Graph Learning for Bipartite Networks

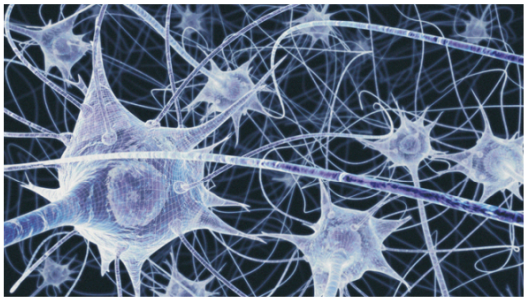
Tian Xie

Collaboration with Chaoyang He, Xiang Ren, Cyrus Shahabi,
and C.-C. Jay Kuo

Real-world Networks

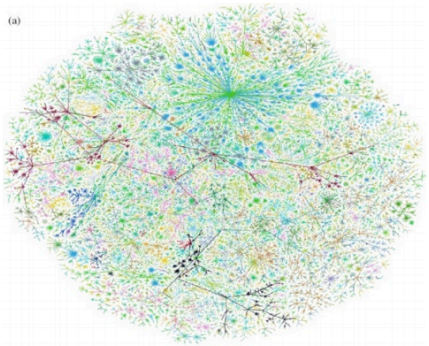
Academic Graph

Office/Social Graph



Biological Neural Networks

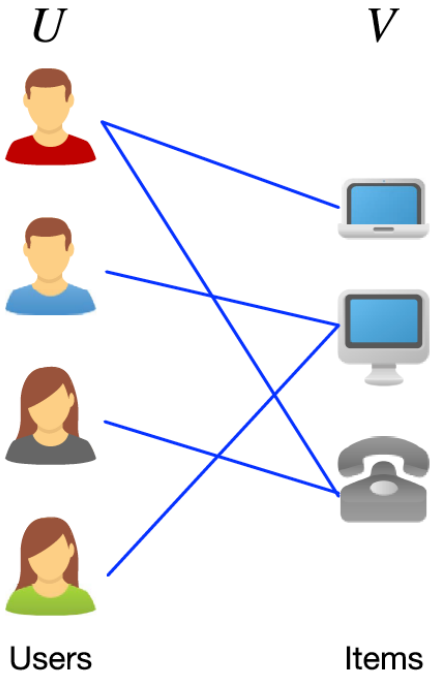
Knowledge Graph



Internet

Transportation

Bipartite Graphs

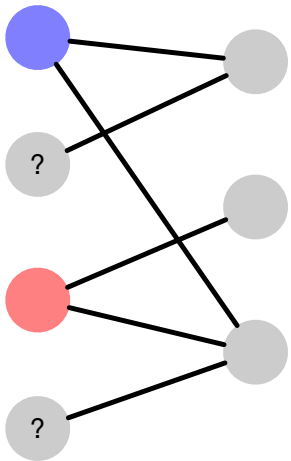


Examples:

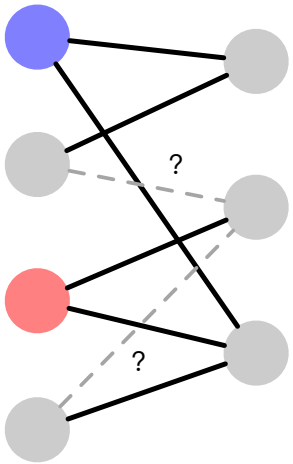
- E-commerce: Users – Items;
- Online ratings: Users – Movies;
- Academic: Authors – Papers;
- Search engines: Queries – Answers;
- …

Machine Learning Tasks on Graphs

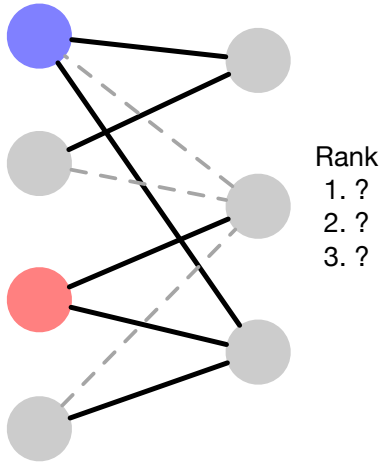
What problems are we trying to solve?



Node classification



Link prediction



Recommendation

More: community detection, anomaly detection, network similarity…

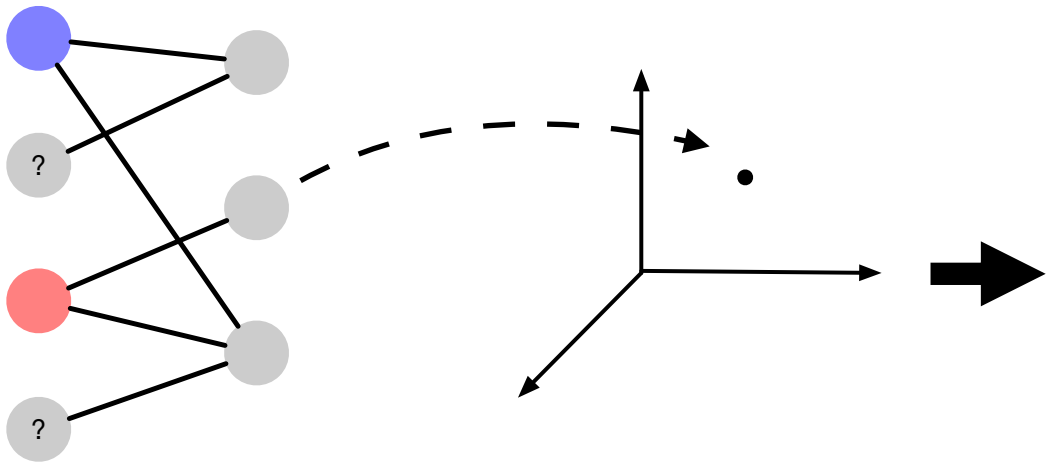
Outline

- Background & Related Work
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Outline

- Background & Related Work
  - Graph Representation Learning
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Traditional Learning Paradigm



