Propagation on Residuals and Rethinking

马佳明

2023.09.20

中国科学技术大学 大数据学院 数据智能实验室

中国科学技术大学
University of Science and Technology of China

Content

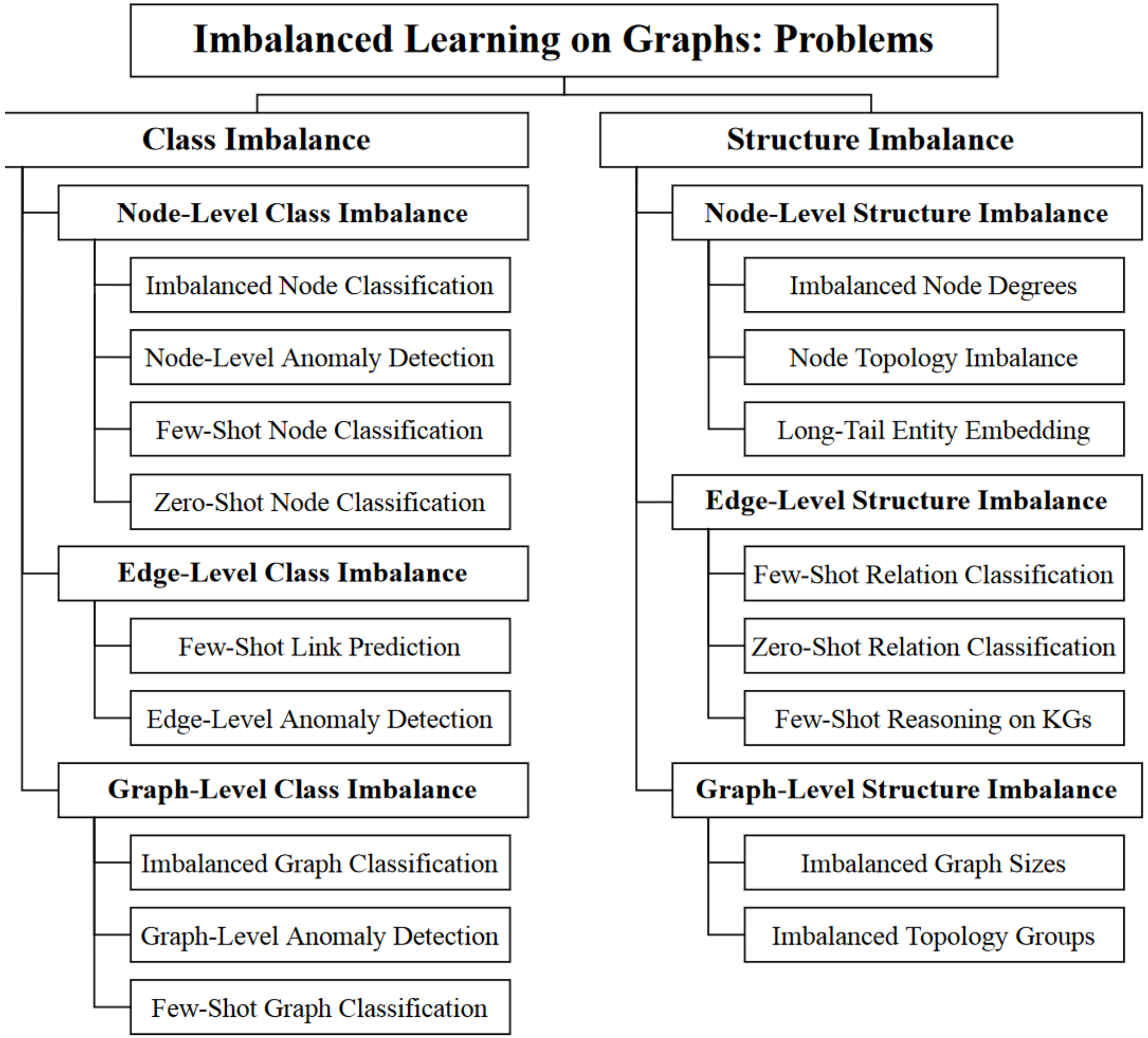
- Few-Shot on Graph
- Scenarios and Definition of Transductive Node Classification
- Traditional Method for TNC
- **Propagation on <u>Residuals</u>**
- Rethinking about Future

1. Few-show on Graph

Few-shot on Graph?

- <u>Node Classification (Transductive)</u>
- Link Prediction
- Graph Classification

- Relation Classification
- Reasoning on Knowledge-Graph



—— *Deep Long-Tailed Learning: A Survey, Zhang & etc., PAMI 2023.*

—— *A Survey of Imbalanced Learning on Graphs: Problems, Techniques, and Future Directions, Liu & etc., arXiv 2023.*

—— *A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection, Jin & etc., arXiv 2023.*

- **CV: 1st Transductive Inference Explicitly in Few-Shot Learning.**

—— Learning to Propagate Labels: Transductive Propagation Network for Few-shot Learning. Liu & etc., ICLR 2019.

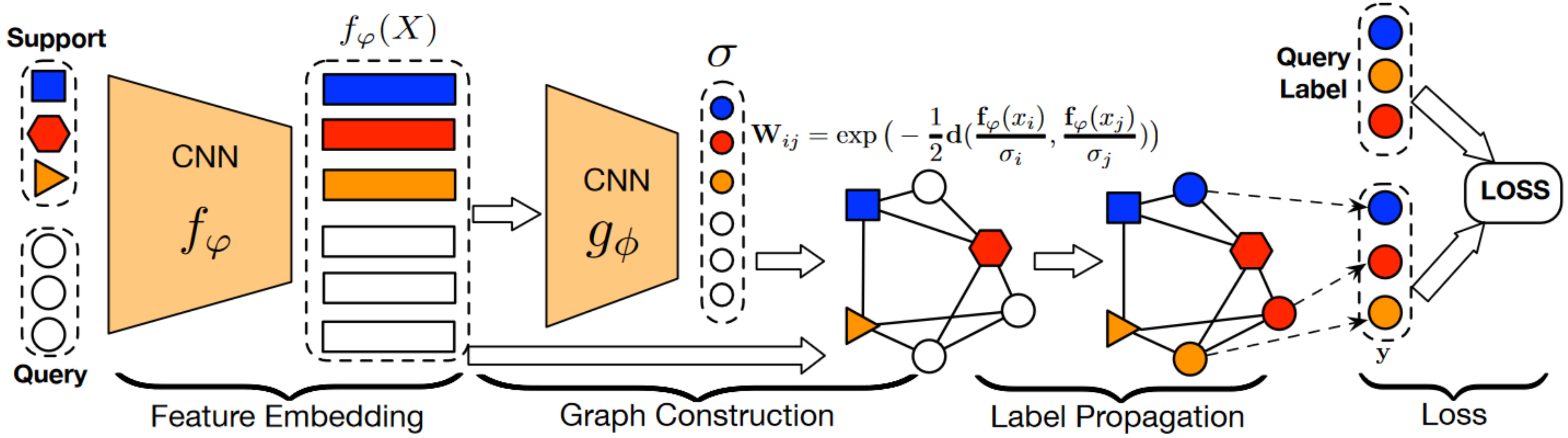
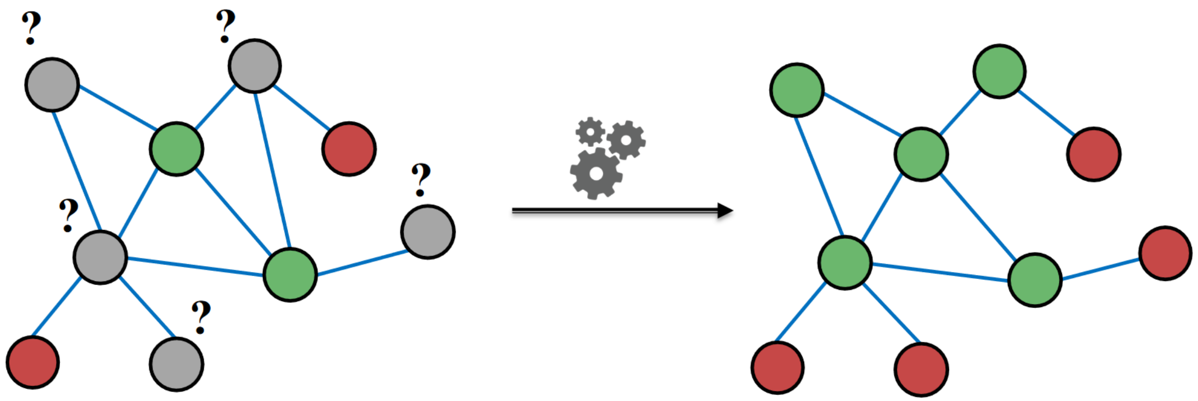


Figure 2: The overall framework of our algorithm in which the manifold structure of the entire query set helps to learn better decision boundary. The proposed algorithm is composed of four components: feature embedding, graph construction, label propagation, and loss generation.

