A Post-Training Framework for Improving Heterogeneous Graph Neural Networks

Authors: Cheng Yang, Xumeng Gong, Chuan Shi, Philip S. Yu

Presented by: Alex Zheng

Cheng Yang, Xumeng Gong, Chuan Shi, and Philip S. Yu. 2023. A Post-Training Framework for Improving Heterogeneous Graph Neural Networks. In Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023.

ILLINOIS

Outline

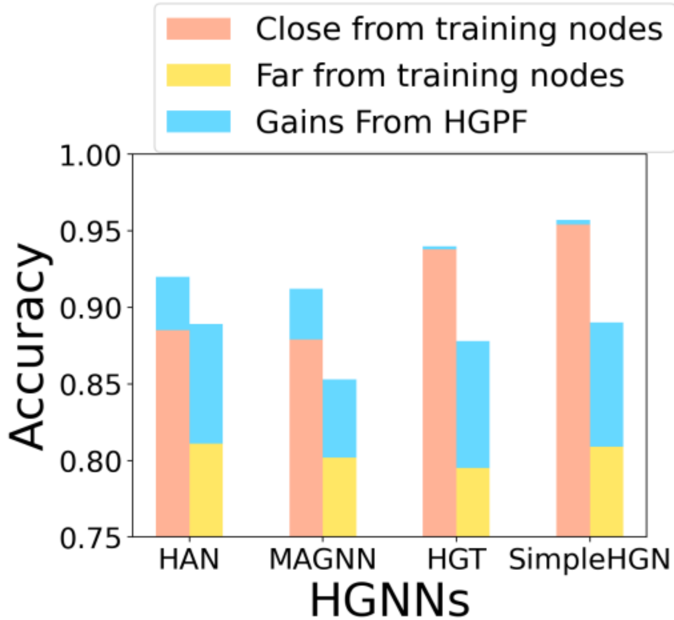
- Introduction
- Background
- Methodology
- Experiments
- Conclusion

Motivation

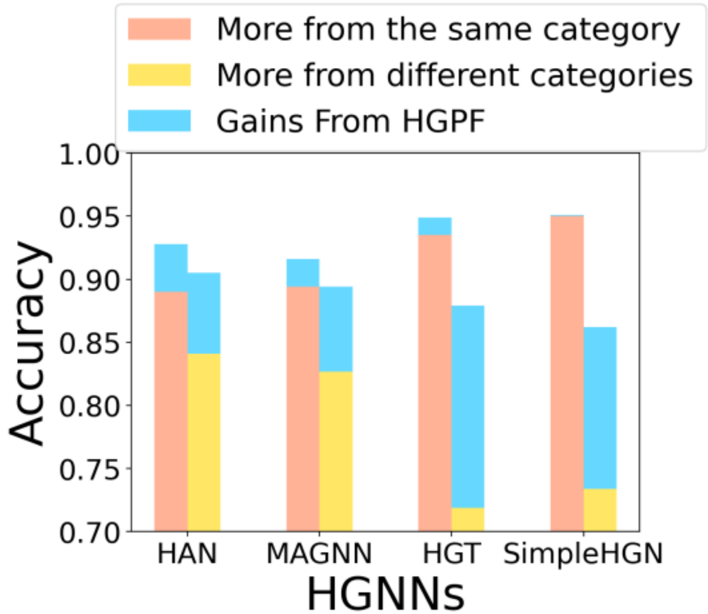
- Current heterogeneous graph neural networks (HGNN) suffer when predicting a test node's label when its receptive field ...
  - Has few training nodes of the **same** category (Sparsity)
  - Has multiple training nodes from **different** categories (Unrelatedness)

- Naïve Approach: Stack more layers to enlarge the receptive field
  - More noise
  - Over-smoothing

Performance Gaps



(a) Close (red) v.s. Far (yellow)

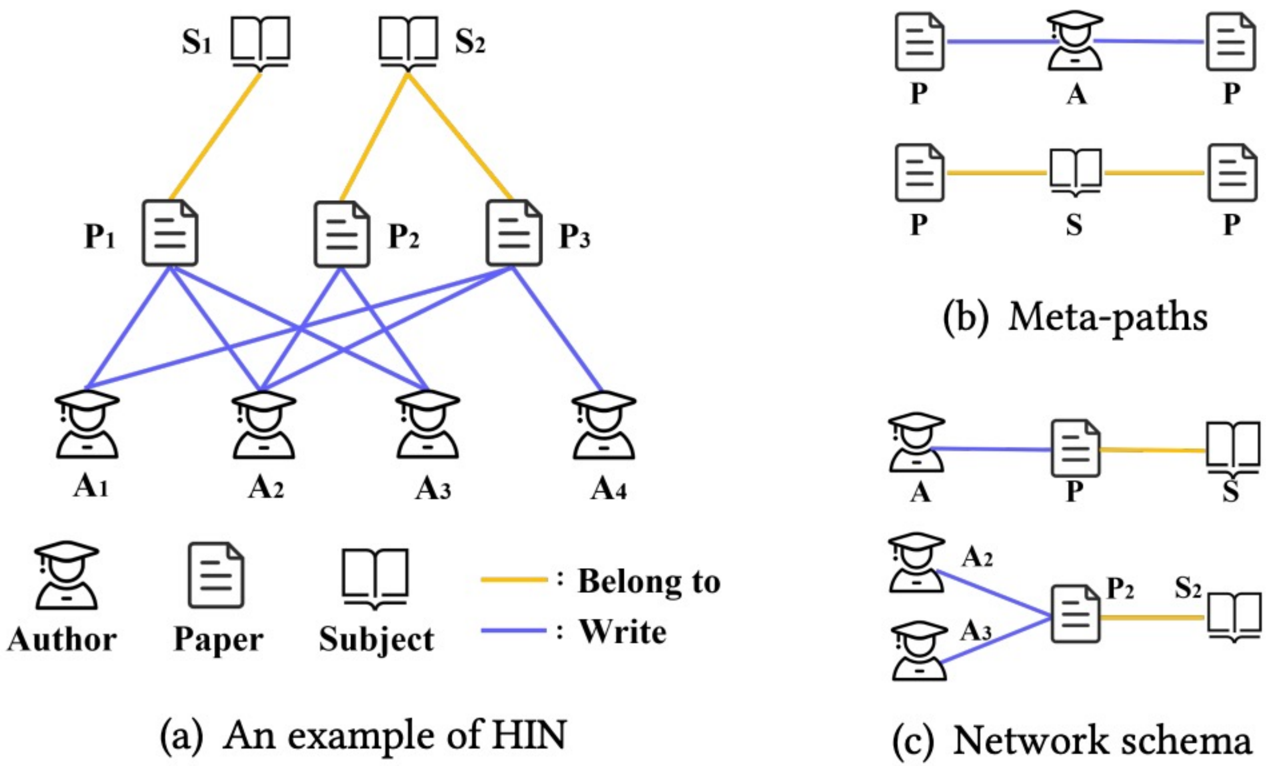


(b) Same (red) v.s. Different (yellow)