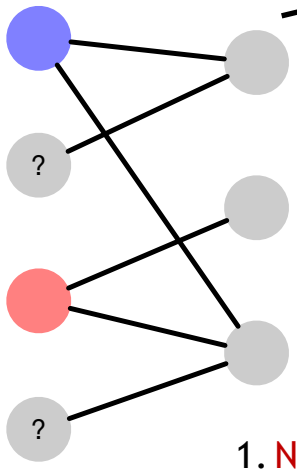
Used to solve the tasks:
- Node classification
- Link prediction
- Recommendation
- …

1. Node attributes;
2. Network structure;

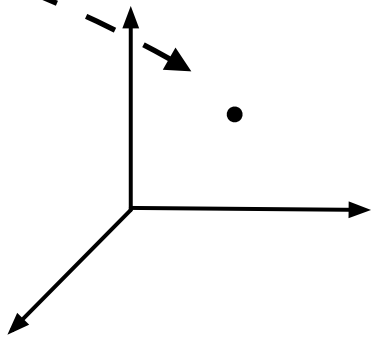
Hand-crafted features:
1. Node attributes;
2. Network structure: statistic of the network
   - Node degree;
   - Eccentricity;
   - Radius;
   - …

Machine learning models
- Logistic regression;
- Random forest;
- Neural networks;
- …

Graph Representation Learning



- DeepWalk [1];
- Node2Vec [2];
- Matrix factorization;
- GCN [3];
- GraphSAGE [4];
- …



Embedding space

1. Node attributes;
2. Network structure;

- Data-driven;
- No feature engineering: end-to-end training;
- Attributes and higher-order network structure information are jointly encoded;

Used to solve the tasks:

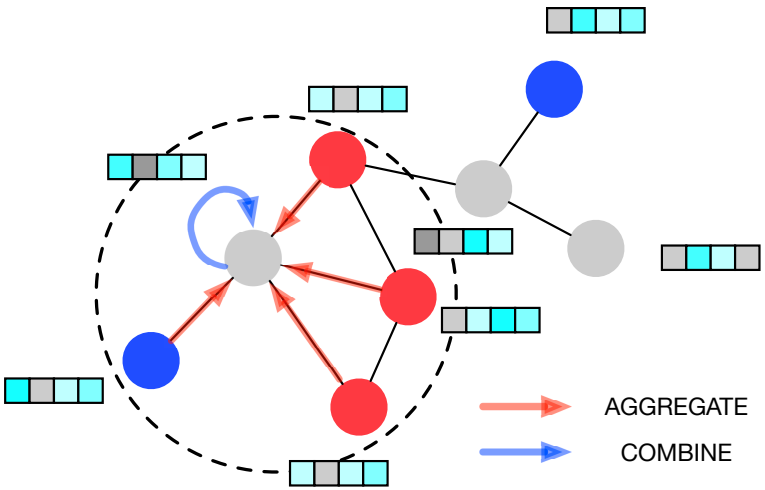
- Node classification
- Link prediction
- Recommendation
- …

[1] Bryan Perozzi etc. KDD 2014;
[2] Aditya Grover etc. KDD 2016;
[3] Thomas N Kipf etc. ICLR 2017;
[4] Will Hamilton etc. NeurIPS 2017

Outline

- Background & Related Work
  - Graph Representation Learning
  - Graph Neural Networks
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Graph Neural Networks



- Message passing: a general framework

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

Layer

$$\mathcal{L} = Loss(h^{(K)}, y)$$

- Graph Convolutional Network (GCN)

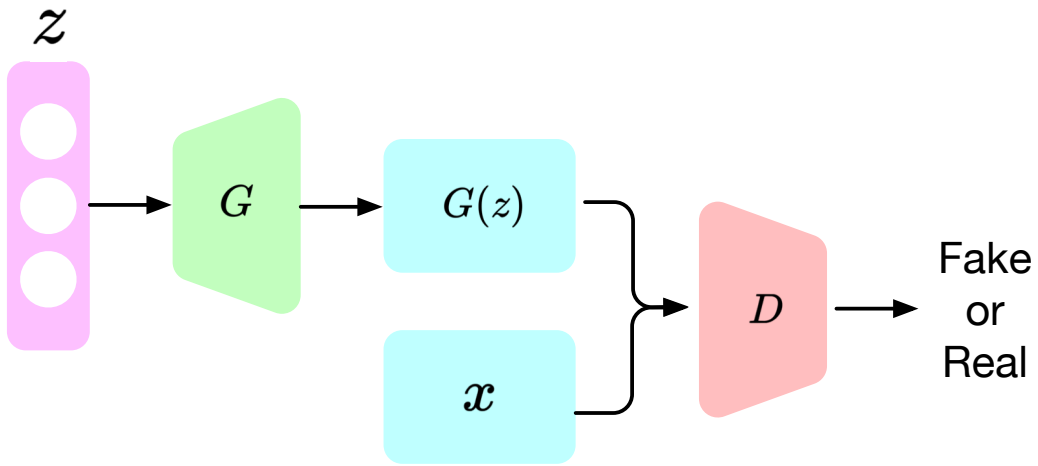
Layer

$$h_v^{(k)} = \text{ReLU}\left(W \cdot \text{MEAN}\left\{h_u^{(k-1)}, \forall u \in \mathcal{N}(v) \cup \{v\}\right\}\right)$$

$$\mathcal{L} = H(h^{(K)}, y)$$

[1] Thomas N Kipf etc. ICLR 2017;
[2] Keyulu Xu etc. ICLR 2019;

