{\pm 0.21}$	-
	Improv.	2.22	3.14	1.82	0.79	1.86	1.19	3.16	2.03
GraphSAGE	-	$82.02_{\pm 0.94}$	$71.76_{\pm 0.49}$	$79.36_{\pm 0.45}$	$90.56_{\pm 1.69}$	$89.29_{\pm 0.77}$	$91.97_{\pm 0.91}$	$71.06_{\pm 0.27}$	-
	GLNN	$81.86_{\pm 0.88}$	$71.52_{\pm 0.54}$	$80.32_{\pm 0.38}$	$91.34_{\pm 1.46}$	$92.00_{\pm 0.57}$	$92.82_{\pm 0.93}$	$68.30_{\pm 0.19}$	-
	KRD (ours)	$\mathbf{84.60}_{\pm 0.76}$	$\mathbf{73.68}_{\pm 0.68}$	$\mathbf{81.60}_{\pm 0.33}$	$\mathbf{92.12}_{\pm 1.50}$	$\mathbf{93.93}_{\pm 0.40}$	$\mathbf{94.18}_{\pm 0.58}$	$\mathbf{71.50}_{\pm 0.25}$	-
	Improv.	2.74	2.16	1.28	0.78	1.93	1.36	3.20	1.92
GAT	-	$81.66_{\pm 1.04}$	$70.78_{\pm 0.60}$	$79.88_{\pm 0.85}$	$90.06_{\pm 1.38}$	$90.90_{\pm 0.37}$	$91.97_{\pm 0.58}$	$71.08_{\pm 0.19}$	-
	GLNN	$81.78_{\pm 0.75}$	$70.96_{\pm 0.86}$	$80.48_{\pm 0.47}$	$91.22_{\pm 1.45}$	$92.44_{\pm 0.41}$	$92.70_{\pm 0.56}$	$68.56_{\pm 0.22}$	-
	KRD (ours)	$\mathbf{84.12}_{\pm 0.39}$	$\mathbf{73.06}_{\pm 0.59}$	$\mathbf{82.02}_{\pm 0.56}$	$\mathbf{92.13}_{\pm 1.48}$	$\mathbf{94.35}_{\pm 0.29}$	$\mathbf{94.19}_{\pm 0.50}$	$\mathbf{71.45}_{\pm 0.26}$	-
	Improv.	2.34	2.10	1.54	0.91	1.91	1.49	2.89	1.88
Inductive Setting									
MLPs	-	$59.20_{\pm 1.26}$	$60.16_{\pm 0.87}$	$73.26_{\pm 0.83}$	$79.02_{\pm 1.42}$	$87.90_{\pm 0.58}$	$89.10_{\pm 0.90}$	$54.46_{\pm 0.52}$	-
GCN	-	$\mathbf{79.30}_{\pm 0.49}$	$71.46_{\pm 0.36}$	$78.10_{\pm 0.51}$	$89.32_{\pm 1.63}$	$90.07_{\pm 0.60}$	$92.05_{\pm 0.78}$	$\mathbf{70.88}_{\pm 0.35}$	-