Outline

- Introduction
- Background
- Methodology
- Experiments
- Conclusion

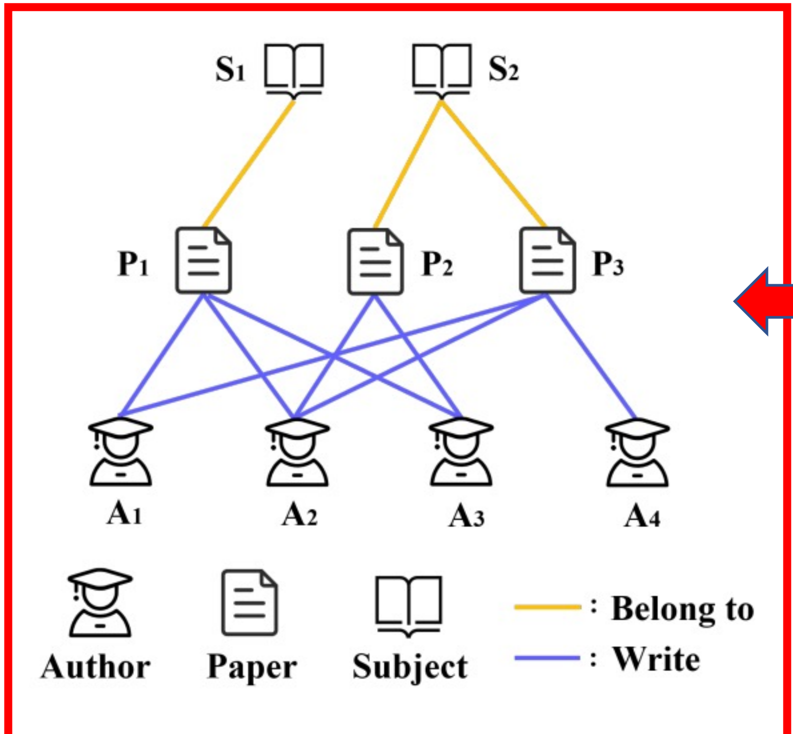
Background (HIN)



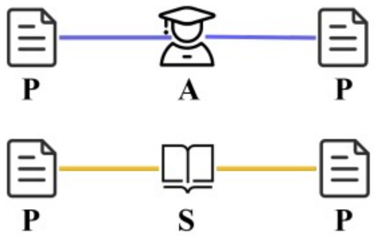
Heterogeneous Information Network

- **Definition:** a Heterogeneous Information Network (HIN) is a graph with the following properties
$$G = \{V, E, T, R, \phi, \psi\}$$
  - V: Node set
  - E: Edge set
  - T: Node types
  - R: Edge types
  - $\phi: V \to T$: Mapping of nodes to node types
  - $\psi: E \to R$: Mapping of edges to edge types
  - $|T| + |R| > 2$

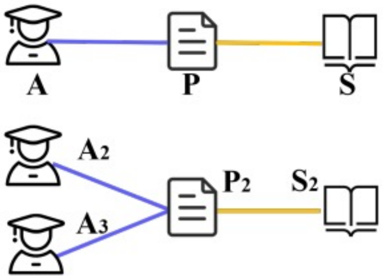
Heterogeneous Information Network



(a) An example of HIN



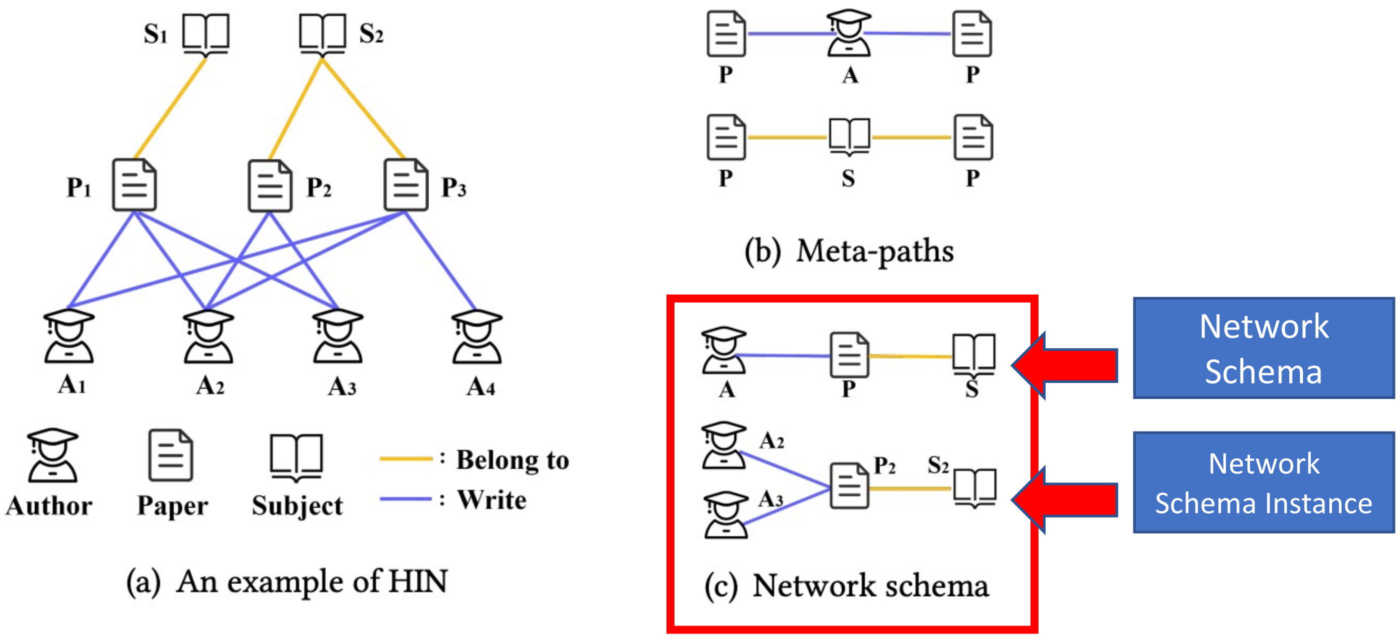
(b) Meta-paths



(c) Network schema

Network Schema

- **Definition:** a Network Schema $S_G = (T, R)$ is a directed graph defined on T and R which is a blueprint for the network structure



