ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

A Collaborative Distillation Framework for Graph Neural Networks

Paul Agbaje[1], Akajyoti Mitra[1], Afia Anjum[1], Pranali Khose[1], Ebelechukwu Nwafor[2], Habeeb Olufowobi[1]

1- University of Texas at Arlington, Arlington, TX, USA

2- Villanova University, Villanova, PA, USA

{pauloluwatowoju.agbaje, habeeb.olufowobi}@uta.edu

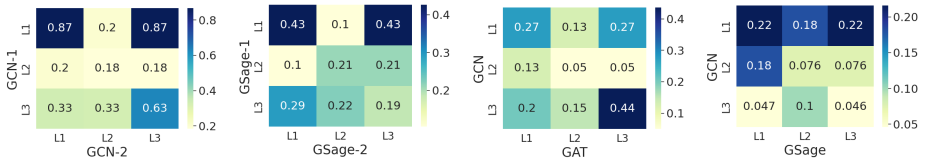
Abstract. Graph Neural Networks (GNNs) power applications such as content recommendation, knowledge graph reasoning, and social network analysis, where modeling both structure and features is essential. Knowledge Distillation (KD) enables transferring knowledge from large GNNs to compact models for efficient deployment, yet most approaches rely on a pre-trained teacher. We propose a mutual learning framework in which shallow GNNs collaboratively distill knowledge by iteratively exchanging predictions during training. The framework integrates adaptive logit weighting to balance peer influence and entropy enhancement to promote exploration and prevent early convergence. Experiments on multiple benchmark datasets show that our approach improves GNN performance and that the learned knowledge can be effectively transferred to lightweight graph-less models, offering a scalable alternative for graph learning.

1 Introduction

Graph Neural Networks (GNNs) are powerful tools for learning representations of structured data and supporting tasks such as semi-supervised node classification [1, 2]. They are especially effective in web and social network applications, such as recommendation, knowledge graph reasoning, and community analysis, where modeling both feature and structural dependencies is essential. GNNs rely on message passing, enabling each node to aggregate information from its neighbors and encode local and global graph structure. Knowledge distillation (KD) has been widely used to balance model size and accuracy by transferring knowledge from large teacher models to smaller students through logit or feature imitation. Although KD from GNNs to MLPs can achieve near-teacher performance [3], applying KD to GNNs remains challenging due to their shallow depth and over-smoothing tendencies [4].

This work departs from the conventional teacher–student paradigm by proposing a mutual learning strategy [5] to train compact but effective GNN whenever a pre-trained teacher GNN is unavailable. In standard KD, a strong teacher delivers significant performance gains, yet such teachers are not always available. Instead, we examine whether any measurable performance improvement can be achieved through peer-to-peer training among lightweight GNNs. Inspired by Guo et al. [6], who show that GNNs encode complementary information through diverse aggregation schemes, we enable peers to exchange predictive signals and

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.



(a) GCN-GCN (b) GSage-GSage (c) GCN-GAT (d) GCN-GSage

Fig. 1: Centered kernel alignment similarity between model layers 1, 2, and 3. (a) and (b) show the similarity between models of the same architecture but different initializations. (c) and (d) show the similarity between models of different architectures and different initializations.

increase posterior entropy, promoting generalization while maintaining diversity through independent initialization.

Although GNNs dominate graph learning, MLPs still serve latency-critical industrial systems [3]. In our work, we focus on *node classification* and show that knowledge gained through GNN mutual learning can be distilled into lightweight MLPs without significant loss in performance. We introduce Graph Mutual Learning (GML), a framework that jointly trains GNN peers to exchange informative knowledge. Our contributions are as follows: (1) We develop a mutual-learning framework for GNNs that improves individual model performance through collaboration. (2) We introduce adaptive logit weighting to prioritize salient information. (3) We adapt GML for KD, transferring collective representations to MLPs. (4) Experiments on benchmark datasets show that GML consistently boosts the performance of shallow GNNs and student MLPs.