	GLNN	$71.24_{\pm 0.55}$	$70.76_{\pm 0.30}$	$80.16_{\pm 0.73}$	$89.92_{\pm 1.34}$	$92.08_{\pm 0.98}$	$92.89_{\pm 0.88}$	$60.92_{\pm 0.31}$	-
	KRD (ours)	$73.78_{\pm 0.55}$	$\mathbf{71.80}_{\pm 0.41}$	$\mathbf{81.48}_{\pm 0.29}$	$\mathbf{90.37}_{\pm 1.79}$	$\mathbf{93.15}_{\pm 0.43}$	$\mathbf{93.86}_{\pm 0.55}$	$62.85_{\pm 0.32}$	-
	Improv.	2.54	1.04	1.32	0.45	1.07	0.97	2.93	1.47
GraphSAGE	-	$\mathbf{79.56}_{\pm 0.47}$	$70.24_{\pm 0.62}$	$79.40_{\pm 0.48}$	$89.76_{\pm 1.51}$	$89.96_{\pm 0.56}$	$91.79_{\pm 0.69}$	$\mathbf{71.13}_{\pm 0.32}$	-
	GLNN	$71.82_{\pm 0.35}$	$70.26_{\pm 0.71}$	$80.46_{\pm 0.34}$	$89.94_{\pm 1.70}$	$92.06_{\pm 0.69}$	$92.97_{\pm 0.94}$	$60.46_{\pm 0.26}$	-
	KRD (ours)	$73.48_{\pm 0.43}$	$\mathbf{70.94}_{\pm 0.49}$	$\mathbf{81.36}_{\pm 0.51}$	$\mathbf{90.37}_{\pm 1.79}$	$\mathbf{92.96}_{\pm 0.44}$	$\mathbf{93.91}_{\pm 0.63}$	$62.56_{\pm 0.33}$	-
	Improv.	1.66	0.68	0.90	0.43	0.90	0.94	2.10	1.09
GAT	-	$\mathbf{79.96}_{\pm 0.63}$	$69.58_{\pm 0.43}$	$79.02_{\pm 0.43}$	$90.54_{\pm 1.73}$	$90.50_{\pm 0.97}$	$91.99_{\pm 1.08}$	$\mathbf{70.65}_{\pm 0.23}$	-
	GLNN	$71.10_{\pm 0.86}$	$70.20_{\pm 0.69}$	$81.28_{\pm 0.58}$	$90.57_{\pm 1.59}$	$92.15_{\pm 0.75}$	$93.17_{\pm 0.92}$	$60.38_{\pm 0.30}$	-
	KRD (ours)	$72.48_{\pm 0.53}$	$\mathbf{70.64}_{\pm 0.38}$	$\mathbf{82.00}_{\pm 0.65}$	$\mathbf{91.10}_{\pm 1.69}$	$\mathbf{93.23}_{\pm 0.48}$	$\mathbf{94.02}_{\pm 0.73}$	$62.16_{\pm 0.24}$	-
	Improv.	1.38	0.44	0.72	0.53	1.08	0.85	1.78	0.97