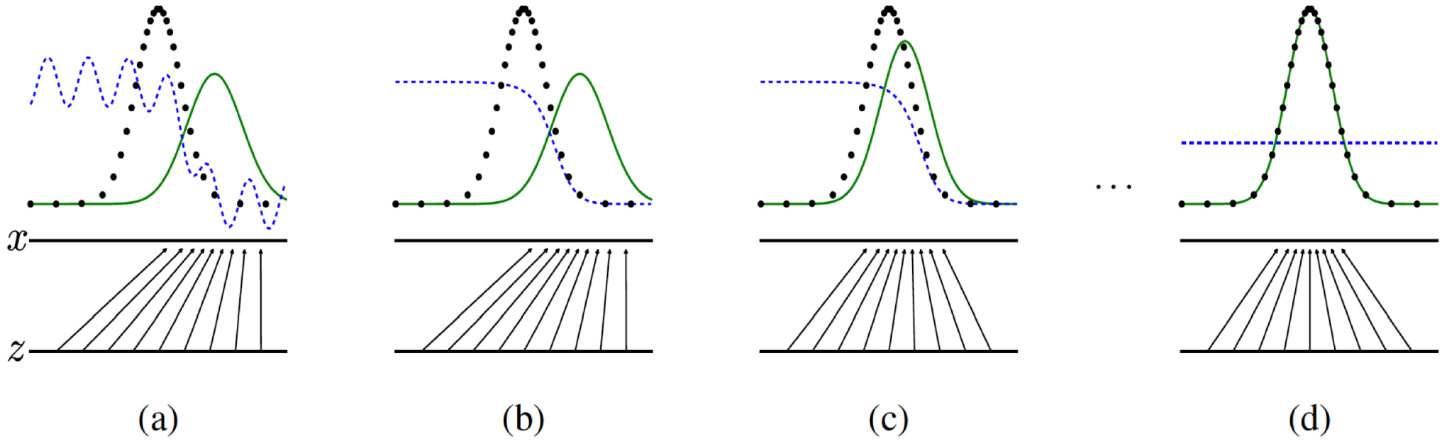
Generative Adversarial Networks (GANs)



$$\min_{\boldsymbol{\theta}^G} \max_{\boldsymbol{\theta}^D} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\log D\left(\boldsymbol{x};\boldsymbol{\theta}^D\right)\right] + \mathbb{E}_{\boldsymbol{z}\sim p_g}\left[\log\left(1 - D\left(G\left(\boldsymbol{z};\boldsymbol{\theta}^G\right);\boldsymbol{\theta}^D\right)\right)\right]$$

[1] Ian Goodfellow etc. NeurIPS 2014.

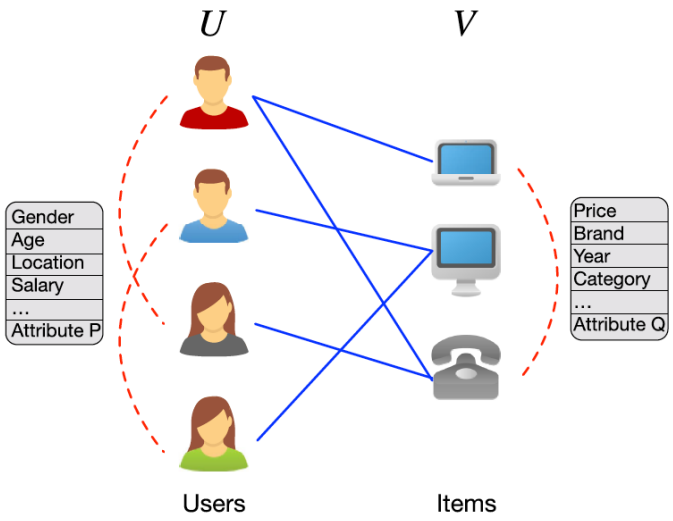
- Non-linear distributions mapping

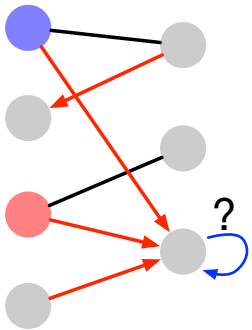


Outline

- Background & Related Work
- Challenges for Bipartite Graphs
  - Domain Inconsistency
  - Efficiency
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Domain Inconsistency





- Distinct attribute distributions;
- Message passing is failed or suboptimal;

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

Possible solutions:

- Convert to graphs with one domain?
  - No feature correlations
- Projection or concatenation?
  - Too simple
- Heterogeneous graph?
  - Only one type of meta-path

Efficiency

- E-commerce networks: millions of nodes and edges

- Scalable & Label Efficiency
  - GNNs are hard for scaling;
  - Extremely few labeled samples (unsupervised);

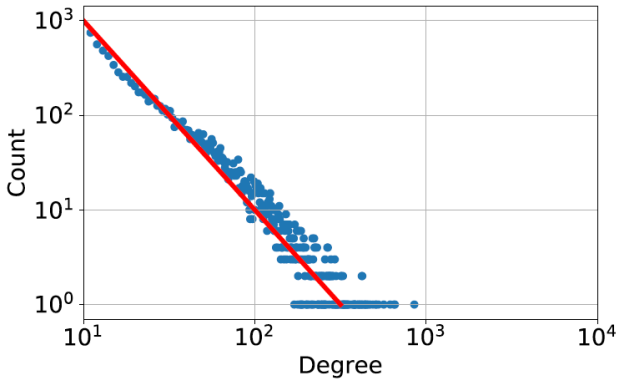


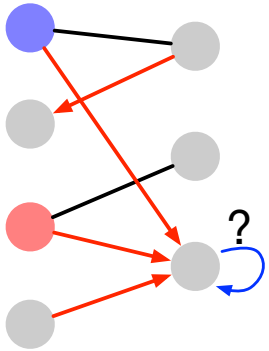
Fig. 2. Power-law distribution of node degrees in a user–item network from the Amazon dataset [33].

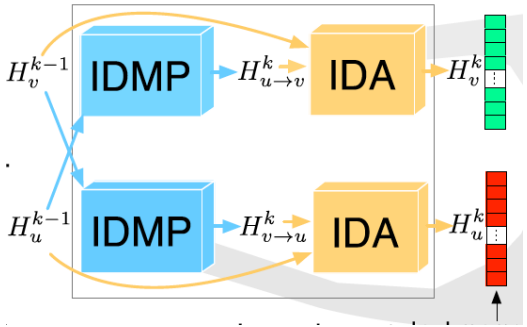
Possible solution:
- Sampling nodes or subgraphs?
  - High bias and variance for the message passing

Outline

- Background & Related Work
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

L-BGNN: Layerwise Trained Bipartite Graph Neural Network





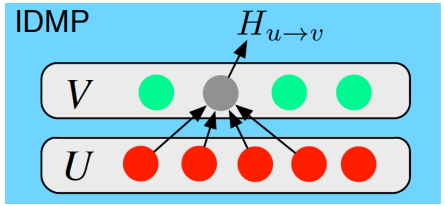
- First challenge: domain inconsistency
- Proposed solutions
  - Interdomain message passing (IDMP)
  - Intradomain alignment (IDA)

