DistillHGNN: A Knowledge Distillation Approach for High-Speed Hypergraph Neural Networks

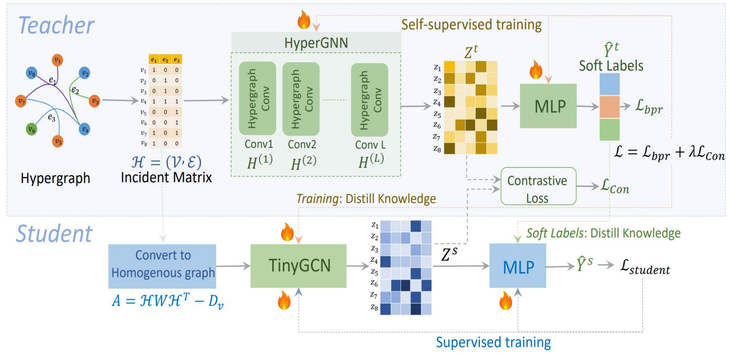
Saman Forouzandeh, Parham Moradi and Mahdi Jalili
{saman.forouzandeh, parham.moradi, mahdi.jalili}@rmit.edu.au
RMIT University, Melbourne, Australia

ICLR 2025

RMIT UNIVERSITY

Contributions

- DistillHGNN transfers soft labels and structural knowledge from HGNN to TinyGCN, resulting in richer and more effective knowledge distillation than methods like LightHGNN.
- DistillHGNN utilises a contrastive learning strategy to maximise the similarity between embeddings generated by the HGNN and TinyGCN, effectively transferring high-order structural knowledge to the student model.
- TinyGCN is streamlined to a single layer without activation functions, reducing computational complexity while effectively capturing the high-order relationships of the teacher HGNN.



- Proposed model consists of a teacher (HGNN) and a student (TinyGCN). The HGNN represents hypergraphs and generates soft labels using high-order relationships.
- TinyGCN captures direct neighborhoods relations in graphs. The soft labels guide the student model via a KL divergence term and are also used to generate labels for datasets with insufficient ground truth annotations.
- Both models are trained using a combination of supervised loss and contrastive loss to align their embeddings, enabling efficient knowledge distillation from teacher to student.Rr

Proposed Model

Teacher

Hypergraph Neural Network (HGNN)

$$H^{(l+1)} = \sigma\left(D_v^{-1/2}\mathcal{H}WD_e^{-1}\mathcal{H}^TD_v^{-1/2}H^{(l)}\Theta^{(l)}\right)$$

$$L_{\text{bpr}} = \frac{1}{|\mathcal{V}^L|}\sum\left(Y_v - Y_v^t\right)^2$$

$$L_{\text{con}} = -\frac{1}{|V|}\sum_{v\in V}\log\left(\frac{\exp\left(Z_v^s \cdot Z_v^t/\tau\right)}{\sum_{v'\in V}\exp\left(Z_{v'}^s \cdot Z_{v'}^t/\tau\right)}\right)$$

$$L_{\text{teacher}} = L_{\text{bpr}} + \gamma L_{\text{con}}$$

The teacher is trained using the contrastive learning strategy combined with supervised learning on limited labelled data.

Student

Graph Convolutional Networks (Single Layer)

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)})$$

$$\hat{Y}_v^s = \text{MLP}^s(Z_v^s),$$

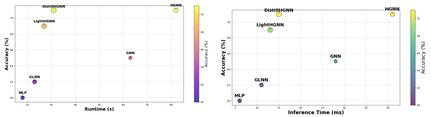
$$L_{\text{student}} = \frac{1}{|\mathcal{V}^L|}\sum_{v\in\mathcal{V}^L}(\hat{Y}_v - Y_v)^2 + \lambda\frac{1}{|V|}\sum_{v\in V}\text{KL}(\hat{Y}_v||Y_v^t)$$

The student model is trained using limited labelled data and soft labels received from the teacher.

Algorithm 1 DistillHGNN: HGNN with Knowledge Distillation

Input: Hypergraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, features X, incidence matrix $\mathcal{H}$, labelled data $D_L = \{V_L, Y_L\}$, number of epochs *epochs*, parameters τ, γ, λ
Output: Student model parameters