Table 2. Performance comparison with leading graph distillation algorithms, where **bold** and underline denote the best and second metrics. The experiments are conducted by adopting GCN as the teacher in the transductive setting (same for Table. 3, Table. 4, Fig. 7, and Fig. 8).

Category	Method	Cora	Citeseer	Pubmed	Photo	CS	Physics	ogbn-arxiv	*Avg. Rank*
Vanilla	MLPs	$59.58_{\pm 0.97}$	$60.32_{\pm 0.61}$	$73.40_{\pm 0.68}$	$78.65_{\pm 1.68}$	$87.82_{\pm 0.64}$	$88.81_{\pm 1.08}$	$54.63_{\pm 0.84}$	12.0
	Vanilla GCNs	$81.70_{\pm 0.96}$	$71.64_{\pm 0.34}$	$79.48_{\pm 0.21}$	$90.63_{\pm 1.53}$	$90.00_{\pm 0.58}$	$92.45_{\pm 0.53}$	$71.20_{\pm 0.17}$	10.1
GNN-to-GNN	LSP	$82.70_{\pm 0.43}$	$72.68_{\pm 0.62}$	$80.86_{\pm 0.50}$	$91.74_{\pm 1.42}$	$92.56_{\pm 0.45}$	$92.85_{\pm 0.46}$	$71.57_{\pm 0.25}$	7.4
	GNN-SD	$82.54_{\pm 0.36}$	$72.34_{\pm 0.55}$	$80.52_{\pm 0.37}$	$91.83_{\pm 1.58}$	$91.92_{\pm 0.51}$	$93.22_{\pm 0.66}$	$70.90_{\pm 0.23}$	8.3
	TinyGNN	$83.10_{\pm 0.53}$	$73.24_{\pm 0.72}$	$81.20_{\pm 0.44}$	$92.03_{\pm 1.49}$	$93.78_{\pm 0.38}$	$93.70_{\pm 0.56}$	$72.18_{\pm 0.27}$	4.7
	RDD	$83.68_{\pm 0.40}$	$73.64_{\pm 0.50}$	$\underline{81.74}_{\pm 0.44}$	$92.18_{\pm 1.45}$	$\mathbf{94.20}_{\pm 0.48}$	$\underline{94.14}_{\pm 0.39}$	$\underline{72.34}_{\pm 0.17}$	2.1
	FreeKD	$83.84_{\pm 0.47}$	$\underline{73.92}_{\pm 0.47}$	$81.48_{\pm 0.38}$	$\mathbf{92.38}_{\pm 1.54}$	$93.65_{\pm 0.43}$	$93.87_{\pm 0.48}$	$\mathbf{72.50}_{\pm 0.29}$	2.9
GNN-to-MLP	GLNN	$82.20_{\pm 0.73}$	$71.72_{\pm 0.30}$	$80.16_{\pm 0.20}$	$91.42_{\pm 1.61}$	$92.22_{\pm 0.72}$	$93.11_{\pm 0.39}$	$67.76_{\pm 0.23}$	9.7
	CPF	$83.56_{\pm 0.48}$	$72.98_{\pm 0.60}$	$81.54_{\pm 0.47}$	$91.70_{\pm 1.50}$	$93.42_{\pm 0.48}$	$93.47_{\pm 0.41}$	$69.05_{\pm 0.18}$	6.4
	RKD-MLP	$82.68_{\pm 0.45}$	$73.42_{\pm 0.45}$	$81.32_{\pm 0.32}$	$91.28_{\pm 1.48}$	$93.16_{\pm 0.64}$	$93.26_{\pm 0.37}$	$69.87_{\pm 0.25}$	7.3
	FF-G2M	$\underline{84.06}_{\pm 0.43}$	$73.85_{\pm 0.51}$	$81.62_{\pm 0.37}$	$91.84_{\pm 1.42}$	$93.35_{\pm 0.55}$	$93.59_{\pm 0.43}$	$69.64_{\pm 0.26}$	4.9
	KRD (ours)	$\mathbf{84.42}_{\pm 0.57}$	$\mathbf{74.86}_{\pm 0.58}$	$\mathbf{81.98}_{\pm 0.41}$	$\underline{92.21}_{\pm 1.44}$	$\underline{94.08}_{\pm 0.34}$	$\mathbf{94.30}_{\pm 0.46}$	$70.92_{\pm 0.21}$	2.1

5.3. Evaluation on Distribution Fitting Function (Q3)

To evaluate the effectiveness of different distribution fitting functions and the momentum updating defined in Eq. (7), we compare the learnable power distribution defined in Eq. (6) with the other four schemes: (A) exponential distribution $p(s_i \mid \rho_i, \alpha) = \alpha \exp^{-\alpha \cdot \frac{\rho_i}{\rho_M}}$ with learnable rate α; (B) Gaussian distribution $p(s_i \mid \rho_i, \alpha) = \mathcal{N}(0, \alpha)$ with learnable variance α; (C) power distribution with fixed power $\alpha = 1$; and (D) power distribution with fixed power $\alpha = 3$. From the results reported in Table. 3, it can be seen that (1) when modeling the sampling probability with power distribution, the *learnable* power is consistently better than the *fixed* power on all datasets, and (2) the exponential,

Gaussian and power distributions perform differently on different datasets, but the power distribution can achieve better overall performance than the other two distributions.

Table 3. Performance comparison of different distribution fitting functions and the momentum updating of Eq. (7), where **bold** and underline denote the best and second metrics on each dataset.