IDMP ≈ Aggregate operator

IDA ≈ Combine operator

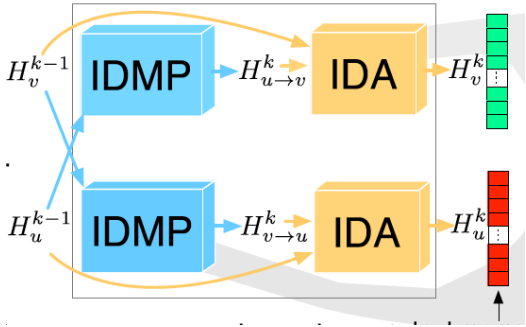
[1] Tian Xie etc. TNNLS 2022;

IDMP: Interdomain Message Passing

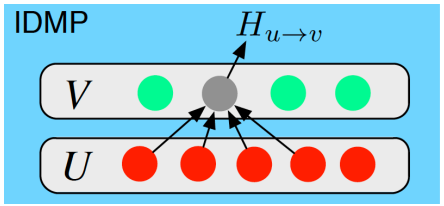


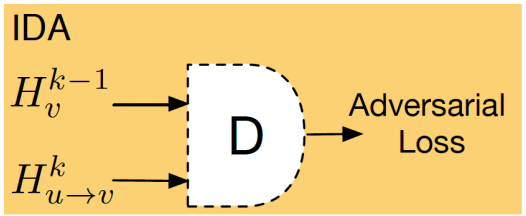
- Aggregate message;
- Independent filter parameters;

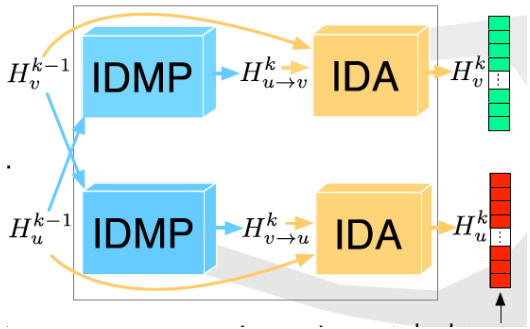
$$\begin{cases} \boldsymbol{H}_{v\to u}^{(k)} = \sigma\left(\hat{\boldsymbol{B}}_u \boldsymbol{H}_v^{(k)} \boldsymbol{W}_u^{(k)}\right) \\ \boldsymbol{H}_{u\to v}^{(k)} = \sigma\left(\hat{\boldsymbol{B}}_v \boldsymbol{H}_u^{(k)} \boldsymbol{W}_v^{(k)}\right) \end{cases} \qquad \boldsymbol{A} = \begin{pmatrix} \boldsymbol{0}_u & \boldsymbol{B}_u \\ \boldsymbol{B}_v & \boldsymbol{0}_v \end{pmatrix}$$



IDA: Intradomain Alignment







- Map domain distributions in an adversarial way.
- Intuition: neighbor nodes should share similar embedding distributions.

IDA (discriminator) loss:

$$\mathcal{L}_D(\boldsymbol{\theta}^D \mid \boldsymbol{\theta}^G) = -\frac{1}{M}\sum_{i=1}^{M}\log P_{\boldsymbol{\theta}^D,\boldsymbol{\theta}^G}\left(\text{ source } = 0 \mid \boldsymbol{h}_{u,i}^{(k)}\right) - \frac{1}{M}\sum_{i=1}^{M}\log P_{\boldsymbol{\theta}^D,\boldsymbol{\theta}^G}\left(\text{ source } = 1 \mid \boldsymbol{h}_{v\to u,i}^{(k)}\right)$$

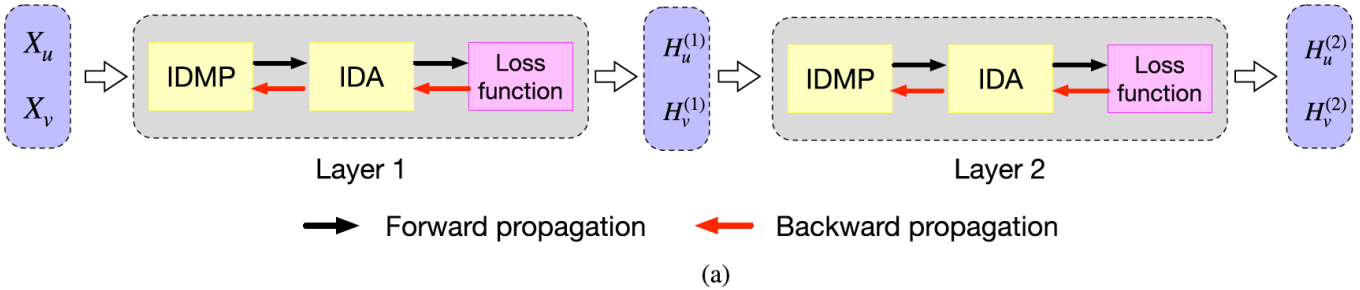
IDMP (generator) loss:

$$\mathcal{L}_G(\boldsymbol{\theta}^G \mid \boldsymbol{\theta}^D) = -\frac{1}{M}\sum_{i=1}^{M}\log P_{\boldsymbol{\theta}^D,\boldsymbol{\theta}^G}\left(\text{ source } = 0 \mid \boldsymbol{h}_{v\to u,i}^{(k)}\right)$$

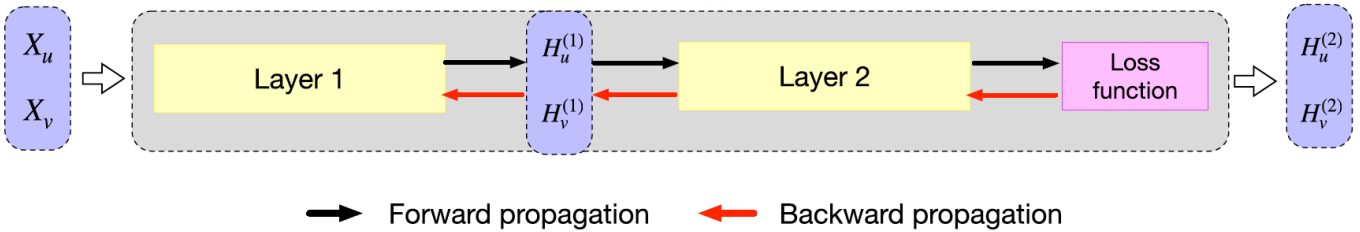
- Second challenge: scalable & label efficiency