(a) An example of HIN

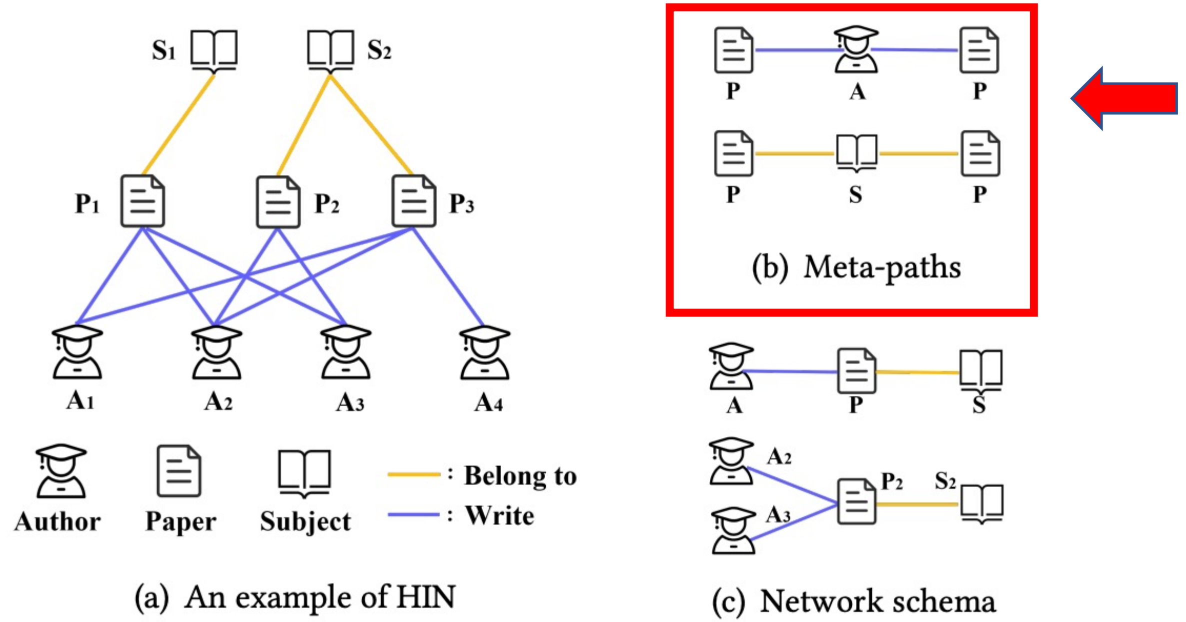
(b) Meta-paths

(c) Network schema

Meta-Path

- **Definition:** a Meta-Path is a path with the form

$$T_1 \xrightarrow{R_1} T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_l} T_1$$



Outline

- Introduction
- Background
- Methodology
- Experiments
- Conclusion

HGNN Agnostic

$$\min_{\Theta} \sum_{v \in \mathcal{V}_L} \mathcal{L}(f_{\Theta}(v), y_v),$$

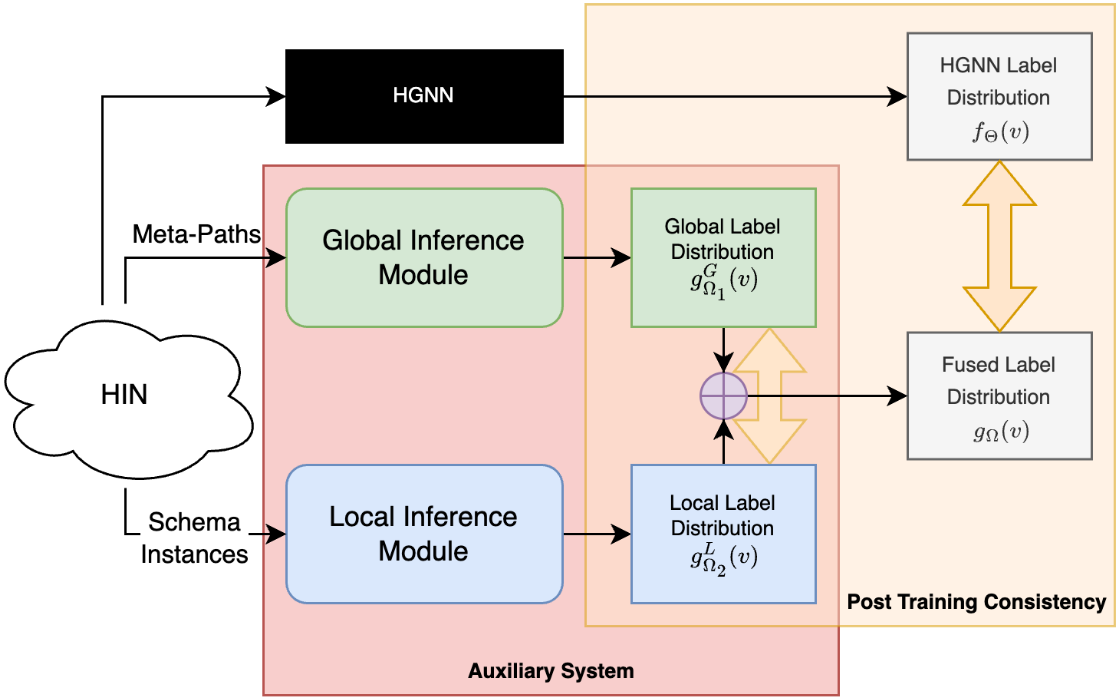
Treat HGNNs as a black box!