2. Scenarios and Definition on Transductive Node Classification

Scenarios: Transductive Node Classification

- 通过使用图上已知标记(Label)的节点，来预测图上未知标记的节点的标记。
- 仅关注于当前图中的节点，而不考虑将来可能出现的新节点。
- Always Few-shot!

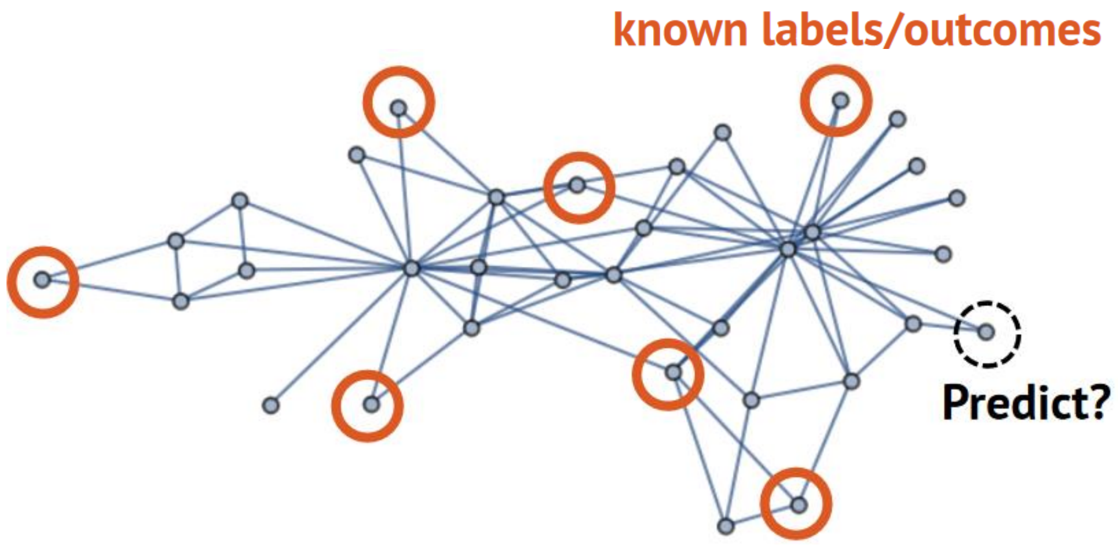


// **Semi-supervised:** 训练数据集中只有部分数据标记有标签，而大部分数据没有标签。目标是通过利用有标签数据和无标签数据的信息（feature），来进行模型训练和预测。
// **Inductive:** 通过使用已知标记的节点来构建一个模型，然后将该模型应用于未来出现的新节点。它更加关注于构建通用的模型，可以应用于未来的节点分类任务。（GraphSAGE）
—— *Inductive Representation Learning on Large Graphs. Hamilton & etc., NIPS 2017.*

Naïve Definition

- Input:
  - Graph $G = (V,\ E,\ X,\ Y)$.
  - Node Feature $X \in \mathbb{R}^{|V| \times d}$.
  - Known Node Label $y_L \in \{0,\ 1\}^{|V_L|}$.
    ($V_L \in V$ The labeled node set.)

- Output:
  - Unknown Node Label $y_u \in \{0,\ 1\}^{|V_U|}$.
    ($V_U = V \setminus V_L \in V$ the unlabeled node set.)



3. Traditional Method for TNC

- **1. Simplicial Label Propagation or GNNs.**

—— Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. Zhu & etc., ICML 2003.

1. Simplicial Label Propagation or GNNs

- **LP assumptions:**

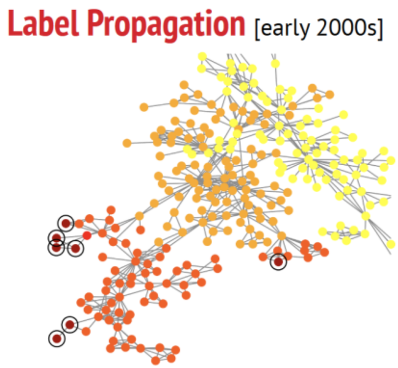
  1. Homophily

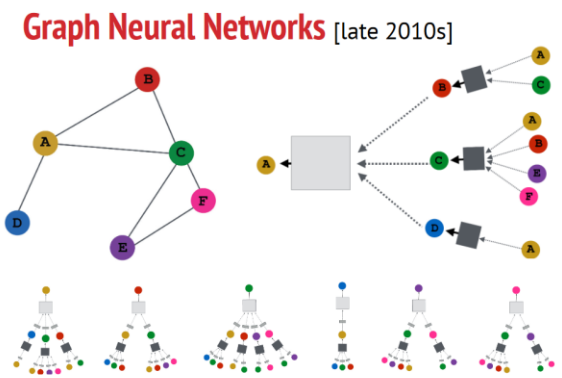
- **GNNs assumptions:**

  1. Labels only depend on neighbor features.

  2. Features are informative.

- **Propagation ≈ Smoothness**





	Training		Inference	
Model	**Feature**	**Label**	**Feature**	**Label**
LPA		✓		✓
GCN	✓		✓	

—— *Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. Zhu & etc., ICML 2003.*

- **1. Simplicial Label Propagation or GNNs.**

—— Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. Zhu & etc., ICML 2003.

- **2. GNNs + Propagation.**

—— Predict then Propagate: Graph Neural Networks meet Personalized PageRank. Gasteiger & etc., ICLR 2019.

—— Unifying Graph Convolutional Neural Networks and Label Propagation. Wang & Leskovec., ariXiv 2020.

2. GNNs + Propagation

PPNP & APPNP:

- GCN + PageRank :

  1. Personalization

  2. Approximation

  3. **Embedding** Propagation

  (≈ Feature Propagation)

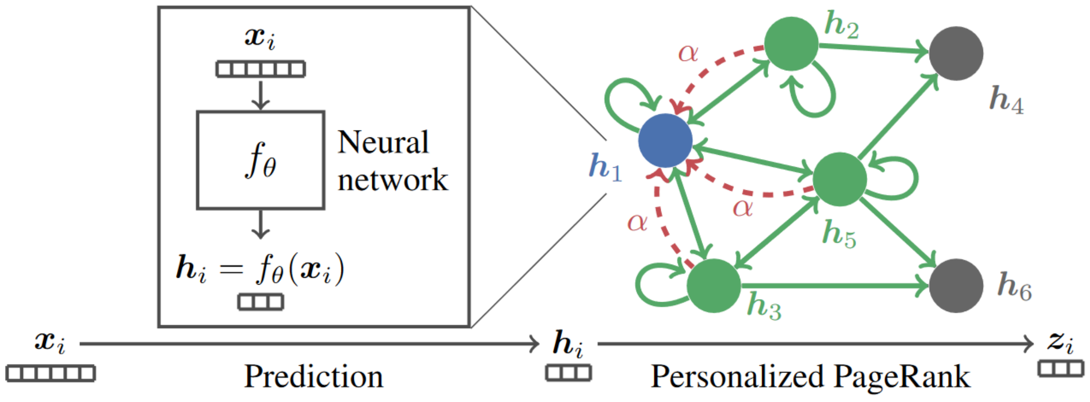


Figure 1: Illustration of (approximate) personalized propagation of neural predictions (PPNP, APPNP). Predictions are first generated from each node's own features by a neural network and then propagated using an adaptation of personalized PageRank. The model is trained end-to-end.

Table 2: Average accuracy with uncertainties showing the 95 % confidence level calculated by bootstrapping. Previously reported improvements vanish on our rigorous experimental setup, while PPNP and APPNP significantly outperform the compared models on all datasets.