- Proposed solutions: Layerwise training
  - Predictions from the IDMP are served as input in next training layer.

$$\begin{cases} \boldsymbol{H}_u^{(k+1)} = \boldsymbol{H}_{v\to u}^{(k)} \\ \boldsymbol{H}_v^{(k+1)} = \boldsymbol{H}_{u\to v}^{(k)} \end{cases}$$

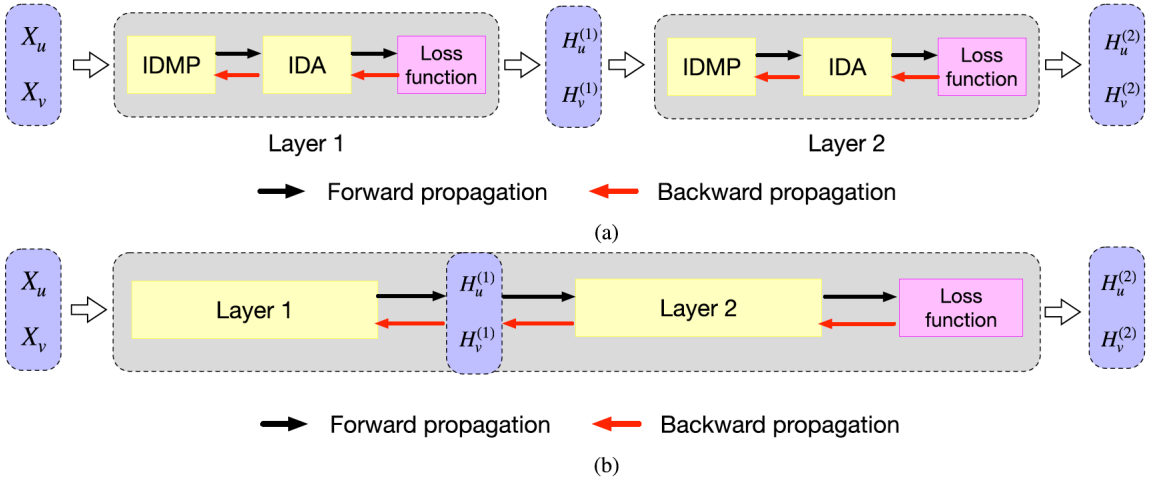


(a)

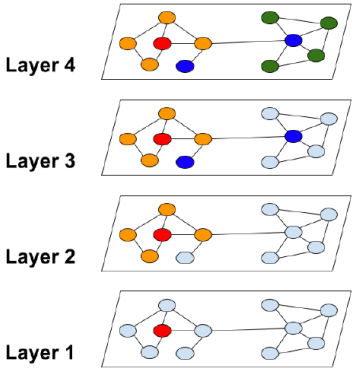


(b)

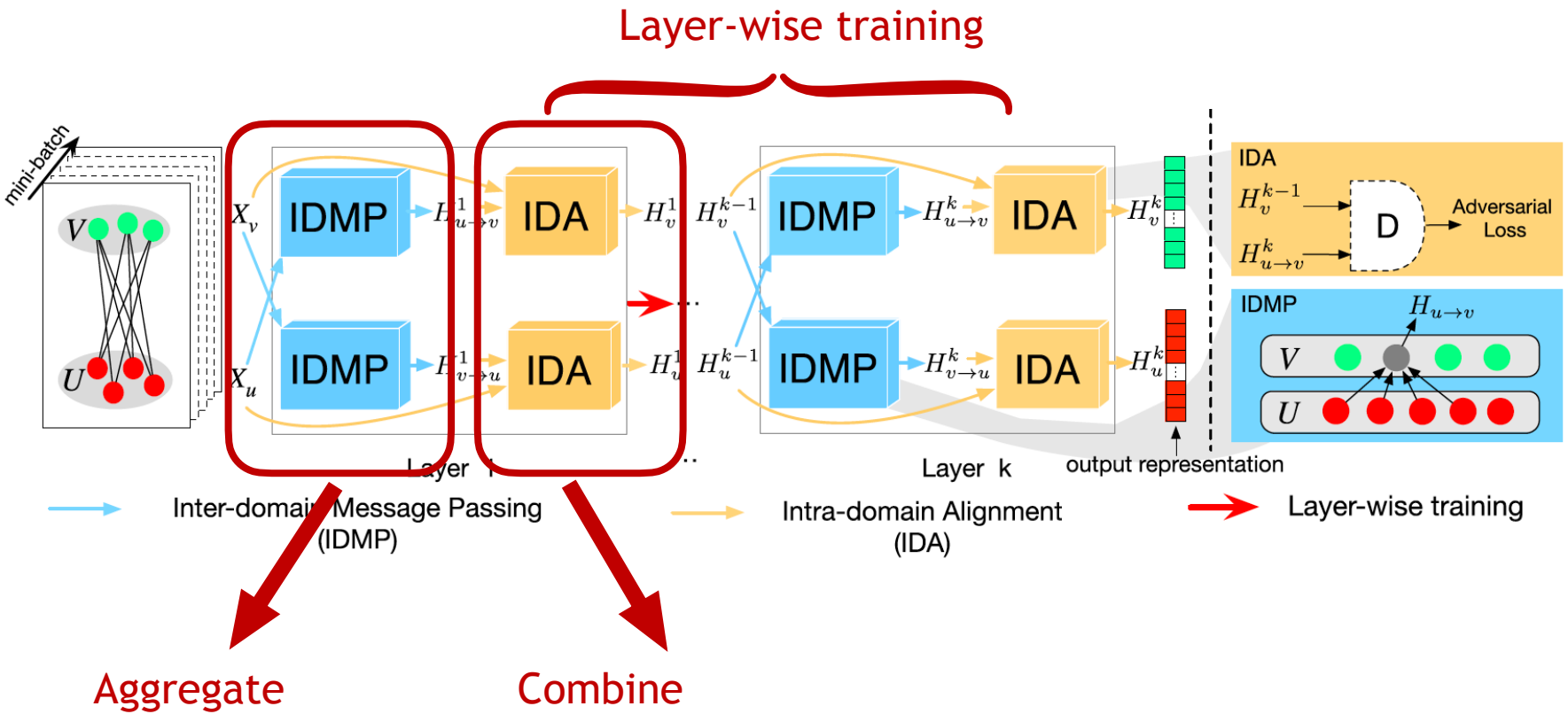
Layerwise Training



- Scalable
  - Only one training layer is alive;
  - No exponential neighborhood expansion: minibatch training;
- Without statistical performance sacrifice (Theorem 1)
  - Also can encode higher-order information;



