Label Efficient Semi-Supervised Learning via Graph Filtering

Qimai Li, Xiao-Ming Wu, Han Liu,
Xiaotong Zhang, Zhichao Guan

Slides by Qimai LI and Lu FAN

Learning with Few Labels

black-and-white striped coats

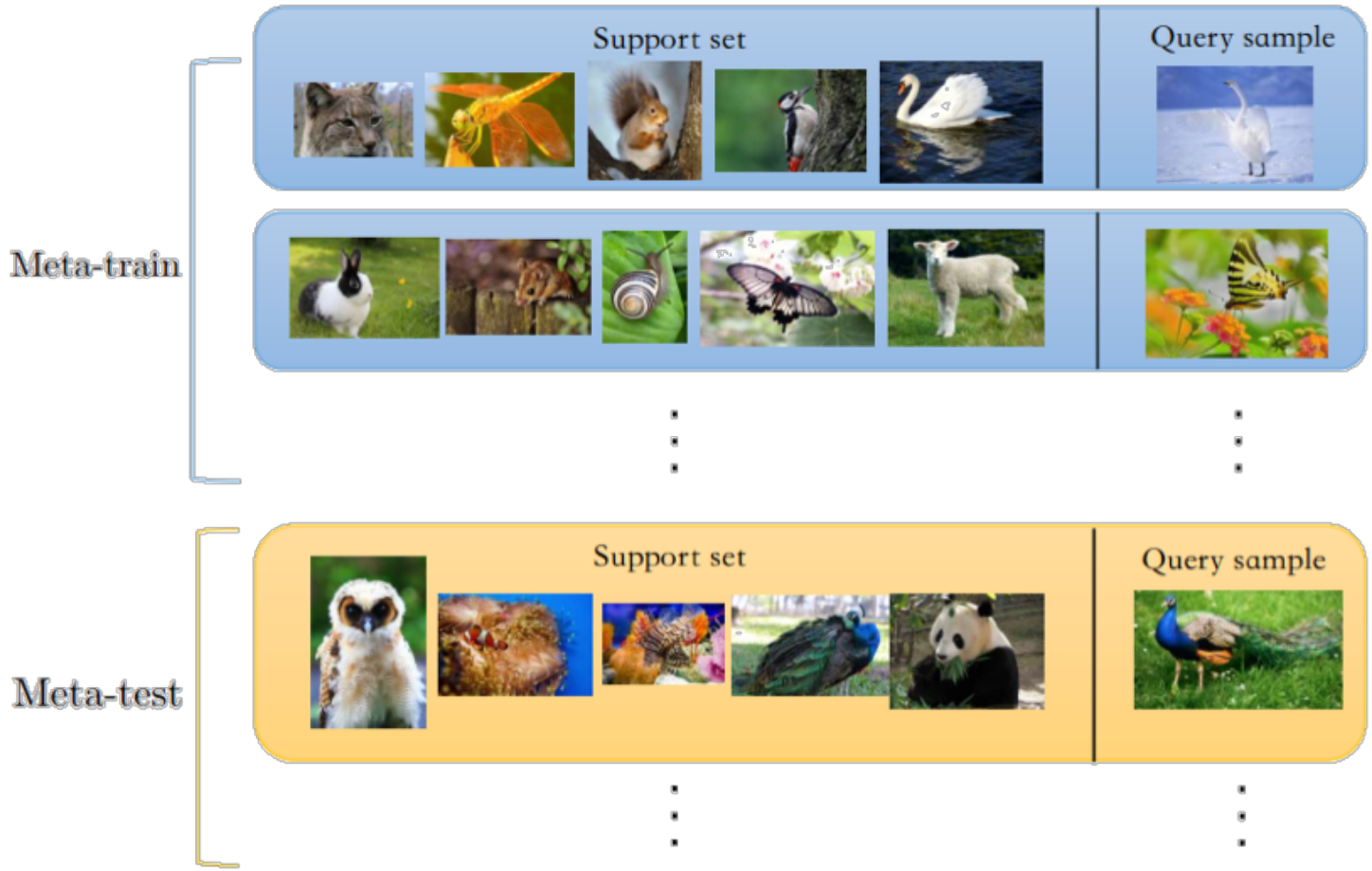
Know what Segway is once we see the photo