Model	CITESEER	CORA-ML	PUBMED	MS ACADEMIC
V. GCN	73.51 ± 0.48	82.30 ± 0.34	77.65 ± 0.40	91.65 ± 0.09
GCN	75.40 ± 0.30	83.41 ± 0.39	78.68 ± 0.38	92.10 ± 0.08
N-GCN	74.25 ± 0.40	82.25 ± 0.30	77.43 ± 0.42	92.86 ± 0.11
GAT	75.39 ± 0.27	84.37 ± 0.24	77.76 ± 0.44	91.22 ± 0.07
JK	73.03 ± 0.47	82.69 ± 0.35	77.88 ± 0.38	91.71 ± 0.10
Bt. FP	73.55 ± 0.57	80.84 ± 0.97	72.94 ± 1.00	91.61 ± 0.24
PPNP*	$\mathbf{75.83 \pm 0.27}$	$\mathbf{85.29 \pm 0.25}$	-	-
APPNP	75.73 ± 0.30	85.09 ± 0.25	$\mathbf{79.73 \pm 0.31}$	$\mathbf{93.27 \pm 0.08}$

*out of memory on PUBMED, MS ACADEMIC (see efficiency analysis in Section 3)

—— Predict then Propagate: Graph Neural Networks meet Personalized PageRank. Gasteiger & etc., ICLR 2019.

—— The Anatomy of a Large-Scale Hypertextual Web Search Engine. Brin & Page, Computer networks and ISDN systems, 1998.

2. GNNs + Propagation

PPNP & APPNP:

- GCN + PageRank :

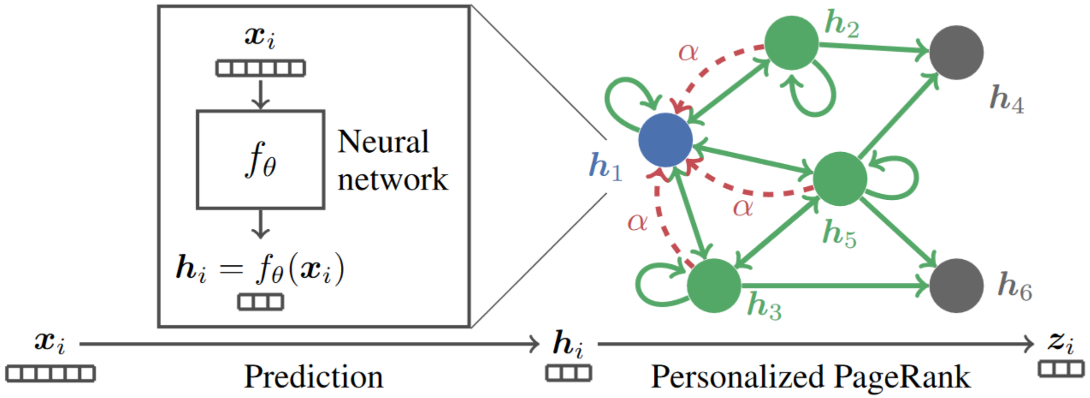
1. Personalization

2. Approximation

3. **Embedding** Propagation (≈ Feature Propagation)

4. No labels utilized during training and inference.

5. Marginal improvement.



Model	Training Feature	Training Label	Inference Feature	Inference Label
LPA		✓		✓
GCN	✓		✓	
APPNP	✓		✓	

—— *Predict then Propagate: Graph Neural Networks meet Personalized PageRank. Gasteiger & etc., ICLR 2019.*
—— *The Anatomy of a Large-Scale Hypertextual Web Search Engine. Brin & Page, Computer networks and ISDN systems, 1998.*

2. GNNs + Propagation

Method	**Cora**	**Citeseer**	**Pubmed**	**Coauthor-CS**	**Coauthor-Phy**
MLP	64.6 ± 1.7	62.0 ± 1.8	85.9 ± 0.3	91.7 ± 1.4	94.1 ± 1.2
LR	77.3 ± 1.8	71.2 ± 1.8	86.0 ± 0.6	91.1 ± 0.6	93.8 ± 1.1
LPA	85.3 ± 0.9	70.0 ± 1.7	82.6 ± 0.6	91.3 ± 0.2	94.9 ± 0.4
GCN	88.2 ± 0.8	77.3 ± 1.5	87.2 ± 0.4	93.6 ± 1.5	96.2 ± 0.2
GAT	87.7 ± 0.3	76.2 ± 0.9	86.9 ± 0.5	93.8 ± 0.4	96.3 ± 0.7
JK-Net	**89.1** ± 1.2	78.3 ± 0.9	85.8 ± 1.1	92.4 ± 0.4	94.8 ± 0.4
GraphSAGE	86.8 ± 1.9	75.2 ± 1.1	84.7 ± 1.6	92.6 ± 1.6	94.5 ± 1.1
GCN-LPA	88.5 ± 1.5	**78.7** ± 0.6	**87.8** ± 0.6	**94.8** ± 0.4	**96.9** ± 0.2

Table 2: Mean and the 95% confidence intervals of test set accuracy for all methods and datasets.

GCN-LPA:

- GCN + LPA :

	Training		Inference	
Model	**Feature**	**Label**	**Feature**	**Label**
LPA		✓		✓
GCN	✓		✓	
APPNP	✓		✓	
GCN-LPA	✓	✓	✓	

1. LPA during training weight.
2. Overemphasis to Label.
3. Depending on GNNs with **trainable weight edge** like GAT.

—— *Unifying Graph Convolutional Neural Networks and Label Propagation. Wang & Leskovec., ariXiv 2020.*
—— *Graph Attention Networks. Velickovic & etc., ICLR 2018.*

- **1. Simplicial Label Propagation or GNNs.**

—— Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. Zhu & etc., ICML 2003.

- **2. GNNs + Propagation.**

—— Predict then Propagate: Graph Neural Networks meet Personalized PageRank. Gasteiger & etc., ICLR 2019.

—— Unifying Graph Convolutional Neural Networks and Label Propagation. Wang & Leskovec., ariXiv 2020.

- **3. Not only Propagation on Feature but also done so on Label.**

—— Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. Shi & etc., IJCAI 2021.

3. GNNs + L&F Propagation

UniMP:

- <u>GAT+ All Propagation:</u>

  1. L&F Propagation;

  2. Masked Known Label.

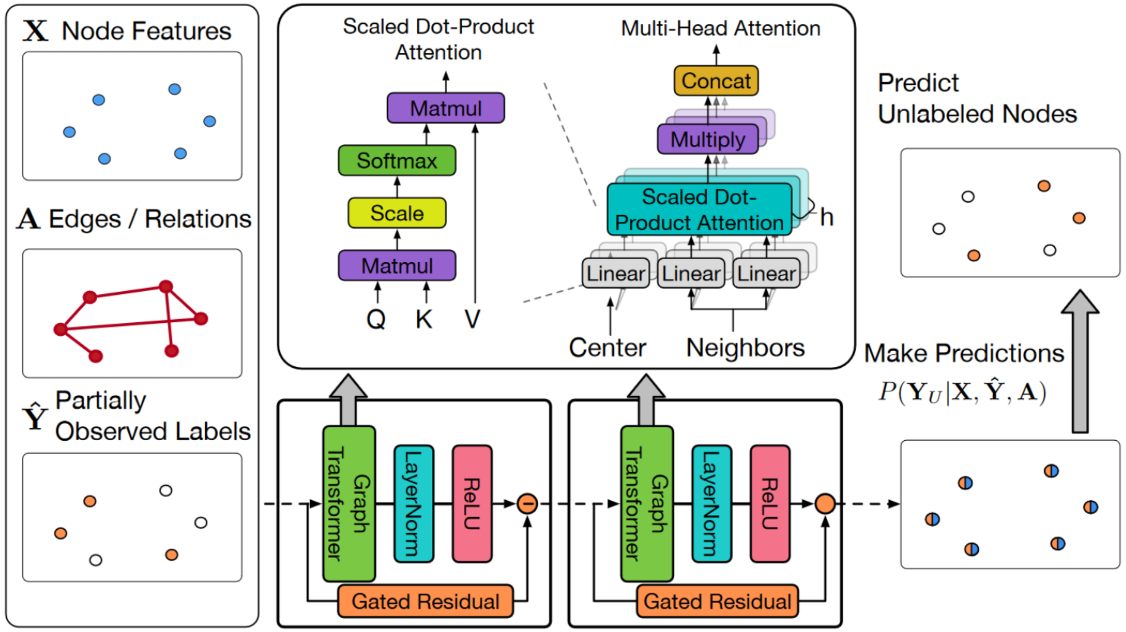


Figure 1: The architecture of UniMP.