2 Exploring Graph Neural Network Architectures for Mutual Learning

We examine how GNN architecture and random initialization shape learned representations using Centered Kernel Alignment (CKA) [7]. Layer-wise similarities are computed for 3-layer GCN, GAT, and GraphSage models on Citeseer, with representations obtained via average-pooled node embeddings (Figure 1). For identical architectures under different seeds, GCN yields CKA scores of 0.87/0.18/0.63, while GraphSage produces 0.43/0.21/0.19, showing high variance even within the same model family. Cross-architecture similarity is lower—GCN–GAT (0.27/0.05/0.44) and GCN–GraphSage (0.22/0.076/0.046)—highlighting distinct feature encoding behaviors. Overall, both architecture and initialization induce diverse embeddings, motivating our use of mutual learning across differently initialized and structurally varied GNNs to encourage complementary feature exchange and improve generalization.

3 Method

Let $G = (V, E, X)$ be a graph with $|V| = N$ nodes, attributes $X \in \mathbb{R}^{N \times D}$, and labels Y for node or graph classification with C classes. For a node v, a model

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

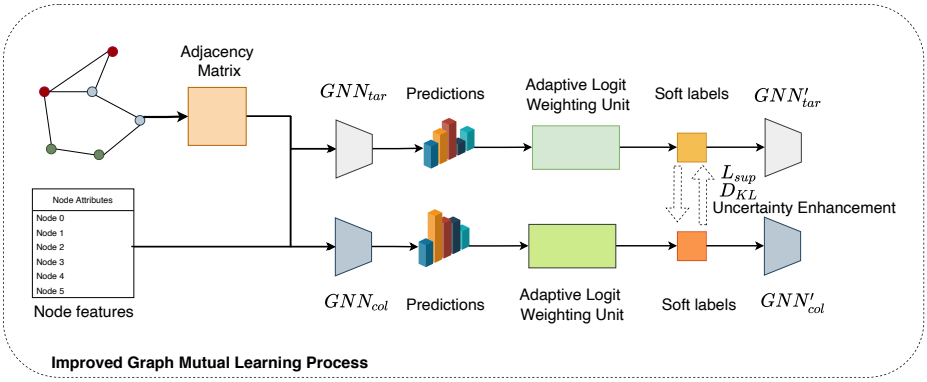


Fig. 2: GML architecture with two GNN models, GNN_{tar} and GNN_{col}, training together.

θ_j outputs logits $z_j^{v,c}$ and temperature (T_v)–scaled posteriors

$$p_j^c(v) = \frac{\exp\left(z_j^{v,c}/T_v\right)}{\sum_{c'=1}^{C} \exp\left(z_j^{v,c'}/T_v\right)}. \tag{1}$$

Our goal is to improve generalization by enabling peers to exchange knowledge via their posteriors as shown in Figure 2. In the two–peer setting (target θ_{tar} and collaborator θ_{col}), each model minimizes a local cross-entropy (CE) loss L_{sup}, and a Kullback–Leibler (KL) alignment term D_{KL}:

$$L_{\text{tar}} = L_{\text{sup}_{\text{tar}}} + D_{\text{KL}}\big(p_{\text{col}} \| p_{\text{tar}}\big), \tag{2}$$

$$L_{\text{col}} = L_{\text{sup}_{\text{col}}} + D_{\text{KL}}\big(p_{\text{tar}} \| p_{\text{col}}\big). \tag{3}$$

For K peers, each model aligns to all others by averaging KL terms:

$$L_{\text{tar}} = L_{\text{sup}_{\text{tar}}} + \frac{1}{K-1} \sum_{k \neq \text{tar}} D_{\text{KL}}(p_k \| p_{\text{tar}}), \tag{4}$$

$$L_{\text{col}} = L_{\text{sup}_{\text{col}}} + \frac{1}{K-1} \sum_{k \neq \text{col}} D_{\text{KL}}(p_k \| p_{\text{col}}). \tag{5}$$

Adaptive Logit Weighting. To prioritize transferable knowledge, each peer j learns class weights from its confidence. Let $H(p_j) \in \mathbb{R}^{N \times 1}$ be (negative) entropy over nodes, and $\chi_j \in \mathbb{R}^{N \times h}$, $\phi_j \in \mathbb{R}^{h \times C}$ be learnable parameters. Define

$$\sigma_j = H(p_j)^\top \chi_j \phi_j \in \mathbb{R}^{1 \times C}, \qquad W_j^c = \frac{\exp(\sigma_j^c)}{\sum_{c'=1}^{C} \exp(\sigma_j^{c'})}. \tag{6}$$