HGPF Framework

Heterogeneous Graph Post-training Framework (HGPF)

Two Components:

- Auxiliary System
- Post-Training Algorithm (Consistency)



Auxiliary System

- Predict node labels based on both global and local inference modules
  - **Global Inference Module:**
    - Diffuse known node labels to distant nodes by multichannel label propagation for each meta-path
  - **Local Inference Module:**
    - Predict node labels with node features based on every node's network schema instance

Global Inference Module

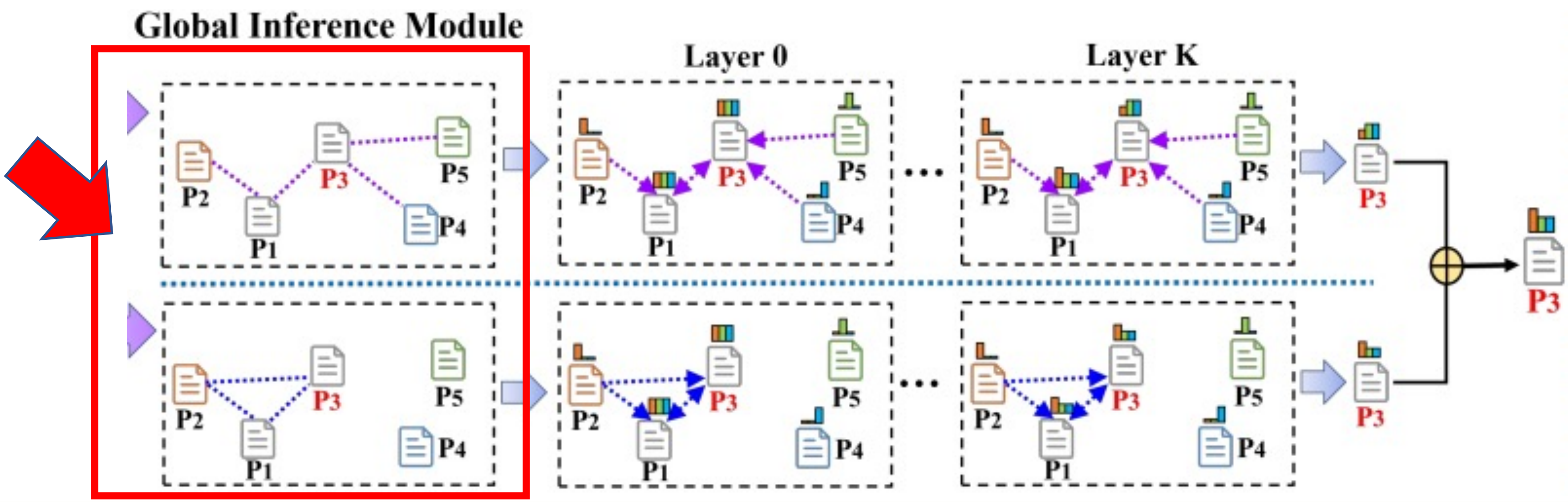
Multi-Channel Label Propagation (MCLP)

- **Key Assumption:** Nodes linked by meta-paths tend to have similar labels
- Each channel corresponds to one of the dataset's pre-defined meta-paths
- Propagate labels, not features
- Can be stacked for more layers e.g., 8-10 (compared to 1-2 in most HGNN frameworks)

Global Inference Module - Algorithm

1. Initialize label prediction vectors to one-hot or uniformly distributed vectors

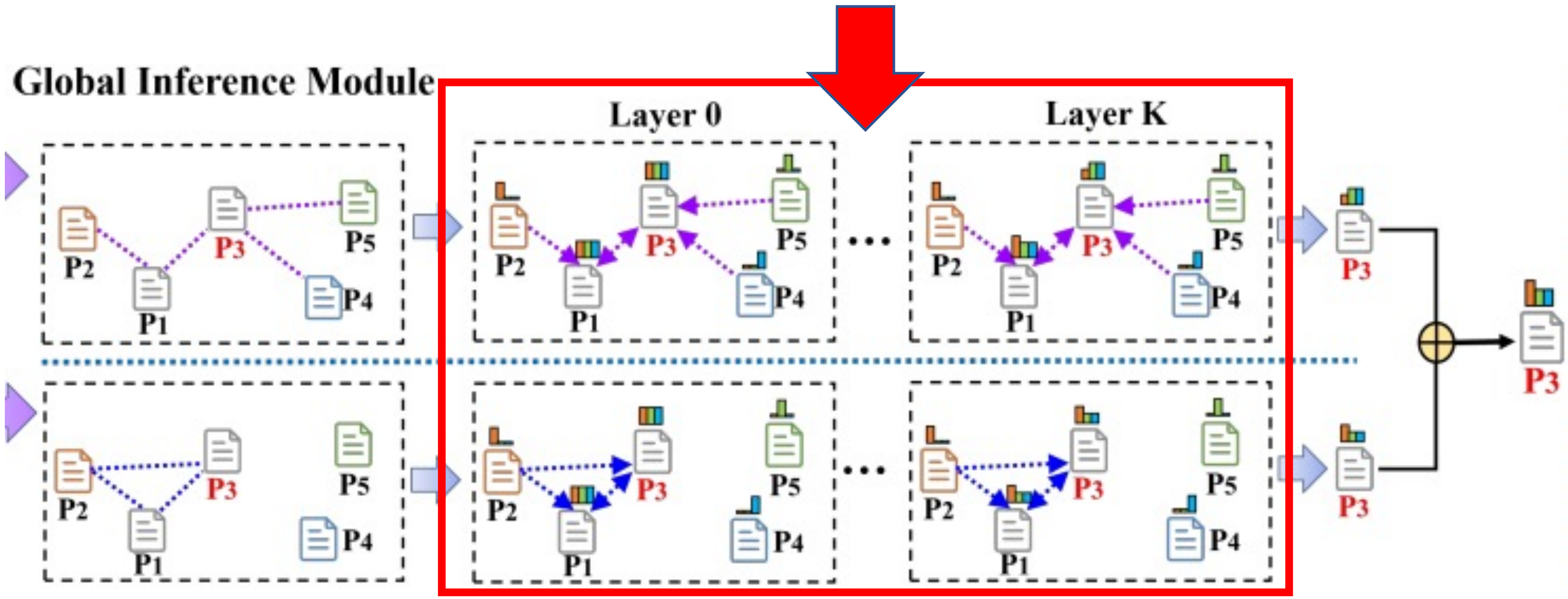
$$l_P^0(v) = \begin{cases} (0, \dots 1, \dots 0) \in \mathbb{R}^{|\mathcal{Y}|}, & \forall v \in \mathcal{V}_L \\ (\frac{1}{|\mathcal{Y}|}, \dots \frac{1}{|\mathcal{Y}|}, \dots \frac{1}{|\mathcal{Y}|}) \in \mathbb{R}^{|\mathcal{Y}|}, & \forall v \in \mathcal{V}_U \end{cases},$$