Hard for conventional machine learning frameworks, which rely heavily on large amount of data

Require the ability to rapidly discover and represent new concepts with few observations

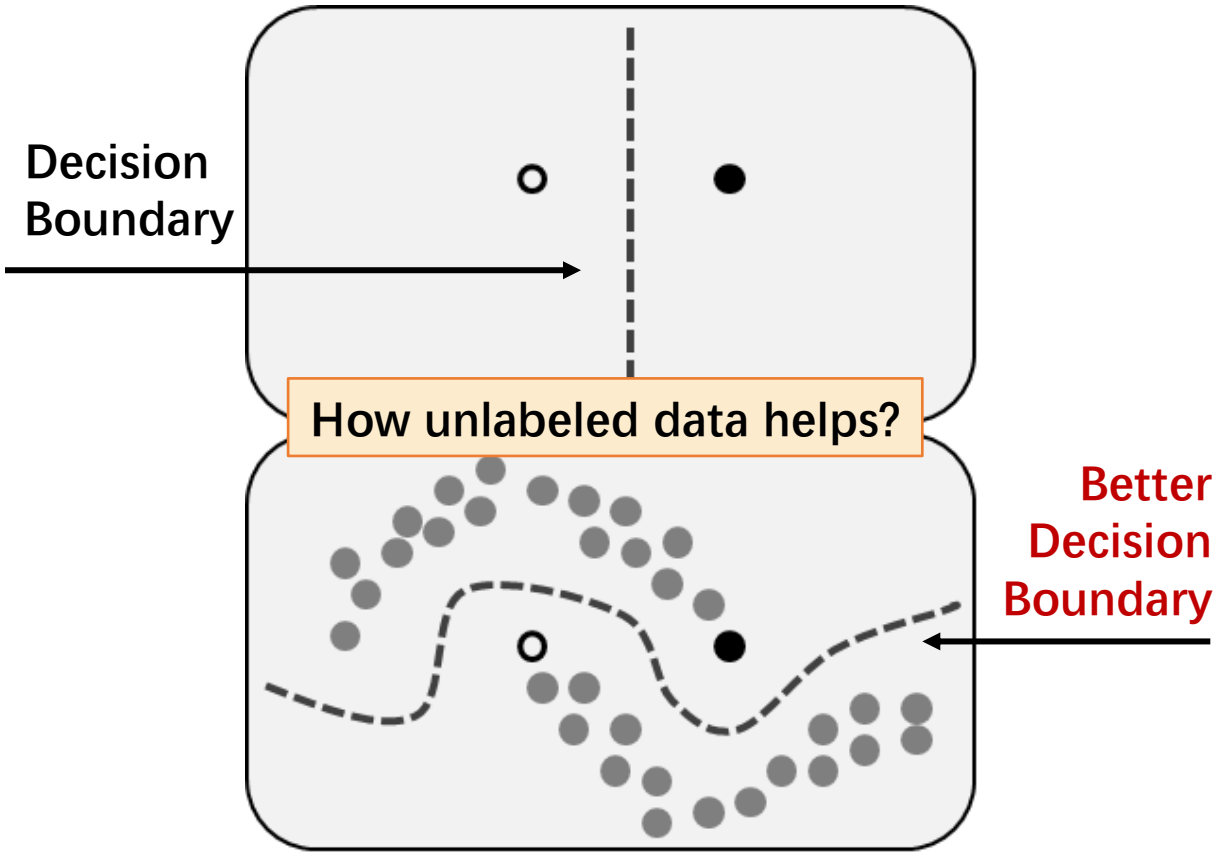
J. Shu, X. Zongben, and M. Deyu. Small Sample Learning in Big Data Era., 2018.
https://en.wikipedia.org/wiki/Zebra; https://en.wikipedia.org/wiki/Zebra; https://www.cpadventure.ie/product/segway-tour/

Few-Shot Learning



Training and testing process of few-shot learning.