We reweight predictions with the Hadamard product $p'_j = p_j \cdot W_j$ and train jointly via

$$L_{\text{tar}} = L_{\text{sup}_{\text{tar}}} + D_{\text{KL}}(p'_{\text{col}} \| p'_{\text{tar}}) + \beta(\|\chi_{\text{tar}}\|_1 + \|\phi_{\text{tar}}\|_1)\,, \tag{7}$$

$$L_{\text{col}} = L_{\text{sup}_{\text{col}}} + D_{\text{KL}}(p'_{\text{tar}} \| p'_{\text{col}}) + \beta(\|\chi_{\text{col}}\|_1 + \|\phi_{\text{col}}\|_1)\,, \tag{8}$$

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Enhancing Uncertainty. Minimizing $D_{\mathrm{KL}}(p_2 \| p_1)$ both matches predictions and shapes entropy; without control, it can induce overconfident and peaky posteriors. We add a confidence penalty [8] to keep outputs high-entropy and generalizable:

$$L_{\mathrm{tar}} = L_{\mathrm{sup}_{\mathrm{tar}}} + D_{\mathrm{KL}}(p_{\mathrm{col}} \| p_{\mathrm{tar}}) - \gamma\, H(p_{\mathrm{tar}}|v), \tag{9}$$

$$L_{\mathrm{col}} = L_{\mathrm{sup}_{\mathrm{col}}} + D_{\mathrm{KL}}(p_{\mathrm{tar}} \| p_{\mathrm{col}}) - \gamma\, H(p_{\mathrm{col}}|v). \tag{10}$$

Unified Objective. With all components enabled for a K-peer cohort, the target's objective is

$$L_{\mathrm{tar}} = L_{\mathrm{sup}_{\mathrm{tar}}} + \frac{1}{K-1} \sum_{k \neq \mathrm{tar}} D_{\mathrm{KL}}(p'_k \| p'_{\mathrm{tar}}) - \gamma\, H(p_{\mathrm{tar}}|v) + \beta(\|\chi_{\mathrm{tar}}\|_1 + \|\phi_{\mathrm{tar}}\|_1)\,, \tag{11}$$

with an analogous form for each peer. Supervision anchors labels, KL drives peer alignment, entropy regularization prevents overconfidence, and adaptive weighting concentrates transfer on salient classes.

4 Experiments

Table 1: Classification results of mutual learning for node and graph tasks. Ind denotes the target model trained independently. S and C represent GraphSage and GCN, respectively; e.g., GraphSage-GCN-S indicates GML with GraphSage as the target. GML-Co uses both adaptive logit weighting and uncertainty enhancement, while GML-W and GML-C apply only weighting or only uncertainty, respectively.

		Node Classification			Graph Classification		
Models	Methods	Cora	Citeseer	PubMed	Ogbg-molbace	Ogbg-molbbbp	PROTEINS
Sage-GCN-S	Ind	86.58 ± 0.68	76.57 ± 1.24	89.03 ± 0.50	77.43 ± 1.90	84.20 ± 0.63	69.64 ± 1.25
	GML	87.00 ± 0.52	**77.80 ± 0.51**	89.66 ± 0.34	78.32 ± 0.94	84.79 ± 0.85	69.28 ± 1.54
	GML-Co	87.32 ± 0.50	75.99 ± 0.66	**90.19 ± 0.30**	**79.03 ± 1.67**	84.07 ± 1.1	69.76 ± 0.79
	GML-W	87.49 ± 0.40	76.93 ± 0.50	89.17 ± 0.25	77.79 ± 0.37	**85.38 ± 0.94**	70.24 ± 0.91
	GML-C	**88.69 ± 0.38**	76.13 ± 0.46	90.17 ± 0.13	78.23 ± 1.23	85.05 ± 0.50	**70.96 ± 1.08**
GAT-Sage-S	Ind	86.58 ± 0.68	76.57 ± 1.24	89.03 ± 0.5	77.43 ± 1.90	84.20 ± 0.63	69.64 ± 1.25
	GML	88.55 ± 0.24	77.37 ± 0.59	89.7 ± 0.24	78.23 ± 1.34	84.07 ± 1.15	69.4 ± 0.99
	GML-Co	87.88 ± 0.68	76.61 ± 0.62	90.13 ± 0.30	78.58 ± 1.55	**84.72 ± 1.2**	69.64 ± 1.45
	GML-W	87.36 ± 2.79	76.45 ± 0.75	89.20 ± 0.20	**78.76 ± 1.13**	84.39 ± 0.82	70.36 ± 0.79
	GML-C	**88.57 ± 0.41**	**77.45 ± 0.6**	**90.24 ± 0.23**	78.05 ± 1.27	84.07 ± 2.03	**70.36 ± 1.44**
GCN-GAT-C	Ind	88.87 ± 0.10	76.55 ± 0.21	87.20 ± 0.05	75.93 ± 1.64	84.52 ± 0.27	71.45 ± 1.32
	GML	89.24 ± 0.39	76.71 ± 0.23	87.20 ± 0.18	73.54 ± 1.48	84.26 ± 0.9	71.20 ± 1.16
	GML-Co	89.06 ± 0.24	**76.73 ± 0.27**	87.33 ± 0.10	75.22 ± 2.00	84.26 ± 0.52	71.20 ± 1.62
	GML-W	89.24 ± 0.26	76.63 ± 0.17	87.15 ± 0.10	**76.48 ± 0.85**	**84.59 ± 0.52**	71.08 ± 0.60
	GML-C	**89.26 ± 0.31**	76.61 ± 0.23	**87.45 ± 0.14**	74.87 ± 1.61	84.33 ± 0.82	**71.81 ± 0.27**