L-BGNN


Layer-wise training
mini-batch
V
U
X_v
X_u
IDMP
IDA
$H_{u\to v}^1$
$H_{v\to u}^1$
H_v^1
H_u^1
H_v^{k-1}
H_u^{k-1}
$H_{u\to v}^k$
$H_{v\to u}^k$
H_v^k
H_u^k
Layer 1
Layer k
output representation
IDA
D
Adversarial Loss
IDMP
$H_{u\to v}$
Inter-domain Message Passing (IDMP)
Intra-domain Alignment (IDA)
Layer-wise training
Aggregate
Combine

Outline

- Background & Related Work
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Datasets

TABLE I

STATISTICS OF EIGHT DATASETS USED IN OUR EXPERIMENTS

Datasets	#Nodes U	#Nodes V	#Edges	#Feat U	#Feat V	#Classes
Group	619,030	90,044	991,734	8	16	2
Cora	734	877	1,802	1,433	1,000	7
CiteSeer	613	510	1,000	3,703	3,000	6
PubMed	13,424	3,435	18,782	400	500	3
Movielens	6,040	3,706	1,000,209	N.A.		N.A.
DBLP	6,001	1,308	29,254	N.A.		N.A.
Yelp	2,614	1,286	30,837	N.A.		N.A.
Amazon	6,170	2,753	195,790	N.A.		N.A.

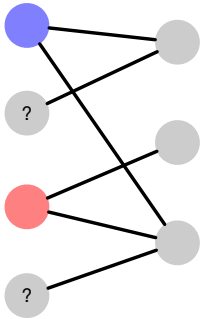
- Real datasets
  - Movielens (user – movies)
  - DBLP (author – paper)
  - Yelp (user – business)
  - Amazon (user – item)
  - Group (user – community)
- Synthetic datasets
  - Cora
  - CiteSeer
  - PubMed

Evaluation: Node Classification

TABLE II

PERFORMANCE COMPARISON FOR THE NODE CLASSIFICATION TASK, WHERE OOM DENOTES "OUT OF MEMORY"

	Group	Cora		CiteSeer		PubMed	
Methods	Accuracy	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
node2vec	0.577	0.682	0.663	0.569	0.476	0.700	0.685
ANRL	OOM	0.797	0.785	0.748	0.678	0.390	0.187
VGAE	OOM	0.782	0.754	0.732	0.645	0.823	0.828
GraphSAGE-MEAN	0.580	0.823	0.801	0.748	0.665	0.838	0.843
HeGAN	$>$ 2 days	0.189	0.167	0.228	0.224	0.396	0.298
FeatWalk	0.514	0.237	0.164	0.276	0.213	0.408	0.306
metapath2vec	OOM	0.628	0.596	0.577	0.466	0.609	0.576
HIN2Vec	0.518	0.635	0.593	0.561	0.514	0.630	0.599
BiNE	$>$ 2 days	0.372	0.077	0.293	0.186	0.391	0.297
BiANE	$>$ 2 days	0.791	0.773	0.739	0.654	0.793	0.799
L-BGNN-Adv	**0.622**	**0.859**	**0.831**	**0.768**	**0.698**	**0.857**	**0.860**



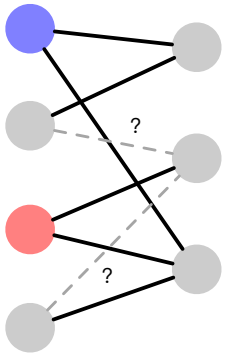
- Train a logistic classifier on the node embeddings;
- Metrics: accuracy; micro (macro) F1;
- SOTA performance, especially on the large-scale graph;

Evaluation: Link Prediction

TABLE III

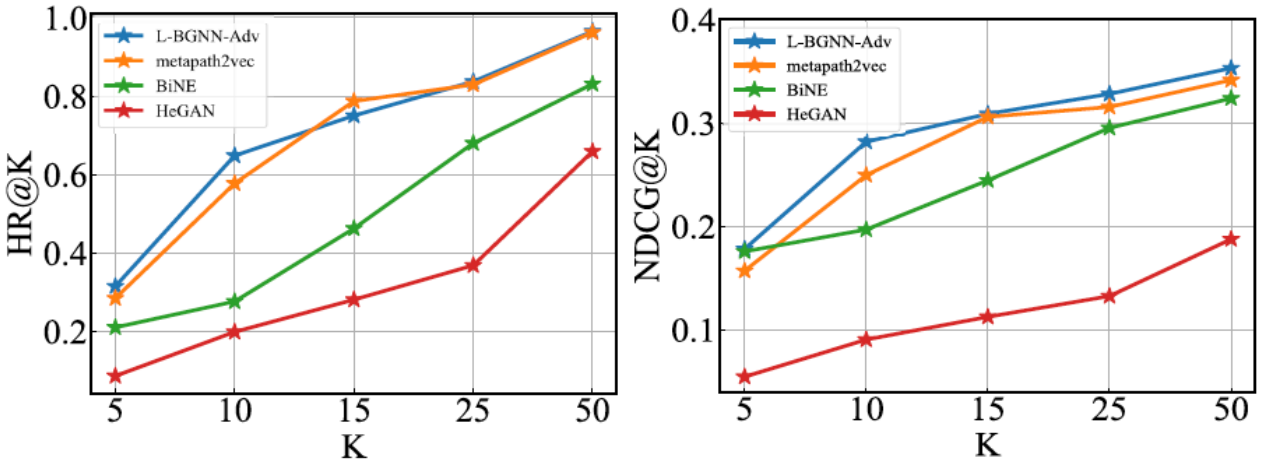
PERFORMANCE COMPARISON FOR THE LINK PREDICTION TASK