Semi-Supervised Learning

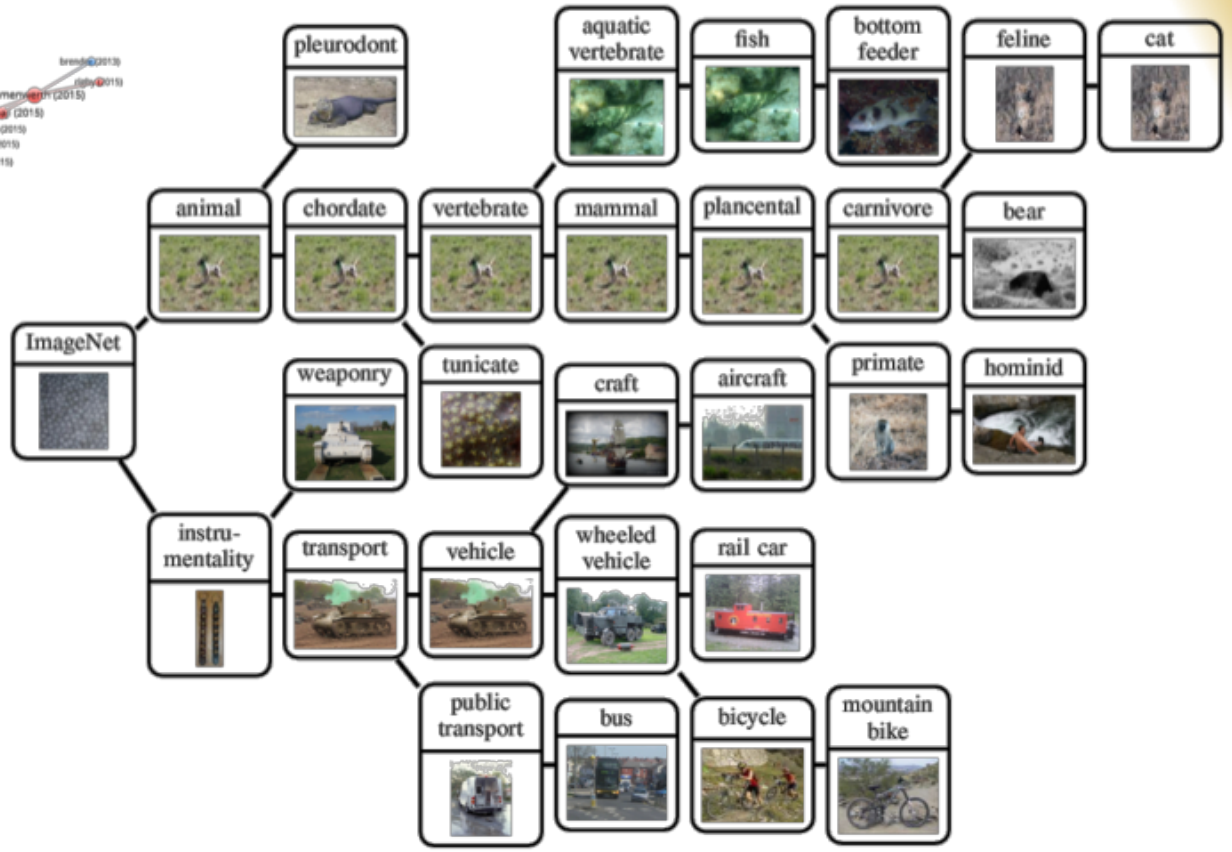


Unlabeled data can significantly improve learning performance

Represent data in a graph and exploit the graph structures

Image via "https://en.wikipedia.org/wiki/Semi-supervised_learning"