Global Inference Module - Algorithm

2. Parametrize propagation weights $w_{uv}^{P} \in [0,1]$ via learnable parameters s_{uv}^{P} and propagate labels

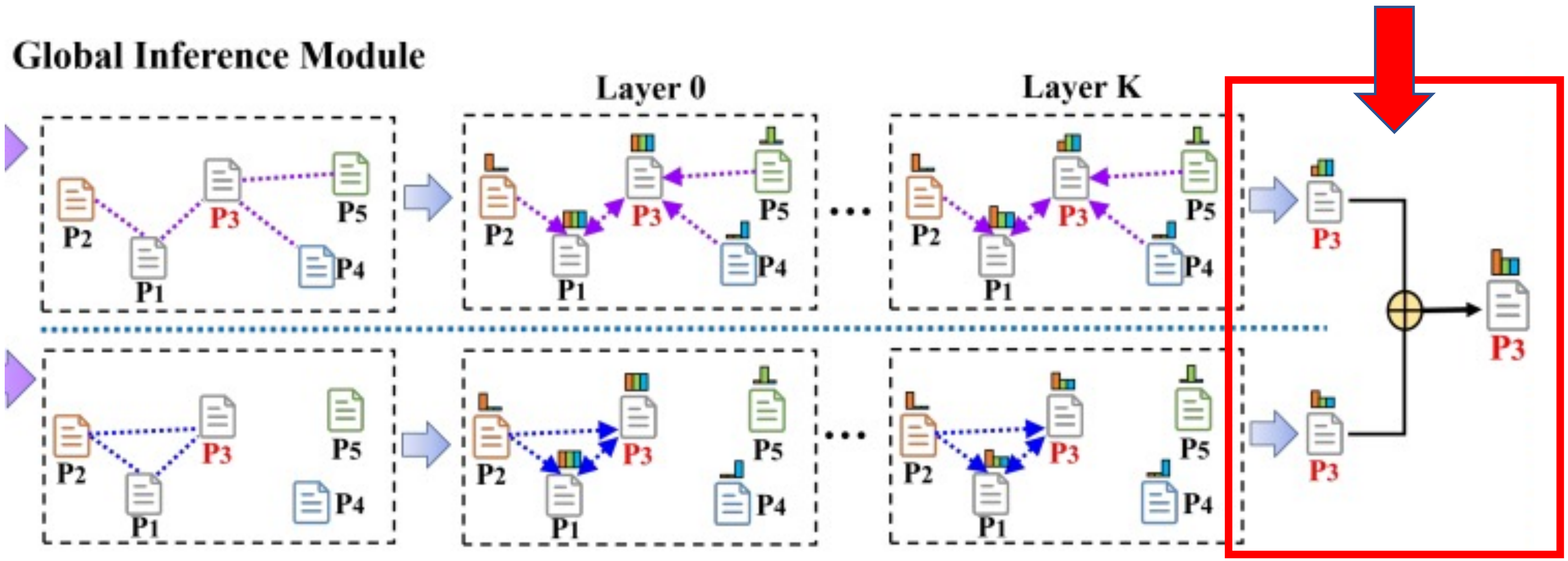
$$w_{uv}^{P} = \frac{\exp(s_{uv}^{P})}{\sum_{u' \in \mathcal{N}_v^P} \exp(s_{u'v}^{P})}, \qquad l_P^{k+1}(v) = \sum_{u \in \mathcal{N}_v^P} w_{uv}^{P} l_P^{k}(u),$$



Global Inference Module - Algorithm

3. Aggregate each channel's label distribution weighted by a learnable parameter

$$g_{\Omega_1}^{G}(v) = \sum_{P \in \mathcal{P}} \alpha_v^P l_P^K(v),$$



Local Inference Module

Goal: Predict node labels based on every node's network schema instance

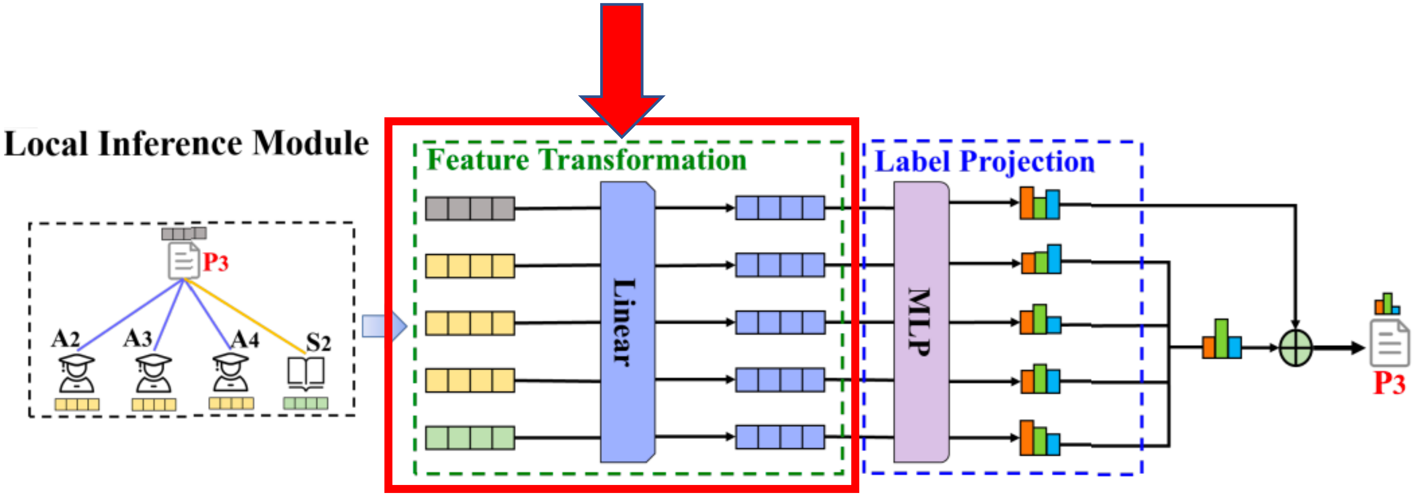
- **Key Assumption:** all nodes with different types in a network schema instance tend to be similar
  - Network Schema Proximity [1]

[1] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye. Network schema preserving heterogeneous information network embedding. In Proceedings of IJCAI, 2020.