Methods	Cora	Citeseer	Pubmed	Photo	CS	Physics
Exponential	83.30	73.84	81.10	92.12	93.80	93.63
Gaussian	84.12	74.52	81.56	92.10	**94.15**	94.08
Power (fixed α=1)	83.84	74.18	81.44	92.04	93.93	93.93
Power (fixed α=3)	83.54	74.32	81.34	91.95	94.01	93.75
Power (learnable)	**84.42**	**74.86**	**81.98**	**92.21**	94.08	**94.30**

5.4. Evaluation on Knowledge Sampling Strategy (Q4)

To explore how different sampling strategies influence the performance of distillation, we compare our knowledge-inspired sampling with other three schemes: (A) *Non-sampling*: directly takes all nodes in the neighborhood as additional supervision and distills their knowledge into the student MLPs; (B) *Random Sampling*: randomly sampling knowledge points with 50% probability in the neighborhood for distillation; (C) *Entropy-based Sampling*: performing min/max normalization on the information entropy of each knowledge point to [0-1], and then sampling by taking entropy as sampling probability. Besides, we also include the performance of vanilla GCN and GLNN as a comparison. We can observe from Table. 3 that (1) Both non-sampling and random sampling help to significantly improve the performance of GLNN, again demonstrating the importance of providing additional supervision for training student MLPs. (2) Entropy- and knowledge-based sampling performs much better than non-sampling and random sampling, suggesting that different knowledge plays different roles during distillation. (3) Compared with entropy-based sampling, knowledge-based sampling fully takes into account the contextual information of the neighborhood as explained in Sec. 4.2, and thus shows better overall performance.

Table 4. Performance comparison of different sampling strategies, where the best/second metrics are marked in **bold** and underline.

Methods	Cora	Citeseer	Pubmed	Photo	CS	Physics
Vanilla GCN	81.70	71.64	79.48	90.63	90.00	92.45
GLNN	82.54	71.92	80.16	90.48	91.48	92.81
Non-sampling	83.26	73.58	80.74	91.45	93.04	93.42
Random	82.42	73.10	81.08	91.28	92.57	93.74
Entropy-based	83.64	73.74	81.32	91.58	93.35	93.63
Knowledge-based	**84.42**	**74.86**	**81.98**	**92.21**	**94.08**	**94.30**

5.5. Evaluation on Confidence Distribution (Q5)

To explore whether providing additional reliable supervision can improve the predictive confidence of distilled MLPs, we compare the confidence distribution of KRD with that of GLNN in Fig. 7 on four datasets. It can be seen that the predictive confidence of student MLPs in GLNN (optimized with only the distillation term defined by Eq. (3) is indeed not very high. Instead, KRD provides additional reliable self-supervision defined in Eq. (8), which helps to greatly improve the predictive confidence of student MLPs.

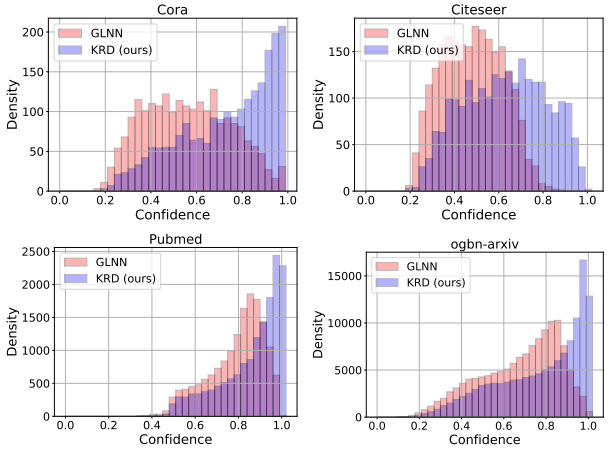
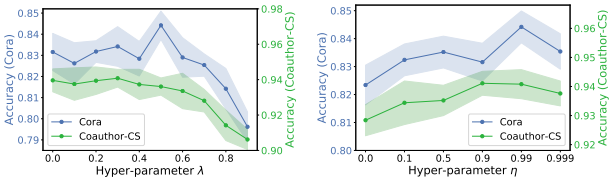


Figure 7. Confidence distribution of the distilled MLPs in GLNN and KRD on four datasets, where GCN is adopted as the teacher.