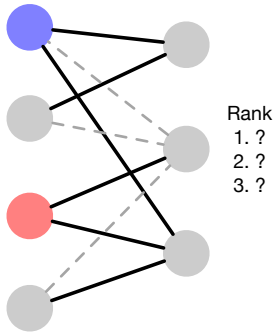
	DBLP			Yelp			Amazon		
Methods	Accuracy	Macro F1	AUC	Accuracy	Macro F1	AUC	Accuracy	Macro F1	AUC
node2vec	0.679	0.669	0.764	0.666	0.666	0.720	0.665	0.665	0.719
ANRL	0.791	0.790	0.853	0.840	0.839	0.890	0.717	0.717	0.795
VGAE	0.553	0.455	0.701	0.760	0.749	0.908	0.623	0.608	0.726
GraphSAGE-MEAN	0.736	0.735	0.760	0.707	0.707	0.712	0.702	0.702	0.704
HeGAN	0.571	0.569	0.610	0.516	0.516	0.520	0.519	0.518	0.525
FeatWalk	0.560	0.553	0.567	0.562	0.554	0.579	0.624	0.623	0.695
metapath2vec	0.822	0.820	0.903	0.846	0.846	0.923	0.741	0.741	0.815
HIN2Vec	0.855	0.854	0.926	0.752	0.744	0.877	0.678	0.672	0.767
BiNE	0.600	0.560	0.780	0.603	0.593	0.660	0.568	0.515	0.713
BiANE	0.858	0.858	0.927	0.857	0.856	0.918	0.739	0.736	0.820
L-BGNN-Adv	**0.861**	**0.860**	**0.931**	**0.882**	**0.882**	**0.932**	**0.749**	**0.749**	**0.827**



- Train a logistic classifier on the node embeddings;
- Metrics: accuracy; macro F1; AUC;
- SOTA performance;

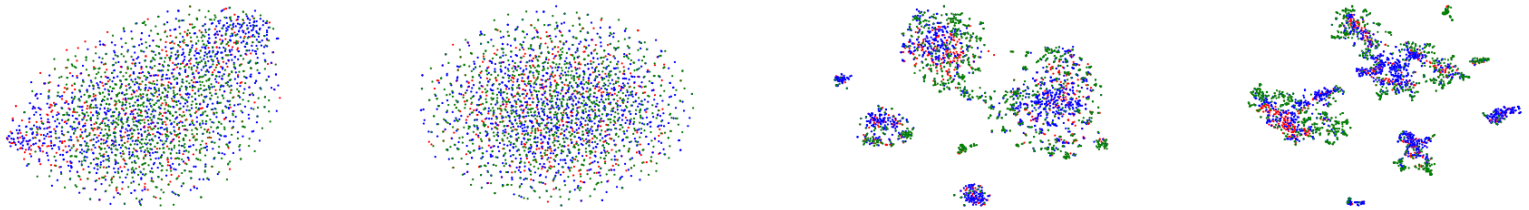
Evaluation: Recommendation





- Leave-one-out method for evaluation;
- Metrics: hit ratio @ k; normalized discounted cumulative gain @ k;
- SOTA performance;

Evaluation: Embedding Visualization



(a) (b) (c) (d)

Fig. 6. Visualization of embedding results on the Yelp dataset, where colors are used to indicate different business categories. (a) BiNE. (b) HeGAN. (c) metapath2vec. (d) L-BGNN-Adv.

- t-SNE visualizes the node embeddings of the Yelp dataset;
- Different colors denote to different labels;
- Denser clusters and cleaner boundary;

Ablation Study: Effect of Intradomain Alignment (IDA)

TABLE IV

PERFORMANCE COMPARISON OF L-BGNN-ADV AGAINST SIMPLE DOMAIN ALIGNMENT, WHERE THE PERFORMANCE METRIC IS THE MICRO F1 SCORE

	Group	Cora	CiteSeer	PubMed
Raw features	0.501	0.693	0.663	0.754
Aggregated features	0.555	0.825	0.612	0.757
Concatenated features	0.556	0.797	0.637	0.770
L-BGNN-Adv	**0.622**	**0.859**	**0.768**	**0.857**

TABLE V

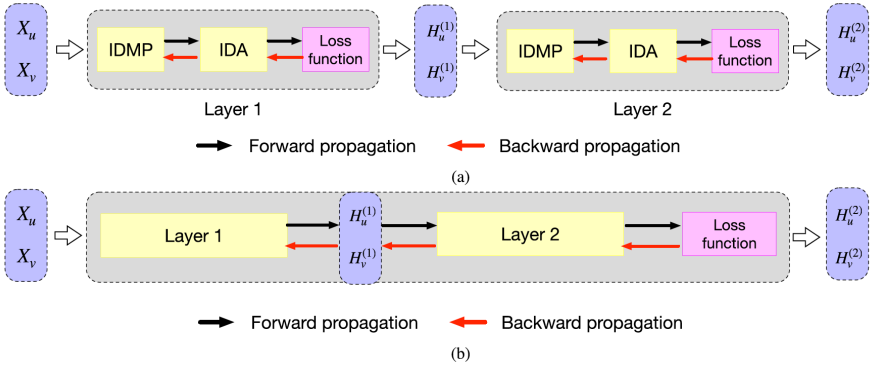
PERFORMANCE COMPARISON OF L-BGNN-ADV AND L-BGNN-MLP, WHERE THE PERFORMANCE METRIC IS THE MICRO F1 SCORE

	Group	Cora	CiteSeer	PubMed
L-BGNN-MLP	0.582	0.784	0.756	0.846
L-BGNN-Adv	**0.622**	**0.859**	**0.768**	**0.857**

- Naive feature engineering;
- Linear domain alignment

$$\mathcal{L}_u = \left\| \mathrm{MLP}\left(\boldsymbol{H}_{v \to u}^{(k)} \right) - \boldsymbol{H}_u^{(k)} \right\|_F^2$$