Local Inference Module - Algorithm

1. Project the features h_u of different types of nodes into the same space

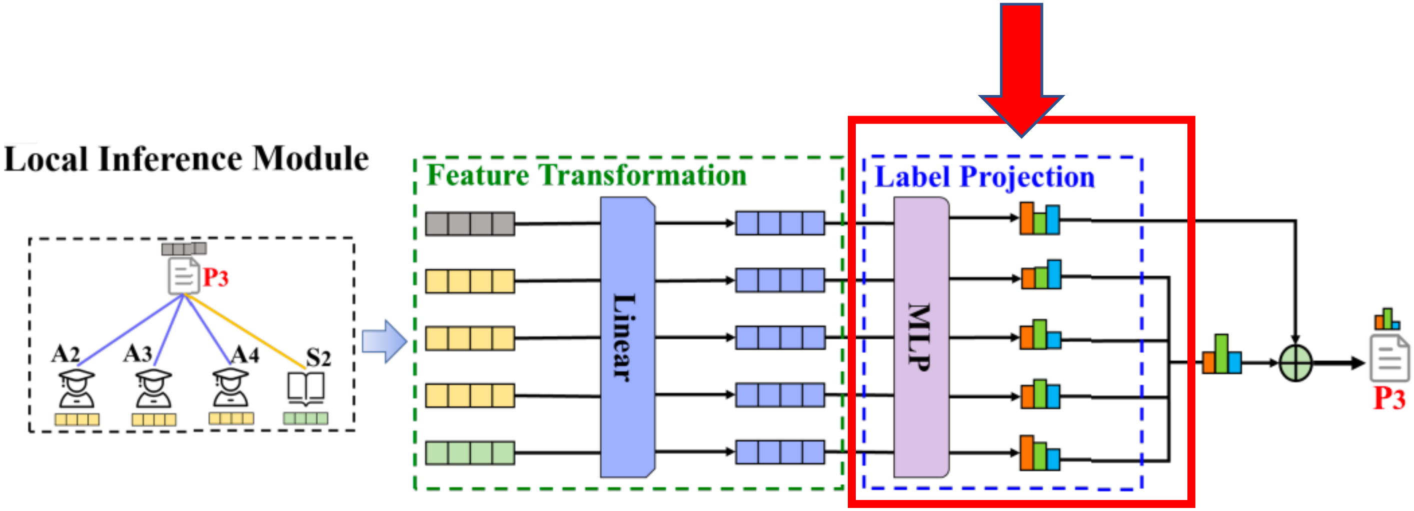
$$h_u = W_{\phi(u)} \cdot x_u, \forall u \in V$$



Local Inference Module - Algorithm

2. Project the features to a label distribution p_u via a multi-layer perceptron (MLP)

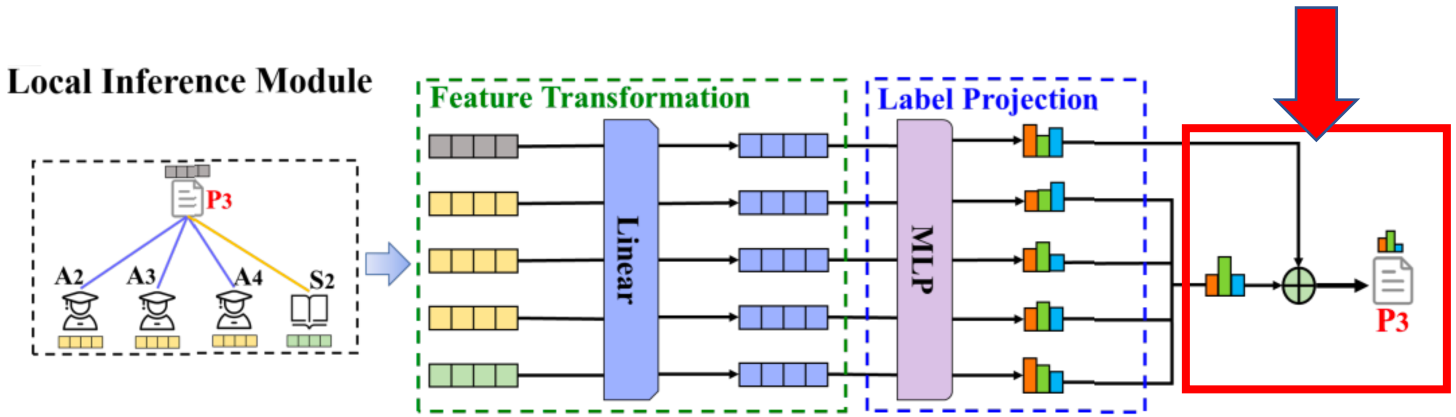
$$p_u = softmax(MLP(h_u))$$



Local Inference Module - Algorithm

3. Using learnable weights, fuse the label distribution of u with the label distributions of the nodes in the same network schema instance

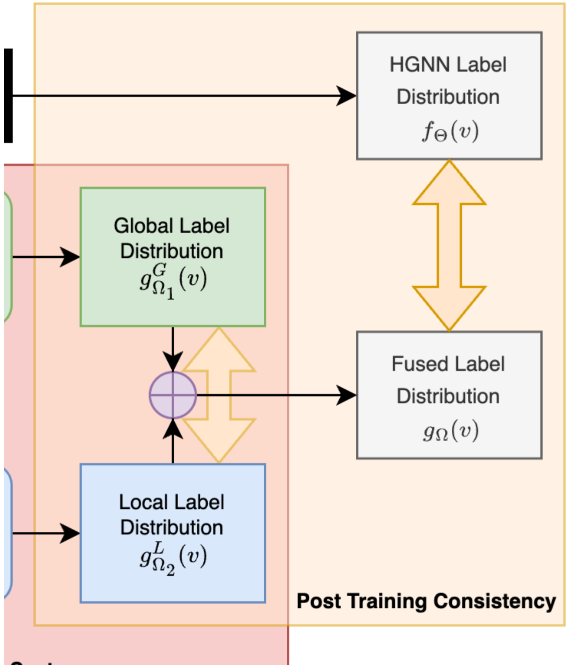
$$g_{\Omega_2}^{L} = \beta_v p_v + (1 - \beta_v) \frac{\sum_{u \in N_v} p_u}{|N_v|}$$



Post-Training Algorithm (Consistency)

Force consistency in prediction gaps between

- Auxiliary System $g_\Omega \leftrightarrow$ HGNN f_Θ
- Global Module $g^G_{\Omega_1} \leftrightarrow$ Local Module $g^L_{\Omega_2}$



Post-Training Algorithm (Consistency)