5.6. Evaluation on Hyperparameter Sensitivity (Q6)

We provide sensitivity analysis for two hyperparameters, loss weights λ and momentum updating rate η in Fig. 8(a) and Fig. 8(b), from which we observe that (1) setting the loss weight λ too large weakens the contribution of the distillation term, leading to poor performance; (2) too large or small η are both detrimental to modeling sampling probability and extracting informative knowledge. In practice, $\eta = 0.9, 0.99$ often yields pretty good performance. In practice, we can determine λ and η by selecting the model with the highest accuracy on the validation set by the grid search.



(a) Loss weight λ (b) Momentum rate η

Figure 8. Hyperparameter sensitivity analysis on λ and η.

6. Conclusion

In this paper, we identified a potential *under-confidence* problem for GNN-to-MLP distillation, and more importantly, we described in detail what it represents, how it arises, what impact it has, and how to deal with it. To address this problem, we design a perturbation invariance-based metric to quantify the reliability of knowledge in GNNs, based on which we propose a *Knowledge-inspired Reliable Distillation* (KRD) framework to make full use of those reliable knowledge points as additional supervision for training MLPs. Limitations still exist; for example, combining our work with other more powerful and expressive teacher/student models may be another promising direction.

7. Acknowledgement

This work was supported by National Key R&D Program of China (No. 2022ZD0115100), National Natural Science Foundation of China Project (No. U21A20427), and Project (No. WU2022A009) from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University.

References

Anonymous. Double wins: Boosting accuracy and efficiency of graph neural networks by reliable knowledge distillation. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NGIFt6BNvLe. under review.

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, volume 14, pp. 585–591, 2001.

Chen, Y., Bian, Y., Xiao, X., Rong, Y., Xu, T., and Huang, J. On self-distilling graph neural network. *arXiv preprint arXiv:2011.02255*, 2020.

Feng, K., Li, C., Yuan, Y., and Wang, G. Freekd: Free-direction knowledge distillation for graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 357–366, 2022.

Giles, C. L., Bollacker, K. D., and Lawrence, S. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, 1998.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Jia, Z., Lin, S., Ying, R., You, J., Leskovec, J., and Aiken, A. Redundancy-free computation for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 997–1005, 2020.

Joshi, C. K., Liu, F., Xun, X., Lin, J., and Foo, C.-S. On representation knowledge distillation for graph neural networks. *arXiv preprint arXiv:2111.04964*, 2021.

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

Lassance, C., Bontonou, M., Hacene, G. B., Gripon, V., Tang, J., and Ortega, A. Deep geometric knowledge distillation with graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8484–8488. IEEE, 2020.

Liu, M., Gao, H., and Ji, S. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 338–348, 2020.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Ren, Y., Ji, J., Niu, L., and Lei, M. Multi-task self-distillation for graph-based semi-supervised learning. *arXiv preprint arXiv:2112.01174*, 2021.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

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

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

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., He, T., Karypis, G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Wu, L., Lin, H., Gao, Z., Tan, C., Li, S., et al. Graphmixup: Improving class-imbalanced node classification on graphs by self-supervised context prediction. *arXiv preprint arXiv:2106.11133*, 2021a.

Wu, L., Lin, H., Gao, Z., Tan, C., Li, S., et al. Self-supervised on graphs: Contrastive, generative, or predictive. *arXiv preprint arXiv:2105.07342*, 2021b.

Wu, L., Lin, H., Huang, Y., and Li, S. Z. Knowledge distillation improves graph structure augmentation for graph neural networks. In *Advances in Neural Information Processing Systems*, 2022a.

Wu, L., Xia, J., Lin, H., Gao, Z., Liu, Z., Zhao, G., and Li, S. Z. Teaching yourself: Graph self-distillation on neighborhood for node classification. *arXiv preprint arXiv:2210.02097*, 2022b.