Initialise: Model parameters for HGNN ($\Theta^{(l)}$), MLP^t (Θ^t), TinyGCN (W^s), MLP^s (Θ^s), and $H^{(0)}$
Step 1: Compute Laplacian and Adjusted Adjacency
$\mathcal{L} \leftarrow D_v^{-1/2}\mathcal{H}WD_e^{-1}\mathcal{H}^TD_v^{-1/2}$
$\tilde{A} \leftarrow A^s + I$
Step 2: Pre-train the Teacher Model
for epoch $= 1$ to *epochs* **do**
  Compute HGNN embeddings: $Z^t \leftarrow H^{(L)}$, where
  $$H^{(l+1)} = \sigma(\mathcal{L}H^{(l)}\Theta^{(l)})$$
  Generate teacher predictions: $Y_v^t \leftarrow \text{MLP}^t(Z^t)$
  Compute TinyGCN embeddings: $Z^s \leftarrow \tilde{A}^sXW^s$
  Compute teacher loss:
  $$L_{\text{teacher}} \leftarrow \frac{1}{|V_L|}\sum_{v\in V_L}(Y_v - Y_v^t)^2 - \gamma\frac{1}{|V|}\sum_{v\in V}\log\frac{\exp(Z_v^s \cdot Z_v^t/\tau)}{\sum_{v'}\exp(Z_{v'}^s \cdot Z_{v'}^t/\tau)}$$
  Update teacher parameters Θ^t
end for
Step 3: Train the Student Model
for epoch $= 1$ to E **do**
  Generate soft labels (Y^t) for all nodes using frosen teacher.
  Compute student outputs:
  $$Z^s \leftarrow \tilde{A}^sXW^s, \quad \hat{Y}_v^s \leftarrow \text{MLP}^s(Z^s)$$
  Compute student loss:
  $$L_{\text{student}} \leftarrow \frac{1}{|V_L|}\sum_{v\in V_L}(\hat{Y}_v^s - Y_v)^2 + \lambda\frac{1}{|V|}\sum_{v\in V}\text{KL}(\hat{Y}_v^s||Y_v^t)$$
  Update student parameters $\{W^s, \Theta^s\}$
end for
Return: Student model parameters

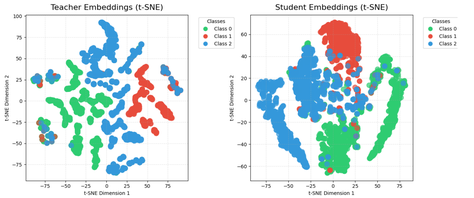


Comparison of model accuracy and inference time across different methods on the IMDB-AW and DBLP datasets. The figure illustrates the trade-off between predictive performance and computational efficiency, highlighting the balance achieved by DistillHGNN.

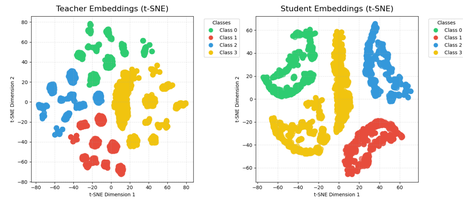
Table 5: Comparison of Inference Time (ms) between DistillHGNN and HGNN for IMDB and DBLP Datasets at Different Node Levels.

Dataset	Hyperedge Count	HGNN	DistillHGNN	Improvement (%)
IMDB	10,000	8.12	0.67	12.12
	20,000	37.35	1.13	33.05
	30,000	84.70	1.68	50.42
	40,635	175.56	2.23	78.70
DBLP	10,000	9.47	0.72	13.15
	20,000	42.15	0.98	43.01
	30,000	75.33	1.51	49.88
	43,128	168.84	2.06	81.97

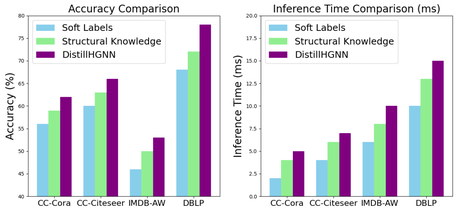
Experiments



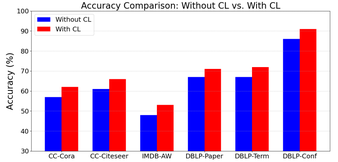
t-SNE visualisations comparing teacher and student embeddings for IMDB-AW dataset shows scattered and overlapping class distributions in the teacher embeddings, which are effectively refined into cohesive and distinct clusters in the student embeddings, indicating successful knowledge transfer.



t-SNE for DBLP dataset reveals fragmented and abnormal class clusters in the teacher embeddings due to the hypergraph model's high-dimensional feature complexities and high-order relationships. In contrast, the student embeddings display well-organised and continuous class regions, demonstrating the effectiveness of knowledge distillation in simplifying and structuring complex representations into interpretable embeddings



The proposed model is evaluated using three different knowledge transfer methods: (1) soft labels alone, (2) structural knowledge alone, and (3) a combination of both (DistillHGNN). The evaluation includes calculating both accuracy and inference time for each method



The proposed model is evaluated based on the absence of contrastive learning (lack of CL) and with contrastive learning (DistillHGNN), focusing on accuracy across four datasets.

Access to the Paper

Saman Forouzandeh

Parham Moradi

Mahdi Jalili