Ablation Study: Effect of Layerwise Training



	End-to-end training	Layerwise training
Group	OOM	**0.622**
Cora	0.837/0.809	**0.859/0.831**
CiteSeer	0.685/0.642	**0.768/0.698**
PubMed	**0.859**/0.859	0.857/**0.860**

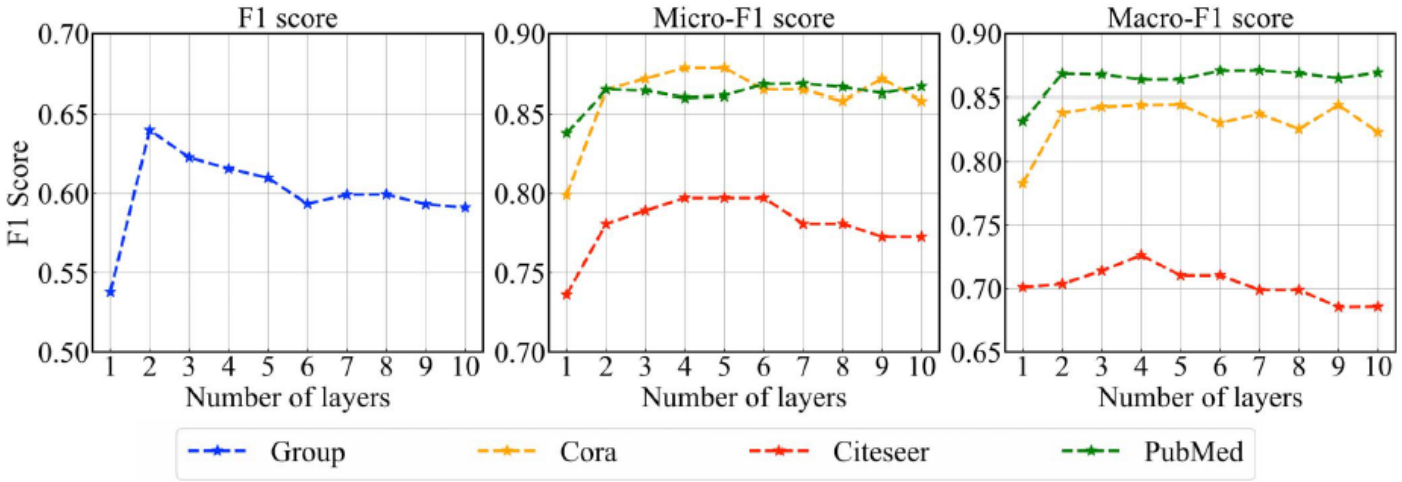
- Adopt concatenation in hidden layers;

$$\begin{cases} \boldsymbol{H}_u^{(k+1)} = \mathrm{CONCAT}\left(\boldsymbol{H}_{v\to u}^{(k)}, \boldsymbol{H}_u^{(k)}\right) \\ \boldsymbol{H}_v^{(k+1)} = \mathrm{CONCAT}\left(\boldsymbol{H}_{u\to v}^{(k)}, \boldsymbol{H}_v^{(k)}\right). \end{cases}$$

- Same adversarial alignment in the end;

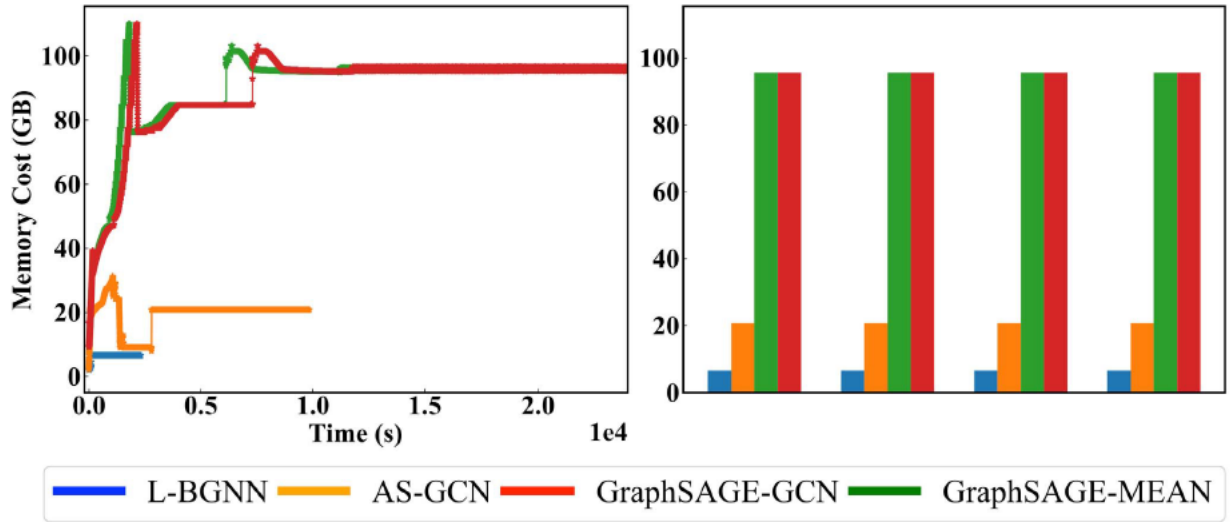
- End-to-end training fails on large-scale graphs;
- Layerwise training performs better;

Ablation Study: Effect of Layerwise Training



- Increase the number of training layers;
- Performance rises first, stabilizes next, and drops at the end;

Ablation Study: Efficiency for Large-scale Graphs



- Benchmarking methods: GNNs specific for large-scale graphs;
- Lower memory cost;
- Faster training speed;

Outline

- Background & Related Work
- Challenges for Bipartite Graphs
- Our Method: L-BGNN
- Experiments & Evaluation
- Conclusion

Conclusion

- Background & Related Work
  - Graph Representation Learning
  - Graph Neural Networks
- Challenges for Bipartite Graphs
  - Domain Inconsistency
  - Scalable & Label Efficiency
- Our Method: L-BGNN
  - Interdomain Message Passing (IDMP)
  - Intradomain Alignment (IDA)
  - Layerwise Training
- Experiments & Evaluation
  - SOTA Performance on Multiple Tasks
  - Ablation Study

Thank You!
Q&A