Citation Network



ImageNet

Graph-Structured Data

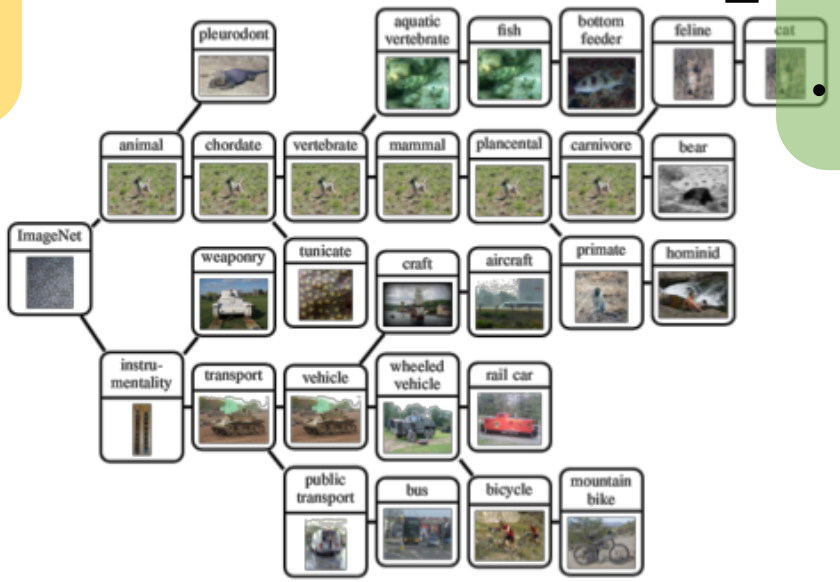
https://scholarlykitchen.sspnet.org/2016/09/26/visualizing-citation-cartels/
http://groups.inf.ed.ac.uk/calvin/imagenet/prototypes.html

Graph-Based Semi-Supervised Learning

Integrate Graph and Feature Information

- Document content
- Title
- ...