	Training		Inference	
Model	**Feature**	**Label**	**Feature**	**Label**
LPA		✓		✓
GCN	✓		✓	
APPNP	✓		✓	
GCN-LPA	✓	✓	✓	
UniMP (Ours)	✓	✓	✓	✓

Table 1: Comparison the input information that message passing models use in training and inference.

—— *Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. Shi & etc., IJCAI 2021.*

—— *Training Sparse Neural Networks. Srinivas& etc., CVPR Workshops, 2017.*

3. GNNs + L&F Propagation

UniMP:

- GAT+ All Propagation:

  1. L&F Propagation;

  2. Masked Known Label.

- SOTA ! (2022.09)

Model	Test Accuracy	Validation Accuracy	Params
GCN-Cluster [Chiang *et al.*, 2019]	0.7897 ± 0.0036	0.9212 ± 0.0009	206,895
GAT-Cluster	0.7923 ± 0.0078	0.8985 ± 0.0022	1,540,848
GAT-NeighborSampling	0.7945 ± 0.0059	-	1,751,574
GraphSAINT [Zeng *et al.*, 2019]	0.8027 ± 0.0026	-	331,661
DeeperGCN [Li *et al.*, 2020]	0.8090 ± 0.0020	0.9238 ± 0.0009	253,743
UniMP	$\mathbf{0.8256 \pm 0.0031}$	$\mathbf{0.9308 \pm 0.0017}$	1,475,605

Table 4: Results for ogbn-products

Inputs	Model	Datasets ogbn-products Test ACC
$\mathbf{X}$	Multilayer Perceptron	0.6106 ± 0.0008
$\mathbf{X}, \mathbf{A}$	GCN	0.7851 ± 0.0011
	GAT	0.8002 ± 0.0063
	Graph Transformer	0.8137 ± 0.0047
$\mathbf{A}, \hat{\mathbf{Y}}$	GCN	0.7832 ± 0.0013
	GAT	0.7751 ± 0.0054
	Graph Transformer	0.7987 ± 0.0104
$\mathbf{X}, \mathbf{A}, \hat{\mathbf{Y}}$	GCN	0.7987 ± 0.0104
	GAT	0.8193 ± 0.0017
	Graph Transformer	$\mathbf{0.8256 \pm 0.0031}$
	∟ w/ Edge Feature	*

—— *Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. Shi & etc., IJCAI 2021.*
—— *Training Sparse Neural Networks. Srinivas& etc., CVPR Workshops, 2017.*

4. Propagation on Residuals

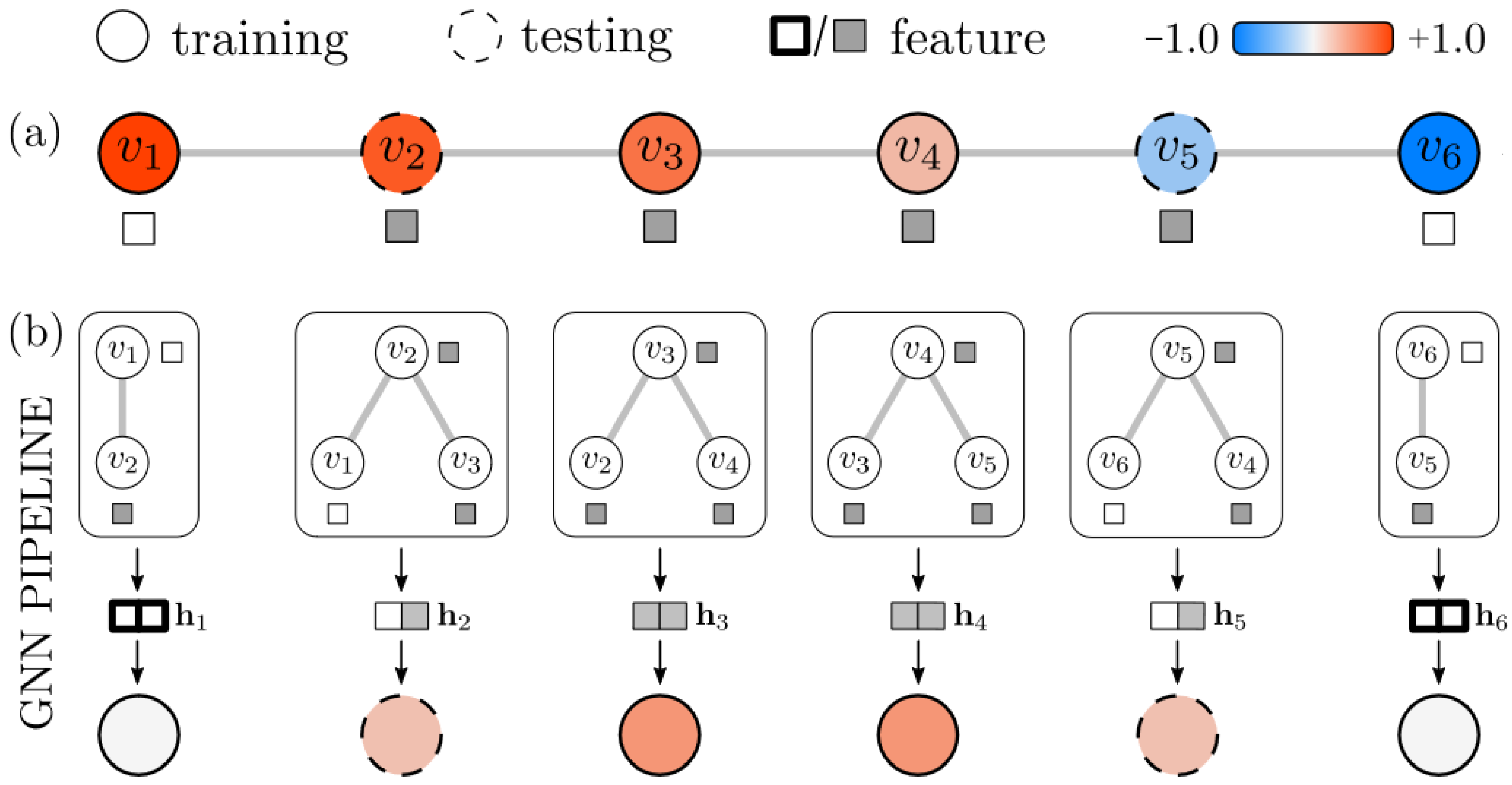
- **1. LP on residuals.**

—— Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.

1. LP on residuals for improving GNN regression algorithms.

Claim: GNNs tend to generate biased predictions, resulting in residuals.

Reason: Labels and features are not necessarily strongly correlated.