Setup. We evaluate GML using benchmark datasets for node (Cora, Citeseer, PubMed) and graph (PROTEINS, Ogbg-molbace, Ogbg-molbbbp) classification. For KD experiments, we further include Amazon Computers and Amazon Photo. Our baseline models are GCN, GAT, and GraphSage, evaluated in both homogeneous and heterogeneous pairings. All results are reported as mean accuracy with standard deviation over ten runs (five for graph classification extension).

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 2: Performance of MLP with KD using the target model. Ind is the performance of the MLP without any mutual learning.

Models	Methods	Cora	Citeseer	PubMed	A-Computers	A-photo
	Ind-MLP	70.10 ± 1.12	67.88 ± 0.53	84.25 ± 0.75	77.59 ± 0.64	87.42 ± 0.75
GraphSage-GCN-S	KD without GML	86.01 ± 0.19	75.11 ± 0.08	88.98 ± 0.14	89.03 ± 0.31	94.87 ± 0.09
	KD + GML	86.18 ± 0.24	76.01 ± 0.16	89.45 ± 0.17	**89.48 ± 0.30**	95.02 ± 0.13
	KD + GML-Co	87.49 ± 0.28	75.81 ± 0.30	**89.53 ± 0.20**	89.44 ± 0.27	94.56 ± 0.16
	KD + GML-W	**87.66 ± 0.24**	**76.27 ± 0.10**	88.46 ± 0.22	89.46 ± 0.26	94.08 ± 0.13
	KD + GML-C	86.03 ± 0.23	75.71 ± 0.26	89.24 ± 0.28	89.04 ± 0.20	**95.18 ± 0.15**
GAT-GraphSage-S	KD without GML	86.01 ± 0.19	75.11 ± 0.08	88.98 ± 0.14	89.03 ± 0.31	94.87 ± 0.09
	KD + GML	85.67 ± 0.10	76.55 ± 0.19	89.58 ± 0.27	90.04 ± 0.23	94.95 ± 0.20
	KD + GML-Co	**87.44 ± 0.16**	75.97 ± 0.26	**89.88 ± 0.23**	89.47 ± 0.20	94.43 ± 0.09
	KD + GML-W	85.99 ± 0.32	76.27 ± 0.01	89.36 ± 0.25	88.96 ± 0.20	94.80 ± 0.19
	KD + GML-C	86.33 ± 0.26	**76.57 ± 0.45**	89.42 ± 0.21	**90.10 ± 0.25**	**95.43 ± 0.10**
GCN-GAT-C	KD without GML	88.52 ± 0.51	78.92 ± 0.23	88.45 ± 0.15	77.04 ± 0.36	90.17 ± 0.52
	KD + GML	87.91 ± 0.22	79.00 ± 0.16	**88.55 ± 0.23**	77.54 ± 0.40	90.68 ± 0.41
	KD + GML-Co	88.42 ± 0.23	78.82 ± 0.38	88.29 ± 0.39	76.36 ± 0.64	90.21 ± 0.40
	KD + GML-W	88.32 ± 0.28	78.42 ± 0.30	88.43 ± 0.20	77.36 ± 0.34	**90.72 ± 0.37**
	KD + GML-C	**88.72 ± 0.19**	**79.48 ± 0.22**	88.49 ± 0.41	**77.82 ± 0.37**	90.62 ± 0.29