Citation Network

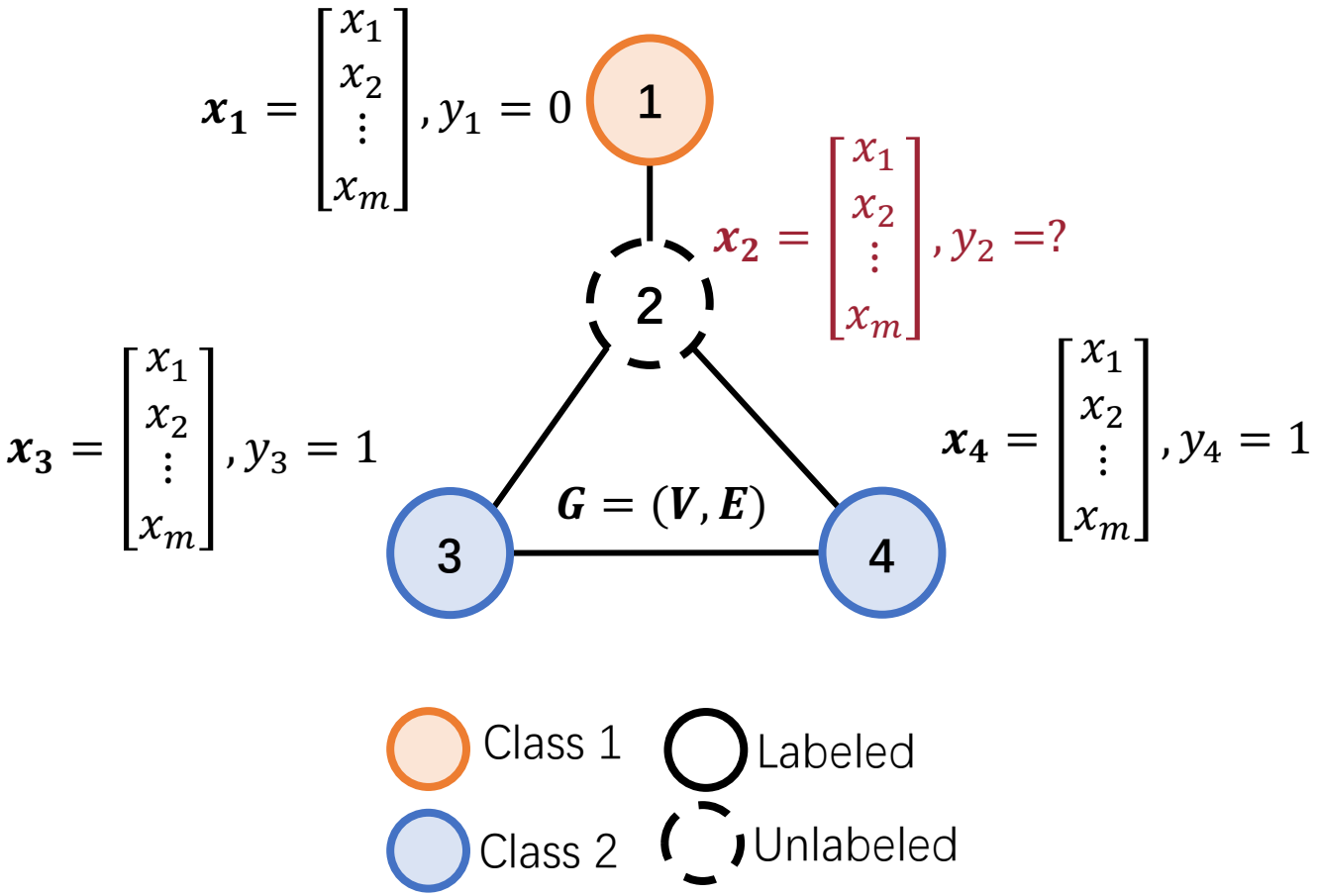


- Image
- Textual category description
- ...

ImageNet

https://scholarlykitchen.sspnet.org/2016/09/26/visualizing-citation-cartels/
http://groups.inf.ed.ac.uk/calvin/imagenet/prototypes.html

Graph-Based Semi-Supervised Learning



Input:

$G = (V, E)$

X, feature matrix

Y, label matrix

Output:

Labels of unlabeled nodes

Graph-Based Semi-Supervised Learning
Current Progress

Non-GCNN Based	LP (Zhu et al., 2003)	**GCNN Based**	Cheyshev (Defferrard et al., 2016)
	ManiReg (Belkin et al., 2006)		MoNet (Monti et al., 2016)
	ICA (Sen et al., 2008)		GCN (Kipf & Welling, 2017)
	SemiEmb (Weston et al., 2012)		GraphSAGE (Hamilton et al., 2017)
	Plantoid (Yang et al., 2016)		GAT (Velickovic et al., 2018)
	...		...

Current Limitations and Our Contributions

Limitation	Contribution
Limited theoretical understanding	Provide **new insights** into graph convolutional networks (GCN) and show its close connection with label propagation (LP) from a low-pass graph filtering perspective.
Not label efficient (require many labels for model training/validation)	Propose **generalized LP (GLP)** **and** **improved GCN (IGCN)** methods to **reduce model complexity** and **tackle label insufficiency** in semi-supervised learning.
High model complexity	Demonstrate the high efficacy of the proposed methods on various tasks including **text and entity classification** and **zero-shot image recognition**.

Graph Signal Processing (GSP)
Graph Signals

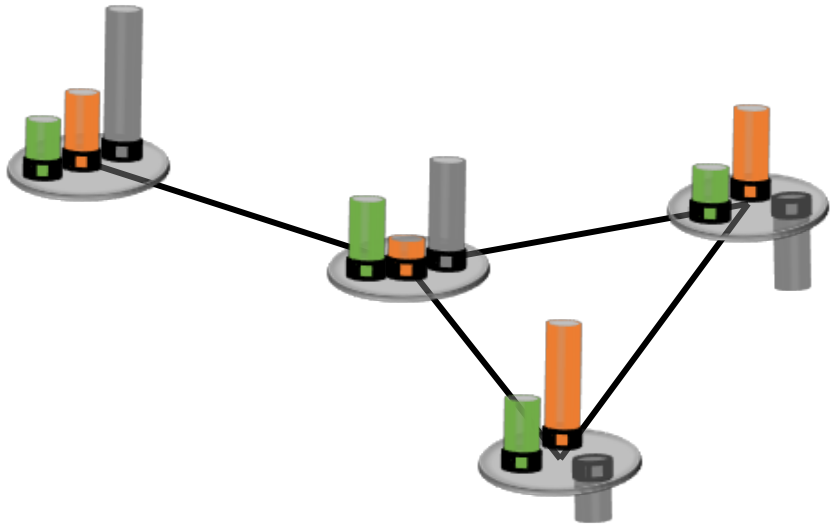
A **graph signal** is a real-valued function defined on vertex set V:

$$f: V \to \mathbb{R}$$

In vector form:

$$f = \left(f(v_1), \ldots, f(v_n)\right)^T$$

Examples: columns of feature matrix (X) and label matrix (Y)