Alternately update the two systems

1. Update parameters $\Omega = \{\Omega_1, \Omega_2\}$ of auxiliary system
2. Update parameters Θ of the HGNN

$$\min_{\Omega} \sum_{v \in \mathcal{V}_U} \mathrm{dist}(f_\Theta(v), g_\Omega(v)) + \lambda \mathrm{dist}(g^G_{\Omega_1}(v), g^L_{\Omega_2}(v)),$$

$$\min_{\Theta} \sum_{v \in \mathcal{V}_U} \mathrm{dist}(f_\Theta(v), g_\Omega(v)) + \sum_{v \in \mathcal{V}_L} \mathcal{L}(f_\Theta(v), y_v),$$

Making Predictions

Two ways:

- Make predictions through the learned auxiliary system
- Make predictions based on the fine-tuned HGNN model

Empirically, the learned auxiliary system outperforms the fine-tuned HGNN

Outline

- Introduction
- Background
- Methodology
- Experiments
- Conclusion

Datasets

- ACM
  - Node Types: Author (A), Paper (P), Subject (S)
  - Meta-Paths: PAP, PSP
- DBLP
  - Node Types: Author (A), Paper (P), Term (T), Venue (V)
  - Meta-Paths: APA, APVPA, APTPA
- IMDB
  - Node Types: Movie (M), Director (D) , Actor (A)
  - Meta-Paths: MAM, MDM

HGNN Backbones

- HAN
- HGT
- Simple-HGN
- MAGNN

Recall: HGPF is meant to be a black box algorithm for HINs

Node Classification (HGPF)

Note: For the auxiliary system, HGPF-self uses the same HGNN type (without sharing parameters). Essentially two HGNNs

Table 1: Classification performance with HAN [21] and HGT [9].

Models		HAN						HGT					
		Pretrain		$HGPF_{self}$		HGPF		Pretrain		$HGPF_{self}$		HGPF	
# Labeled Nodes		20	50	20	50	20	50	20	50	20	50	20	50
ACM	Micro-F1	0.8826	0.8838	0.8949	0.9091	**0.9163**	**0.9271**	0.8693	0.8701	0.8835	0.8852	**0.9173**	**0.9177**
	Macro-F1	0.8785	0.8864	0.8901	0.9054	**0.9165**	**0.9280**	0.8679	0.8699	0.8827	0.8807	**0.9173**	**0.9174**
DBLP	Micro-F1	0.9092	0.9217	0.9251	0.9300	**0.9280**	**0.9349**	0.8941	0.9256	0.9011	0.9317	**0.9084**	**0.9342**
	Macro-F1	0.9038	0.9165	0.9220	0.9242	**0.9258**	**0.9287**	0.8871	0.9229	0.8925	0.9239	**0.8995**	**0.9275**
IMDB	Micro-F1	0.4581	0.4809	0.4629	0.5060	**0.4879**	**0.5149**	0.4600	0.5067	0.4672	0.5117	**0.4724**	**0.5286**
	Macro-F1	0.4346	0.4817	0.4574	0.5059	**0.4792**	**0.5189**	0.4540	0.5123	**0.4623**	0.5139	0.4611	**0.5328**

Table 2: Classification performance with Simple-HGN [14] and MAGNN [3].

Models		Simple-HGN						MAGNN					
		Pretrain		$HGPF_{self}$		HGPF		Pretrain		$HGPF_{self}$		HGPF	
# Labeled Nodes		20	50	20	50	20	50	20	50	20	50	20	50
ACM	Micro-F1	0.8816	0.8865	0.8945	0.8994	**0.9179**	**0.9216**	0.8776	0.8831	0.8917	0.9022	**0.9157**	**0.9179**
	Macro-F1	0.8815	0.8881	0.8895	0.8943	**0.9179**	**0.9210**	0.8715	0.8824	0.8923	0.9014	**0.9112**	**0.9173**
DBLP	Micro-F1	0.9108	0.9253	0.9245	0.9315	**0.9279**	**0.9366**	0.9121	0.9223	0.9271	0.9322	**0.9291**	**0.9359**
	Macro-F1	0.9026	0.9249	0.9152	0.9286	**0.9177**	**0.9316**	0.9056	0.9228	0.9234	0.9269	**0.9252**	**0.9295**
IMDB	Micro-F1	0.4698	0.5109	0.4798	0.5318	**0.4925**	**0.5412**	0.4518	0.5090	0.4877	0.5173	**0.4962**	**0.5292**
	Macro-F1	0.4562	0.5141	0.4522	0.5296	**0.4874**	**0.5396**	0.4515	0.5119	0.4880	0.5204	**0.4921**	**0.5256**

Node Classification (HGPF)

Table 1: Classification performance with HAN [21] and HGT [9].

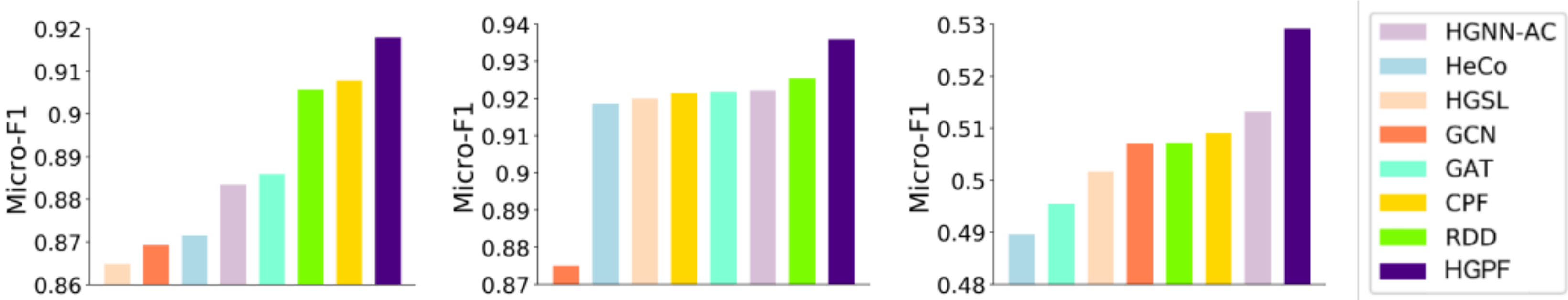
Models		HAN						HGT					
		Pretrain		$HGPF_{self}$		HGPF		Pretrain		$HGPF_{self}$		HGPF	
# Labeled Nodes		20	50	20	50	20	50	20	50	20	50	20	50
ACM	Micro-F1	0.8826	0.8838	0.8949	0.9091	**0.9163**	**0.9271**	0.8693	0.8701	0.8835	0.8852	**0.9173**	**0.9177**
	Macro-F1	0.8785	0.8864	0.8901	0.9054	**0.9165**	**0.9280**	0.8679	0.8699	0.8827	0.8807	**0.9173**	**0.9174**
DBLP	Micro-F1	0.9092	0.9217	0.9251	0.9300	**0.9280**	**0.9349**	0.8941	0.9256	0.9011	0.9317	**0.9084**	**0.9342**
	Macro-F1	0.9038	0.9165	0.9220	0.9242	**0.9258**	**0.9287**	0.8871	0.9229	0.8925	0.9239	**0.8995**	**0.9275**
IMDB	Micro-F1	0.4581	0.4809	0.4629	0.5060	**0.4879**	**0.5149**	0.4600	0.5067	0.4672	0.5117	**0.4724**	**0.5286**
	Macro-F1	0.4346	0.4817	0.4574	0.5059	**0.4792**	**0.5189**	0.4540	0.5123	**0.4623**	0.5139	0.4611	**0.5328**