Results. We evaluate GML's ability to enhance baseline GNNs through collaborative training across diverse architectures and initialization schemes, including GraphSage-GCN, GAT-GraphSage, and GCN-GAT. For GraphSage-GCN and GAT-GraphSage, GraphSage serves as the target; for GCN-GAT, GCN is the target. As shown in Table 1, GML consistently improves accuracy across datasets. For example, on Cora, GAT-GraphSage (target: GraphSage) rises from 86.58% to 88.55%; on PROTEINS, accuracy increases from 69.64% to 70.36% with a confidence penalty; and on Ogbg-molbbp, GraphSage-GCN improves from 84.20% to 85.38% using adaptive logit weighting. These results confirm that GML with targeted enhancements strengthens shallow GNNs.

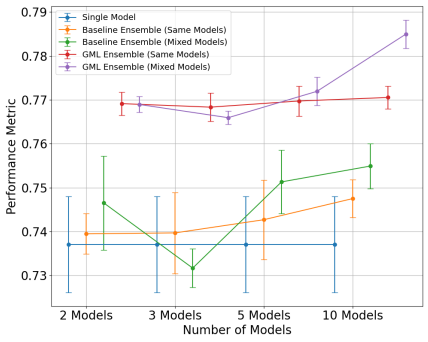


Fig. 3: Comparison of GML ensemble method with deep ensembles.

We further assess whether GML-trained teachers can transfer their improvements to MLPs via KD for efficient inference. Using GraphSage and GCN as teachers from the above pairs, we observe substantial gains (Table 2); e.g., the GraphSage teacher from GraphSage-GCN boosts MLP accuracy on Cora from 70.10% to 87.66% with adaptive logit weighting, indicating that KD captures transferable, generalizable knowledge from GML.

Finally, we evaluate GML's scalability by aggregating predictions from complementary models. In addition, we compare GML's performance with ensemble learning. As shown in Figure 3, performance improves with larger and more diverse cohorts, surpassing standard ensemble methods. This demonstrates GML's effectiveness in leveraging model diversity for enhanced graph learning.

4.1 Extension to Graph-Aware Adaptive Logit Weighting

Our GML framework offers (i) expressiveness through knowledge transfer among GNNs suited for graph data and (ii) adaptive learning via adaptive logit weighting

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 3: Comparison of Accuracy Before and After Graph-Aware Optimization for Different Datasets

Dataset	Best Accuracy Before (%)	Accuracy After (%)
Cora	88.69 ± 0.38	$\mathbf{89.48 \pm 0.42}$
Citeseer	77.80 ± 0.51	$\mathbf{78.78 \pm 0.32}$
PubMed	90.19 ± 0.30	$\mathbf{90.31 \pm 0.24}$

and entropy enhancement, which dynamically guide training—especially for shallow models. To leverage relational structure, we extend GML by incorporating message passing into entropy computation. We refine the negative entropy $H(p_j)$ through graph convolution, $H_g(p_j) = \text{GCNConv}(H(p_j), \text{edge\_index})$, where $H_g(p_j) \in \mathbb{R}^{N\times 1}$ integrates neighborhood information. The class-importance score becomes $\sigma_j^c = H_g(p_j)^\top \chi_j \phi_j$, yielding graph-aware adaptive weights. Experiments with the GCN–SAGE–S setup on Cora (Table 3) show additional improvements, confirming that incorporating message passing strengthens mutual learning.

5 Conclusion and Future Work

In this paper, we presented GML, a framework that enhances GNNs through deep mutual learning. GML integrates adaptive logit weighting and a confidence-penalty to improve knowledge exchange among peers and encourage high entropy during training. We also showed that the knowledge produced during training can be transferred to an MLP for downstream tasks. Experiments on different datasets demonstrate that GML strengthens shallow GNNs through distillation.

References

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

[2] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

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

[4] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[5] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328, 2018.

[6] Z. Guo, C. Zhang, Y. Fan, Y. Tian, C. Zhang, and N. V. Chawla, "Boosting graph neural networks via adaptive knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 7793–7801, 2023.

[7] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International conference on machine learning*, pp. 3519–3529, PMLR, 2019.

[8] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.