Graph Signal Processing (GSP)
Fourier Basis

Graph Laplacian: $L = D - W$,

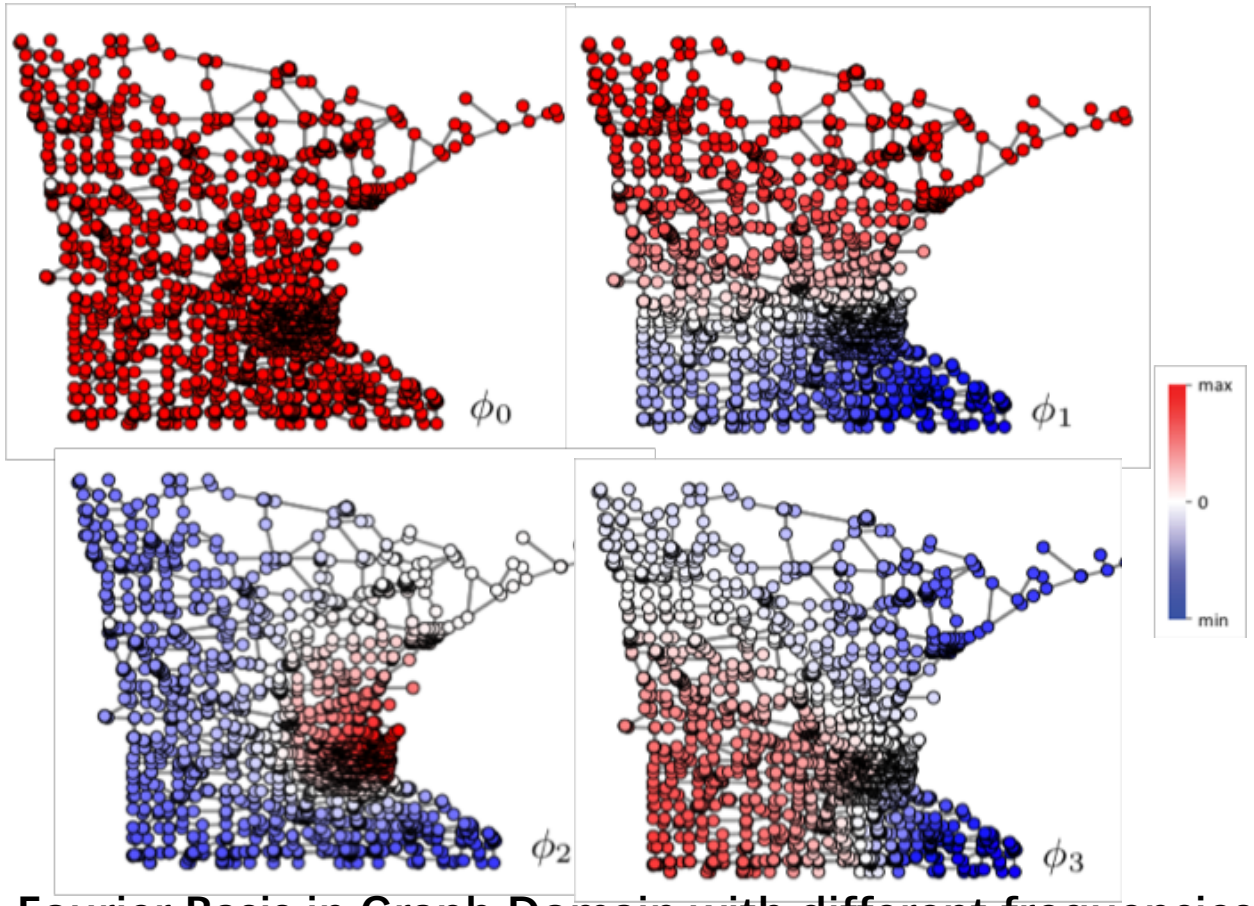
where $D = \mathrm{diag}(d_i)$ and $d_i = \sum_j w_{ij}$

Eigenvectors of L serve as **Fourier basis**, **eigenvalues** of L are interpretaed as **frequency**:

$$L = \Phi \Lambda \Phi^{-1}$$

$$f = \Phi \boldsymbol{c}$$

where $\Phi = (\phi_1, \dots, \phi_n)$, $\Lambda = diag(\lambda_1, \dots, \lambda_n)$, $\boldsymbol{c} = (c_1, \dots, c_n)^T$ and c_i is the coefficient of ϕ_i.