Table 2: Classification performance with Simple-HGN [14] and MAGNN [3].

Models		Simple-HGN						MAGNN					
		Pretrain		$HGPF_{self}$		HGPF		Pretrain		$HGPF_{self}$		HGPF	
# Labeled Nodes		20	50	20	50	20	50	20	50	20	50	20	50
ACM	Micro-F1	0.8816	0.8865	0.8945	0.8994	**0.9179**	**0.9216**	0.8776	0.8831	0.8917	0.9022	**0.9157**	**0.9179**
	Macro-F1	0.8815	0.8881	0.8895	0.8943	**0.9179**	**0.9210**	0.8715	0.8824	0.8923	0.9014	**0.9112**	**0.9173**
DBLP	Micro-F1	0.9108	0.9253	0.9245	0.9315	**0.9279**	**0.9366**	0.9121	0.9223	0.9271	0.9322	**0.9291**	**0.9359**
	Macro-F1	0.9026	0.9249	0.9152	0.9286	**0.9177**	**0.9316**	0.9056	0.9228	0.9234	0.9269	**0.9252**	**0.9295**
IMDB	Micro-F1	0.4698	0.5109	0.4798	0.5318	**0.4925**	**0.5412**	0.4518	0.5090	0.4877	0.5173	**0.4962**	**0.5292**
	Macro-F1	0.4562	0.5141	0.4522	0.5296	**0.4874**	**0.5396**	0.4515	0.5119	0.4880	0.5204	**0.4921**	**0.5256**

Node Classification (SOTA GNNs)

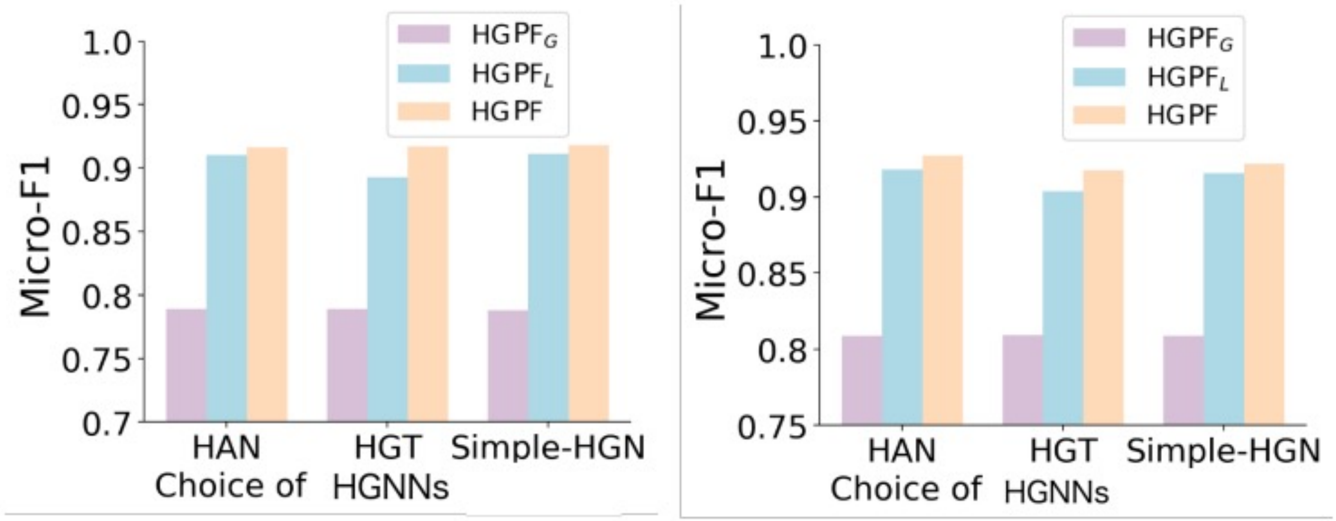
- Use MAGNN as the backbone for GNN and HGPF when needed
- Large performance gain on all datasets



(a) ACM (b) DBLP (c) IMDB

Ablation Study

Experiments demonstrate the importance of the global module, as well as the local module



(a) 20 labeled nodes per class. (b) 50 labeled nodes per class.

Outline

- Introduction
- Background
- Methodology
- Experiments
- Conclusion

Conclusion

- **Existing Problem:** Poor performance on test nodes with following properties:
  - Far from training nodes of same label
  - Training nodes in receptive field are of different label
- **Solution:** Enhance semi-supervised training of HGNNs
  - Global and Local Inference Modules
  - Post-Training Consistency Scheme

Strengths/Weaknesses

- **Strengths**
  - Time/Space complexity linear in scale to HIN
  - HGNN agnostic
  - Strong improvement in performance
- **Weaknesses/Future Directions**
  - Assumptions
    - All nodes of different types in network schema instance = similar
    - Nodes linked by a meta-path instance -> similar labels
  - Imbalanced Classes
    - Training/validation sets are balanced
  - Semi-Supervised
    - Possible to extend to unsupervised setting?
  - Other graph tasks
    - Link Prediction, Node Clustering
  - Other graph types
    - Dynamic, Heterophily

Thank you!

Cheng Yang, Xumeng Gong, Chuan Shi, and Philip S. Yu. 2023. A Post-Training Framework for Improving Heterogeneous Graph Neural Networks. In Proceedings of the ACM Web Conference 2023 (WWW ’23), May 1–5, 2023.