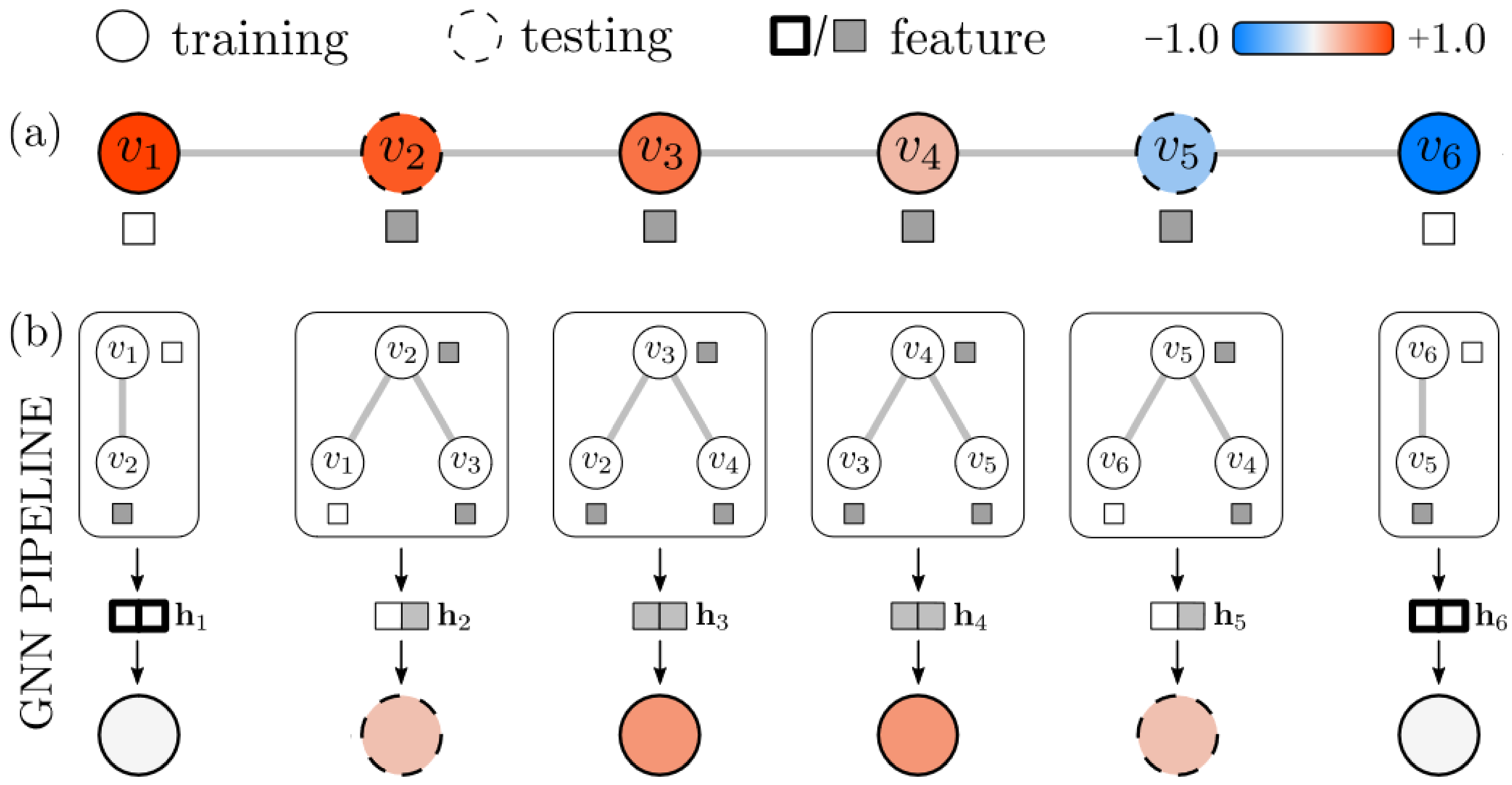


—— *Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.*

1. LP on residuals for improving GNN regression algorithms.

Problem: LPA appended can not figure out this situation.

Reason: The pseudo-label distribution has changed.



—— *Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.*

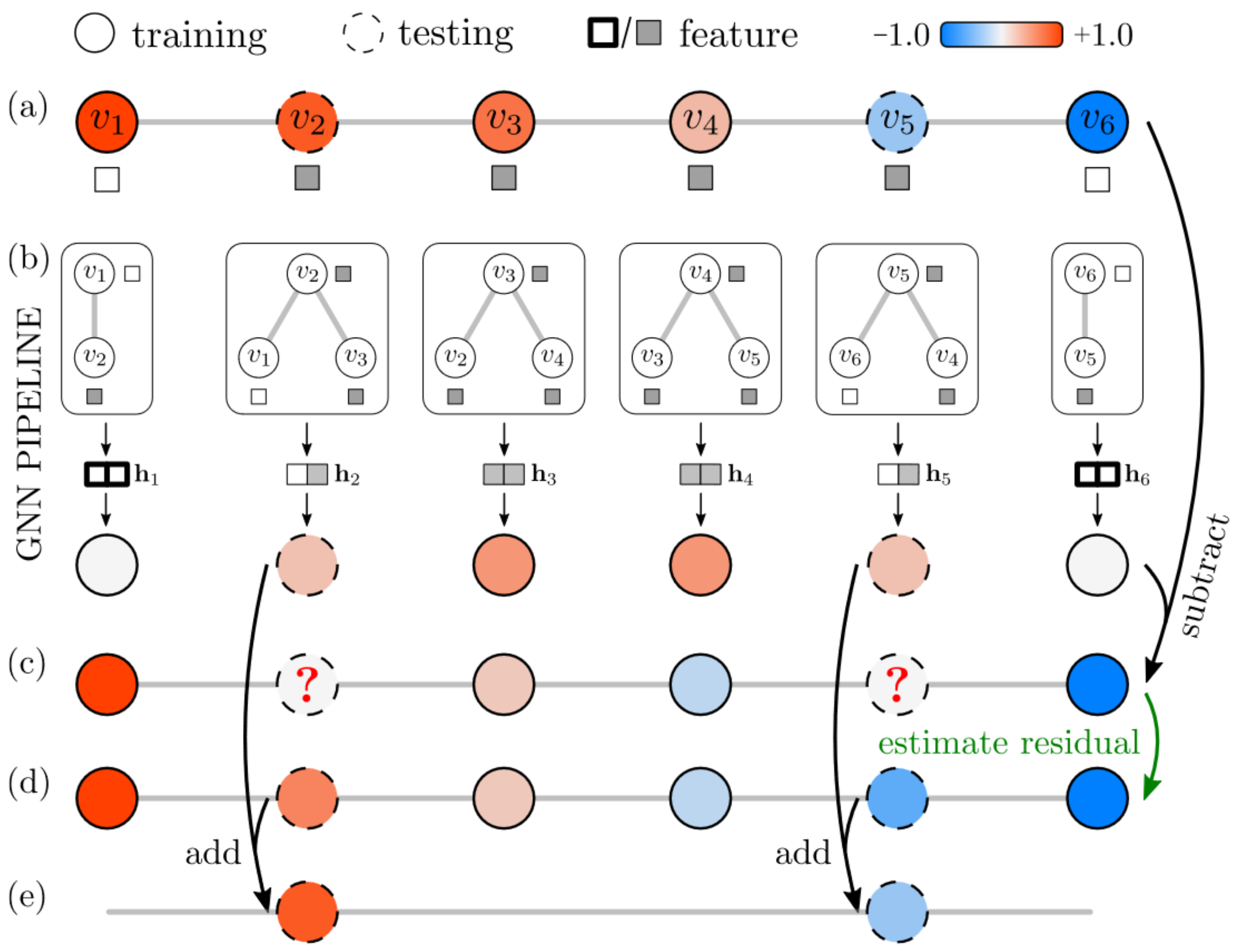
1. LP on residuals for improving GNN regression algorithms.

Solution: Residual Propagation

1. Base prediction.
2. Residual Cal. on labeled nodes.
3. Residual propagation.
4. Final prediction

   = smoothed residual + base prediction

   (= true value on labeled nodes)



—— *Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.*

1. LP on residuals for improving GNN regression algorithms.

Table 1: Transductive learning accuracy of our C-GNN and LP-GNN models compared to competing baselines. The best accuracy is in green. Our C-GNN outperforms GNN on all datasets, often by a substantial margin. Even C-MLP, which does not use neighbor features, outperforms GNN in many cases, highlighting the importance of label correlation. LP, LP-MLP and LP-GNN assume positive label correlation among neighboring vertices and perform poorly for datasets where most edges encode negative interactions, as highlighted in orange. We also report the learned $\{\alpha_i\}$ values from C-GNN.