Fourier Basis in Graph Domain with different frequencies

https://arxiv.org/pdf/1611.08097.pdf

Graph Signal Processing (GSP)
Convolutional Filters

Filtering process:

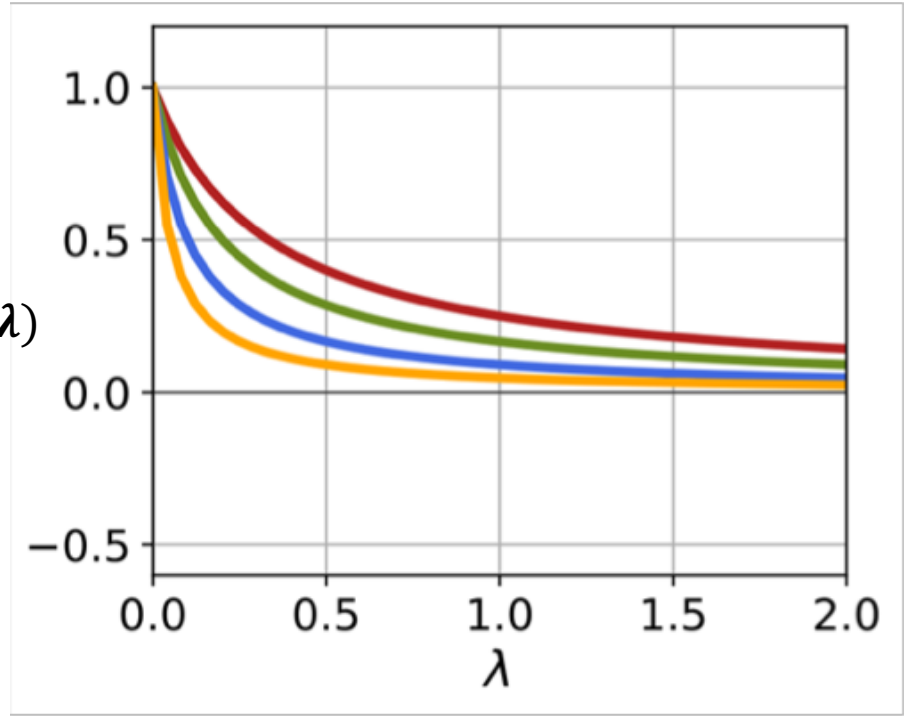
$$\bar{f} = Gf$$

Convolutional filters:

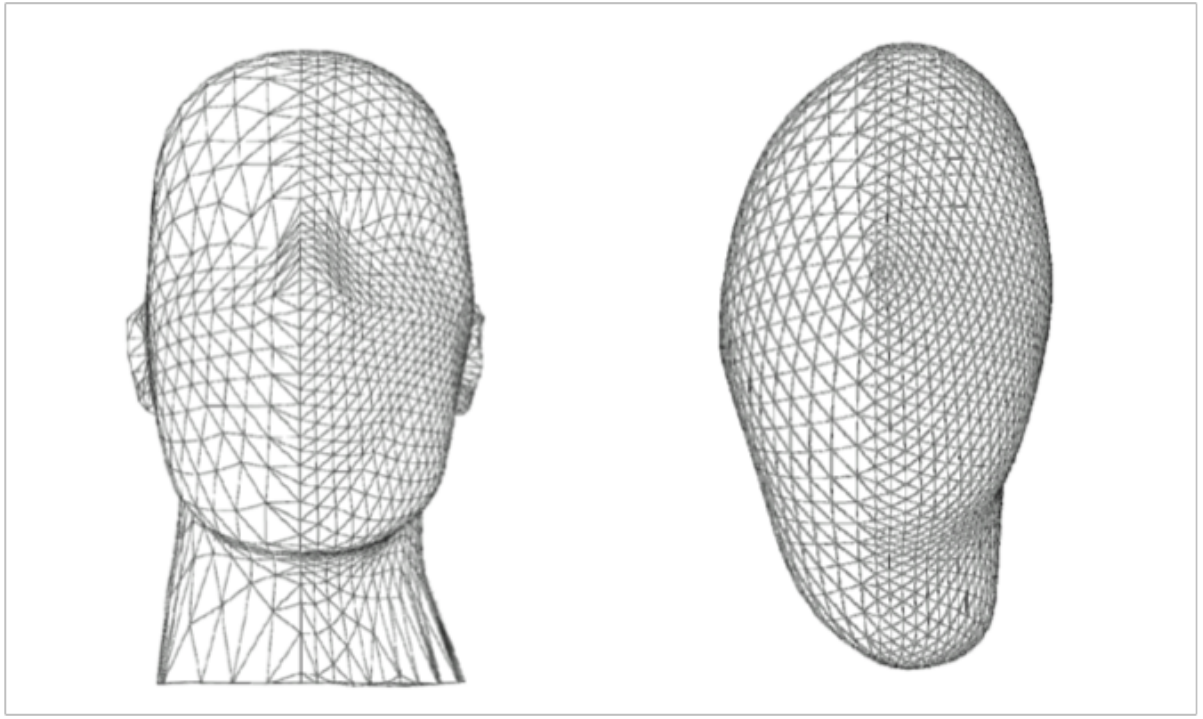
$$G = \Phi \begin{bmatrix} p(\lambda_1) & & \\ & \ddots & \\ & & p(\lambda_n) \end{bmatrix} \Phi^{-1} \triangleq \Phi p(\Lambda) \Phi^{-1} \triangleq p(L)$$