Wu, L., Lin, H., Huang, Y., Fan, T., and Li, S. Z. Extracting low-/high- frequency knowledge from graph neural networks and injecting it into mlps: An effective gnn-to-mlp distillation framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Yan, B., Wang, C., Guo, G., and Lou, Y. Tinygnn: Learning efficient graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1848–1856, 2020.

Yang, C., Wang, R., Yao, S., Liu, S., and Abdelzaher, T. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020a.

Yang, C., Liu, J., and Shi, C. Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework. In *Proceedings of the Web Conference 2021*, pp. 1227–1237, 2021.

Yang, Y., Qiu, J., Song, M., Tao, D., and Wang, X. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7074–7083, 2020b.

Zhang, H., Lin, S., Liu, W., Zhou, P., Tang, J., Liang, X., and Xing, E. P. Iterative graph self-distillation. *arXiv preprint arXiv:2010.12609*, 2020a.

Zhang, S., Liu, Y., Sun, Y., and Shah, N. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*, 2021.

Zhang, W., Miao, X., Shao, Y., Jiang, J., Chen, L., Ruas, O., and Cui, B. Reliable data distillation on graph convolutional network. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 1399–1414, 2020b.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

A. Dataset Statistics

Seven publicly available real-world graph datasets have been used to evaluate the proposed KRD framework. An overview summary of the statistical characteristics of these datasets is given in Table. A1. For the three small-scale datasets, namely Cora, Citeseer, and Pubmed, we follow the data splitting strategy in (Kipf & Welling, 2016). For the three large-scale datasets, including Coauthor-CS, Coauthor-Physics, and Amazon-Photo, we follow (Zhang et al., 2021; Yang et al., 2021) to randomly split the data into train/val/test sets, and each random seed corresponds to a different data splitting. For the ogbn-arxiv dataset, we use the public data splits provided by the authors (Hu et al., 2020).

Table A1. Statistical information of the datasets.

Dataset	Cora	Citeseer	Pubmed	Photo	CS	Physics	ogbn-arxiv
# Nodes	2708	3327	19717	7650	18333	34493	169343
# Edges	5278	4614	44324	119081	81894	247962	1166243
# Features	1433	3703	500	745	6805	8415	128
# Classes	7	6	3	8	15	5	40
Label Rate	5.2%	3.6%	0.3%	2.1%	1.6%	0.3%	53.7%

B. Implementation Details

The following hyperparameters are set the same for all datasets: Epoch E = 500, noise variance $\delta = 1.0$, and momentum rate $\eta = 0.99$ (0.9 for ogb-arxiv). The other dataset-specific hyperparameters are determined by an AutoML toolkit NNI with the hyperparameter search spaces as: hidden dimension $F = \{128, 256, 512, 1024, 2048\}$, layer number $L = \{2, 3\}$, distillation temperature $\tau = \{0.8, 0.9, 1.0, 1.1, 1.2\}$, loss weight $\alpha = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, learning rate $lr = \{0.001, 0.005, 0.01\}$, and weight decay $decay = \{0.0, 0.0005, 0.001\}$. For a fairer comparison, the model with the highest validation accuracy is selected for testing. Besides, the best hyperparameter choices of each setting are available in the supplementary. Moreover, the experiments on both baselines and our approach are implemented based on the standard implementation in the DGL library (Wang et al., 2019) using the PyTorch 1.6.0 with Intel(R) Xeon(R) Gold 6240R @ 2.40GHz CPU and NVIDIA V100 GPU.