Dataset	n	m	LP	MLP	LP-MLP	C-MLP	GNN	LP-GNN	C-GNN	$\{\alpha_i\}$
Ising(+)	1.2K	2.4K	**0.76** ± 0.02	0.68 ± 0.03	**0.76** ± 0.02	**0.76** ± 0.02	0.67 ± 0.04	**0.76** ± 0.02	**0.76** ± 0.02	+0.89
Ising(-)	1.2K	2.4K	**0.30** ± 0.03	0.47 ± 0.02	**0.30** ± 0.03	**0.77** ± 0.03	0.47 ± 0.03	**0.30** ± 0.03	**0.77** ± 0.03	−0.93
income	3.2K	12.7K	0.54 ± 0.04	0.64 ± 0.03	0.73 ± 0.03	0.74 ± 0.03	0.75 ± 0.03	**0.81** ± 0.03	**0.81** ± 0.02	+0.92
education	3.2K	12.7K	0.36 ± 0.05	0.67 ± 0.03	0.71 ± 0.02	**0.72** ± 0.02	0.70 ± 0.02	**0.72** ± 0.03	**0.72** ± 0.03	+0.78
unemployment	3.2K	12.7K	0.70 ± 0.03	0.43 ± 0.05	0.69 ± 0.04	0.77 ± 0.03	0.55 ± 0.04	0.75 ± 0.05	**0.78** ± 0.03	+0.99
election	3.2K	12.7K	0.58 ± 0.02	0.37 ± 0.02	0.61 ± 0.03	0.63 ± 0.03	0.51 ± 0.04	**0.69** ± 0.03	**0.69** ± 0.03	+0.95
Anaheim	914	3.8K	0.49 ± 0.08	0.75 ± 0.02	0.81 ± 0.04	**0.82** ± 0.03	0.76 ± 0.03	0.81 ± 0.04	**0.82** ± 0.03	+0.95, +0.17
Chicago	2.2K	15.1K	0.59 ± 0.05	0.60 ± 0.05	0.65 ± 0.06	0.65 ± 0.05	0.68 ± 0.04	**0.72** ± 0.04	0.71 ± 0.04	+0.85, +0.68
sexual	1.9K	2.1K	**0.37** ± 0.06	0.68 ± 0.02	**0.64** ± 0.03	0.83 ± 0.03	0.88 ± 0.02	**0.86** ± 0.02	**0.93** ± 0.01	−0.98
Twitch-PT	1.9K	31.3K	0.00 ± 0.04	0.61 ± 0.03	0.60 ± 0.04	0.66 ± 0.03	0.69 ± 0.03	0.69 ± 0.03	**0.74** ± 0.03	+0.99

—— *Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.*

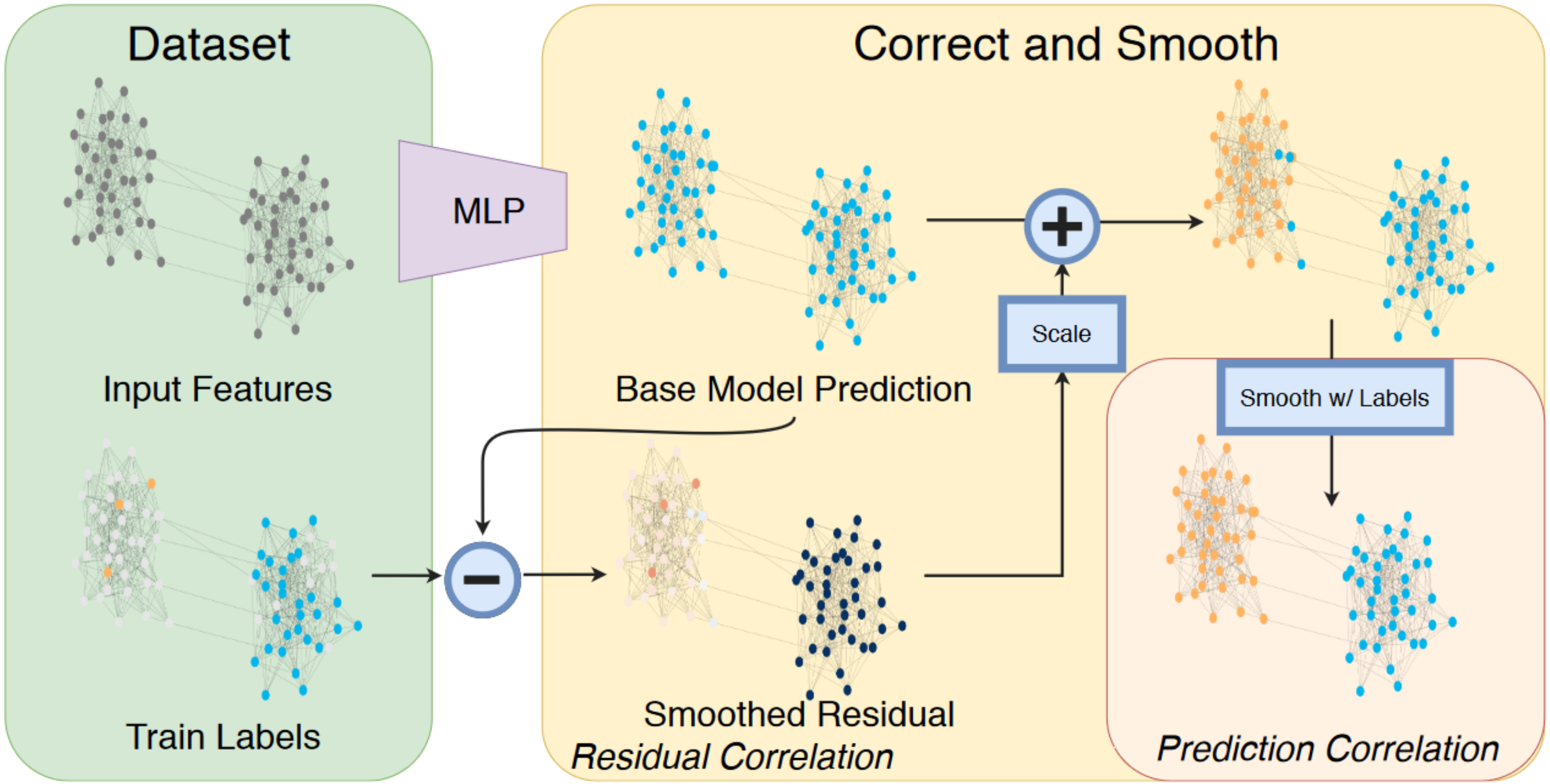
- **1. LP on residuals.**

—— Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.

- **2. Using LP on residuals but removing GNNs altogether.**

—— Combining Label Propagation and Simple Models Out-performs Graph Neural Networks. Huang et al., ICLR 2021.

2. Using LP but removing GNNs altogether for good classification performance.



—— *Combining Label Propagation and Simple Models Out-performs Graph Neural Networks. Huang et al., ICLR 2021.*

• Leaderboard for ogbn-products.

——https://ogb.stanford.edu/docs/leader_nodeprop/

- **Amazon product co-purchasing network:**
  - Undirected and unweighted;
  - Node: products sold in Amazon (# 2,449,029);
  - Edge: indicates the products are purchased together (# 61,859,140);
  - Node features: the product descriptions (PCA).
- **Prediction task:**
  - Category of a product (# 47).
- **Methods related to C&S (up to now):**
  - **Top 5:** 4/5 90%
  - **Top 20:** 9/20 45%

Rank	Method	Ext. data	Test Accuracy	Validation Accuracy	Contact	References	#Params	Hardware	Date
1	GLEM+GIANT+SAGN+SCR	Yes	0.8737 ± 0.0006	0.9400 ± 0.0003	Jianan Zhao (Mila & MSRA Team)	Paper, Code	139,792,525	Tesla V100 (32GB)	Oct 27, 2022
2	**GraDBERT+GIANT & SAGN+SLE+CnS**	Yes	0.8692 ± 0.0007	0.9371 ± 0.0003	Costas Mavromatis (UMN & AWS)	Paper, Code	1,154,654	GeForce RTX 3090 (24GB GPU)	Apr 20, 2023
3	GIANT-XRT+R-SAGN+SCR+C&S	Yes	0.8684 ± 0.0005	0.9365 ± 0.0003	LeeXue (HIT Team)	Paper, Code	1,154,142	TITAN RTX (24GB GPU)	Sep 30, 2022
4	GIANT-XRT+SAGN+SCR+C&S	Yes	0.8680 ± 0.0007	0.9357 ± 0.0004	Yufei He (CogDL Team)	Paper, Code	1,154,654	GeForce RTX 3090 24GB (GPU)	Jun 13, 2022
5	GIANT-XRT+SAGN+MCR+C&S	Yes	0.8673 ± 0.0008	0.9387 ± 0.0002	Yufei He (CogDL Team)	Paper, Code	1,154,654	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
6	GIANT-XRT+SAGN+SCR	Yes	0.8667 ± 0.0009	0.9364 ± 0.0005	Yufei He (CogDL Team)	Paper, Code	1,154,654	GeForce RTX™ 3090 24GB (GPU)	Jun 13, 2022
7	GIANT-XRT+SAGN+MCR	Yes	0.8651 ± 0.0009	0.9389 ± 0.0002	Yufei He (CogDL Team)	Paper, Code	1,154,654	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
8	GIANT-XRT+SAGN+SLE+C&S (use raw text)	Yes	0.8643 ± 0.0020	0.9352 ± 0.0005	Eli Chien (UIUC)	Paper, Code	1,548,382	Tesla T4 (16GB GPU)	Nov 8, 2021
9	GIANT-XRT+SAGN+SLE (use raw text)	Yes	0.8622 ± 0.0022	0.9363 ± 0.0005	Eli Chien (UIUC)	Paper, Code	1,548,382	Tesla T4 (16GB GPU)	Nov 8, 2021
10	GIANT-XRT+GAMLP+MCR	Yes	0.8591 ± 0.0008	0.9402 ± 0.0004	Yufei He (CogDL Team)	Paper, Code	2,144,151	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
11	GAMLP+RLU+SCR+C&S	No	0.8520 ± 0.0008	0.9304 ± 0.0005	Yufei He (CogDL Team)	Paper, Code	3,335,831	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
12	GAMLP+RLU+SCR	No	0.8505 ± 0.0009	0.9292 ± 0.0005	Yufei He (CogDL Team)	Paper, Code	3,335,831	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
13	SAGN+SLE (4 stages)+C&S	No	0.8485 ± 0.0010	0.9302 ± 0.0003	Chuxiong Sun (CTRI)	Paper, Code	2,179,678	Tesla V100 (16GB GPU)	Sep 21, 2021
14	SAGN+SLE (4 stages)	No	0.8468 ± 0.0012	0.9309 ± 0.0007	Chuxiong Sun (CTRI)	Paper, Code	2,179,678	Tesla V100 (16GB GPU)	Sep 21, 2021
15	GAMLP+MCR	No	0.8462 ± 0.0003	0.9319 ± 0.0003	Yufei He (CogDL Team)	Paper, Code	3,335,831	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
16	GAMLP+RLU	No	0.8459 ± 0.0010	0.9324 ± 0.0005	Wentao Zhang (PKU Tencent Joint Lab)	Paper, Code	3,335,831	Tesla V100 (32GB)	Aug 19, 2021
17	Spec-MLP-Wide + C&S	No	0.8451 ± 0.0006	0.9132 ± 0.0010	Huixuan Chi (AML@ByteDance)	Paper, Code	406,063	Tesla V100 (32GB)	Jul 27, 2021
18	SAGN+MCR	No	0.8441 ± 0.0005	0.9325 ± 0.0004	Yufei He (CogDL Team)	Paper, Code	2,179,678	GeForce RTX™ 3090 24GB (GPU)	Dec 8, 2021
19	SAGN+SLE	No	0.8428 ± 0.0014	0.9287 ± 0.0003	Chuxiong Sun	Paper, Code	2,179,678	Tesla V100 (16GB GPU)	Apr 19, 2021
20	MLP + C&S	No	0.8418 ± 0.0007	0.9147 ± 0.0009	Horace He (Cornell)	Paper, Code	96,247	GeForce RTX 2080 (11GB GPU)	Oct 27, 2020

20	**MLP + C&S**	No	0.8418 ± 0.0007	0.9147 ± 0.0009	Horace He (Cornell)	Paper, Code	96,247	GeForce RTX 2080 (11GB GPU)	Oct 27, 2020

...

25	**Linear + C&S**	No	0.8301 ± 0.0001	0.9134 ± 0.0001	Horace He (Cornell)	Paper, Code	10,763	GeForce RTX 2080 (11GB GPU)	Oct 27, 2020
26	**UniMP**	No	0.8256 ± 0.0031	0.9308 ± 0.0017	Yunsheng Shi (PGL team)	Paper, Code	1,475,605	Tesla V100 (32GB)	Sep 8, 2020

...

GAT (Neighbor sampling or Cluster) (+ C&S)

...

GraphSAGE (+ C&S)

...

49	Label Propagation	No	0.7434 ± 0.0000	0.9091 ± 0.0000	Horace He (Cornell)	Paper, Code	0	GeForce RTX 2080 (11GB GPU)	Oct 3, 2020

...

Node2vec

...

54	MLP	No	0.6106 ± 0.0008	0.7554 ± 0.0014	Matthias Fey – OGB team	Paper, Code	103,727	GeForce RTX 2080 (11GB GPU)	Jun 10, 2020

- **1. LP on residuals.**

—— Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.

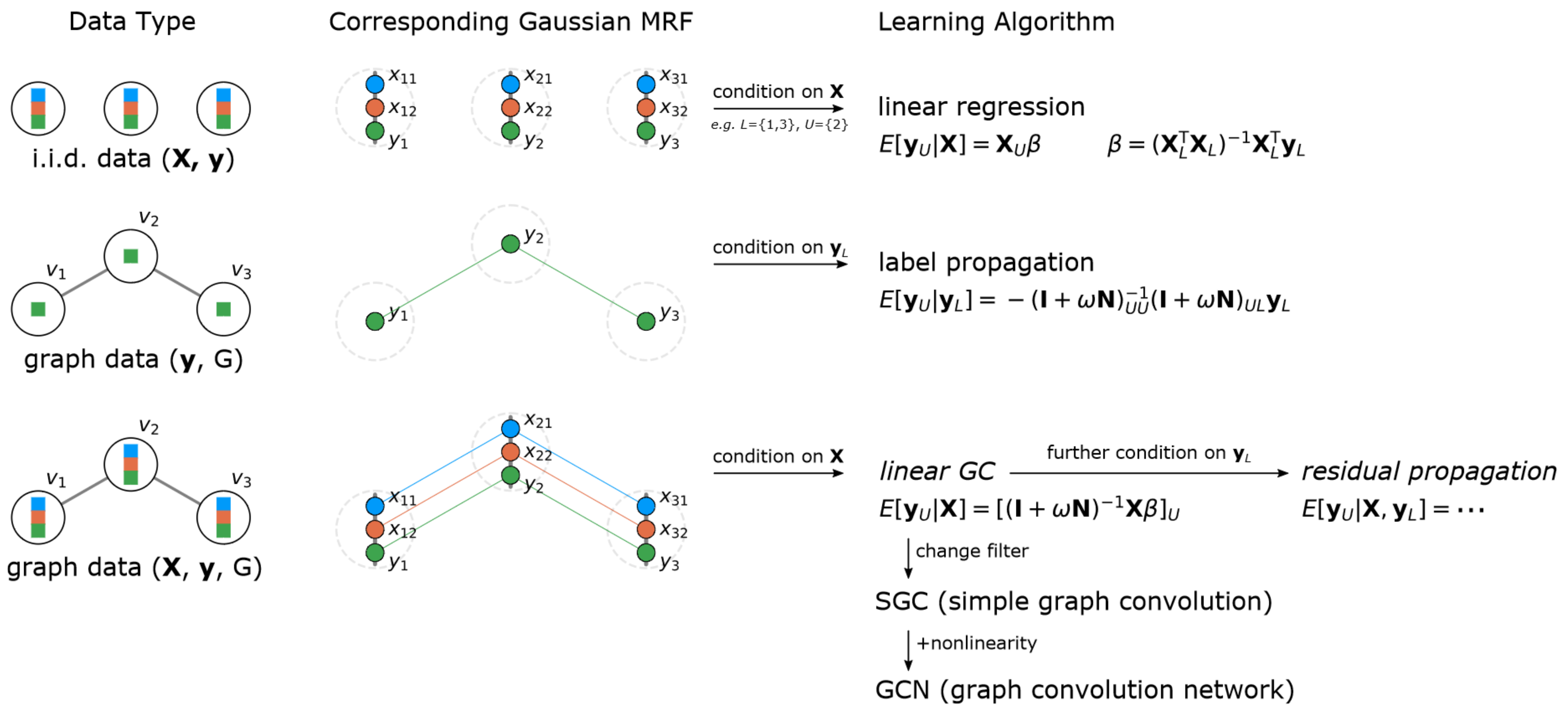
- **2. Using LP on residuals but removing GNNs altogether.**

—— Combining Label Propagation and Simple Models Out-performs Graph Neural Networks. Huang et al., ICLR 2021.