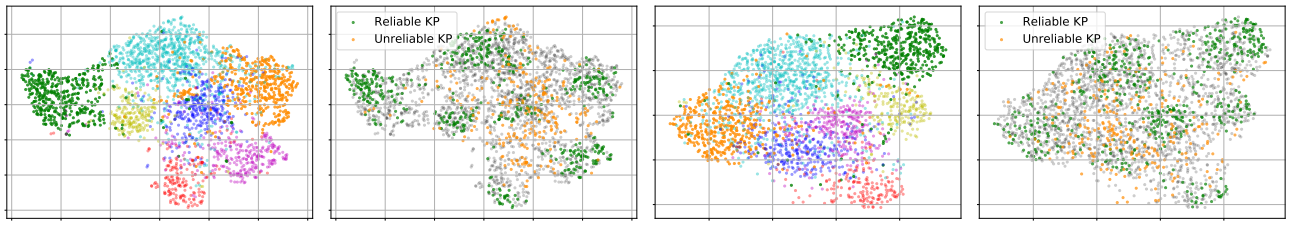
Transductive *vs.* Inductive. We evaluate our model under two evaluation settings: transductive and inductive. Their main difference is whether to use the test data for training. Specifically, we partition node features and labels into three disjoint sets, i.e., $\mathbf{X} = \mathbf{X}^L \sqcup \mathbf{X}^U_{obs} \sqcup \mathbf{X}^U_{ind}$, and $\mathcal{Y} = \mathcal{Y}^L \sqcup \mathcal{Y}^U_{obs} \sqcup \mathcal{Y}^U_{ind}$. Concretely, the input and output of two settings are: (1) *Transductive*: training on $\mathbf{X}$ and $\mathcal{Y}^L$ and testing on $(\mathbf{X}^U, \mathcal{Y}^U)$. (2) *Inductive*: training on $\mathbf{X}^L \sqcup \mathbf{X}^U_{obs}$ and $\mathcal{Y}^L$ and testing on $(\mathbf{X}^U_{ind}, \mathcal{Y}^U_{ind})$ (Anonymous, 2023).

C. More Results on Spatial Distribution

The embeddings of teacher GNNs and student MLPs on the Cora dataset are visualized in Fig. A1(a)(c). Then, we mark the knowledge points with the reliability ranked in the top 20% and bottom 10% as green and orange in Fig. A1(b)(d), respectively. It can be seen that most reliable knowledge points are distributed around the class centers, while those unreliable ones are distributed at the class boundaries.

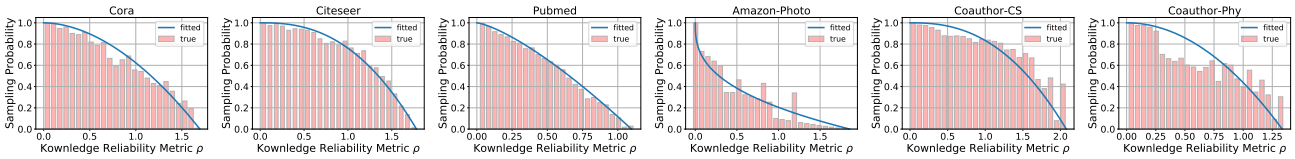
D. More Results on Fitted Distributions

We report histograms of *"True Positive"* sample density w.r.t the reliability metric ρ as well as the fitted distributions in Fig. A2. It can be seen that the fitted distributions of the sampling probability closely matches the true histograms.



(a) Visualizations in GNNs (b) Spatial Distribution in GNNs (c) Visualizations in MLPs (d) Spatial Distribution in MLPs

Figure A1. ***(a)(c)*** Visualizations of the embeddings of teacher GNNs and student MLPs for two classes on Cora. ***(b)(d)*** Spatial distribution of knowledge points with the reliability ranked in the top 20% and bottom 10%, which are marked in green and orange, respectively.



(a) Cora (b) Citeseer (c) Pubmed (d) Amazon-Photo (e) Coauthor-CS (f) Coauthor-Physics

Figure A2. Histograms of *"True Positive"* sample density w.r.t the reliability metric ρ, as well as the fitted distributions (by learnable power distribution) of the sampling probability on six graph datasets, where the sample density has been min/max normalized to [0, 1].