- **3. Statistical framework that unifies LP and GNN ideas.**

—— A Unifying Generative Model for Graph Learning Algorithms: Label Propagation, Graph Convolutions, and Combinations. Jia & Benson, SIAM Journal on Mathematics of Data Science 2022.

3. Statistical framework that unifies LP and GNN ideas.



—— A Unifying Generative Model for Graph Learning Algorithms: Label Propagation, Graph Convolutions, and Combinations. Jia & Benson, SIAM Journal on Mathematics of Data Science 2022.

3. Statistical framework that unifies LP and GNN ideas.

- Input:
  - Graph $G = (V,\ E,\ X,\ Y)$.
  - Node Feature $X \in \mathbb{R}^{|V|\times d}$.
  - Known Node Label $y_L \in \{0,\ 1\}^{|V_L|}$, ($V_L \in V$ The labeled node set.)

- Output:
  - Unknown Node Label $y_U \in \{0,\ 1\}^{|V_U|}$, ($V_U = V \setminus V_L \in V$ the unlabeled node set.)

- **Solution:**
  - $y_U = \mathbb{E}[y_U | G, X, y_L]$

—— *A Unifying Generative Model for Graph Learning Algorithms: Label Propagation, Graph Convolutions, and Combinations. Jia & Benson, SIAM Journal on Mathematics of Data Science 2022.*

3. Statistical framework that unifies LP and GNN ideas.

Linear Graph Conv. (LGC)	Simplified Graph Conv. (SGC)	Graph Conv. Network (GCN)
$(1-\alpha)(I+\alpha S+\alpha^2 S^2+\cdots)X\beta$	$\tilde{S}^k X\beta$	$\sigma(\tilde{S}\ldots\sigma(\tilde{S}XW^1)\ldots W^k)\beta$
$S=D^{-1/2}AD^{-1/2}$	$\tilde{S}=(D+I)^{-1/2}(A+I)(D+I)^{-1/2}$	$\tilde{S}=(D+I)^{-1/2}(A+I)(D+I)^{-1/2}$

Dataset	Outcome	LP	LR	LGC (α)	SGC (K)	GCN (K)	LGC/RP	SGC/RP	GCN/RP
U.S.	income	0.40	0.63	0.66 (0.46)	0.51 (1.0)	0.53 (1.3)	**0.69**	0.55	0.55
	education	0.31	**0.71**	**0.71** (0.00)	0.43 (1.0)	0.47 (1.0)	**0.71**	0.46	0.48
	unemployment	0.47	0.34	0.39 (0.59)	0.32 (1.3)	0.45 (2.5)	**0.54**	0.52	0.53
	election	0.52	0.42	0.49 (0.68)	0.43 (1.1)	0.52 (2.1)	**0.64**	0.61	0.61
CDC	airT	0.95	0.85	0.86 (0.78)	0.86 (2.6)	0.95 (3.0)	0.96	**0.97**	**0.97**
	landT	0.89	0.81	0.81 (0.09)	0.79 (1.0)	0.91 (2.4)	0.90	**0.93**	**0.93**
	precipitation	0.89	0.59	0.61 (0.93)	0.61 (2.3)	0.79 (3.0)	0.89	**0.90**	**0.90**
	sunlight	0.96	0.75	0.81 (0.97)	0.80 (3.0)	0.90 (3.0)	0.96	**0.97**	**0.97**
	pm2.5	0.96	0.21	0.27 (0.99)	0.23 (2.7)	0.78 (3.0)	0.96	0.96	**0.97**
London	income	0.46	**0.85**	**0.85** (0.00)	0.64 (1.0)	0.63 (1.0)	**0.85**	0.65	0.64
	education	0.65	0.81	0.83 (0.40)	0.74 (1.6)	0.79 (1.4)	**0.86**	0.77	0.79
	age	0.65	0.73	0.73 (0.17)	0.66 (1.2)	0.70 (1.7)	**0.75**	0.72	0.72
	election	0.67	0.73	0.81 (0.74)	0.74 (2.0)	0.76 (2.1)	**0.85**	0.78	0.78
Twitch	days	0.08	0.58	0.59 (0.67)	0.22 (1.4)	0.26 (1.7)	**0.60**	0.23	0.26

h_0	LP (α)	LR	LGC (α)	SGC (K)	GCN (K)	LGC/RP (α)	SGC/RP (K,α)	GCN/RP (K,α)
1	0.19 (0.79)	0.68	0.70 (0.28)	0.37 (1.8)	0.34 (1.7)	**0.73** (0.29)	0.40 (1.8, 0.21)	0.37 (1.7, 0.21)
10	0.43 (0.95)	0.48	0.58 (0.57)	0.45 (2.1)	0.45 (2.0)	**0.68** (0.56)	0.56 (2.1, 0.46)	0.54 (2.0, 0.43)
100	0.59 (0.99)	0.24	0.42 (0.85)	0.38 (2.3)	0.45 (2.5)	**0.64** (0.85)	0.63 (2.3, 0.81)	0.62 (2.5, 0.79)

- GCN more expressive but prone to overfitting.
- More homophily $\longrightarrow$ larger K, α
- Adding residual prop never hurts!
- GCN better with more homophily?
  "memorizing" neighborhood features (zero training error)
  \+ smoothness in data $\longrightarrow$ better out-of-sample prediction

—— *A Unifying Generative Model for Graph Learning Algorithms: Label Propagation, Graph Convolutions, and Combinations. Jia & Benson, SIAM Journal on Mathematics of Data Science 2022.*
—— *Graph Signal Processing: Overview, Challenges and Applications. Ortega & etc., Proceedings of the IEEE 2018.*

5. Rethinking about Future

LP is a powerful tool:

- Applied to residuals (correlated errors), features (smoothing / de-noising), final predictions (more smoothness)

- Linear models are often superior to nonlinear ones (GNNs) in TNC.

Few-shot on Graph:

- Even nothing holds meaning.
- Even a lack can suffice.

Feature? Label?

- Feature: attribute we use.
- Label: attribute we predict.

Residual?

- Explicitness: What we need to correct.
- Implicitness: **Influence** we have modeled.

Transductive? Inductive?

- Learning the influence of the lack component in the Few-shot Problem.
- Latent law.

Reference

[1] Deep Long-Tailed Learning: A Survey, Zhang & etc., PAMI 2023.

[2] A Survey of Imbalanced Learning on Graphs: Problems, Techniques, and Future Directions, Liu & etc., arXiv 2023.

[3] A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection, Jin & etc., arXiv 2023.

[4] Learning to Propagate Labels: Transductive Propagation Network for Few-shot Learning. Liu & etc., ICLR 2019.

[5] Inductive Representation Learning on Large Graphs. Hamilton & etc., NIPS 2017.

[6] Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. Zhu & etc., ICML 2003.

[7] Predict then Propagate: Graph Neural Networks meet Personalized PageRank. Gasteiger & etc., ICLR 2019.

[8] The Anatomy of a Large-Scale Hypertextual Web Search Engine. Brin & Page, Computer networks and ISDN systems, 1998.

[9] Unifying Graph Convolutional Neural Networks and Label Propagation. Wang & Leskovec., ariXiv 2020.

[10] Graph Attention Networks. Velickovic & etc., ICLR 2018.

[11] Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. Shi & etc., IJCAI 2021.

[12] Training Sparse Neural Networks. Srinivas& etc., CVPR Workshops, 2017.

[13] Residual Correlation in Graph Neural Network Regression. Jia & Benson, KDD 2020.

[14] Combining Label Propagation and Simple Models Out-performs Graph Neural Networks. Huang et al., ICLR 2021.

[15] A Unifying Generative Model for Graph Learning Algorithms: Label Propagation, Graph Convolutions, and Combinations. Jia & Benson, SIAM Journal on Mathematics of Data Science 2022.

[16] Graph Signal Processing: Overview, Challenges and Applications. Ortega & etc., Proceedings of the IEEE 2018.

[17] Stanford CS224W: Label propagation on graph, Leskovec, http://web.stanford.edu/class/cs224w/slides/08-message.pdf.

[18] Label Propagation and Graph Neural Networks, Benson & etc., https://www.cs.cornell.edu/~arb/slides/2021-03-12-northeastern.pdf.

Thx for Q&A   :)