$p(\cdot)$ is a real-valued function, called the **frequency response function** of G, $(\Lambda) = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$.

Graph Signal Processing(GSP)
Low-Pass Filters



Examples of frequency response of low-pass filters.



Before and after low-pass filtering. (Take vertex coordinates as signals)

From Desbrun et al., Siggraph 1999

Label Propagation (LP)

- LP is one of the most popular graph-based SSL methods.
- Assumption: closely related nodes tend to share the same label.

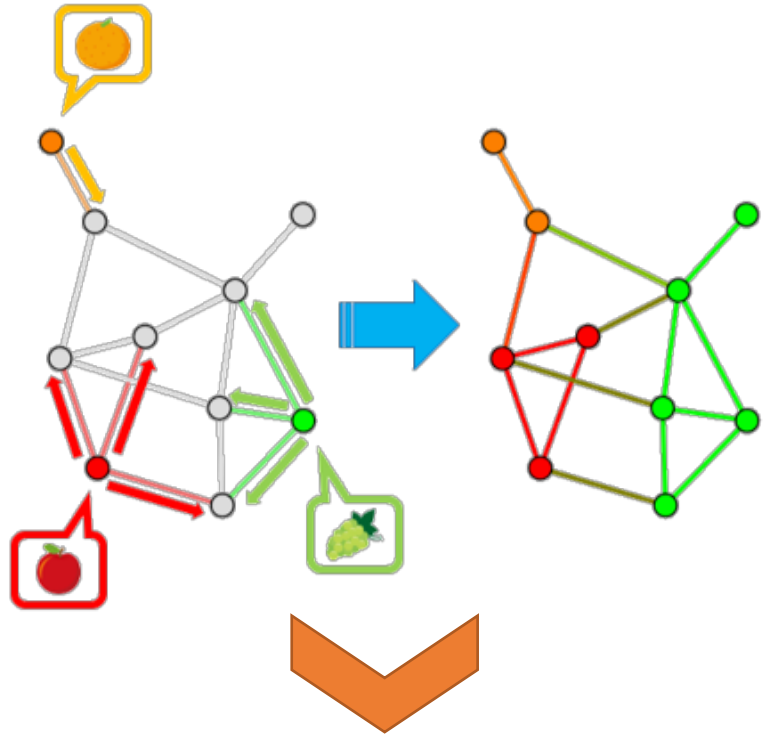
Objective Function:

$$Z = \arg\min_{Z} \{ \underbrace{||Z - Y||_2^2}_{\text{fitting error}} + \alpha \underbrace{\mathrm{Tr}(Z^\top L Z)}_{\text{regularization}} \},$$

Closed-Form Solution:

$$\boldsymbol{Z = (I + \alpha L)^{-1} Y}$$

Here $\boldsymbol{L = D - W}$ with $\boldsymbol{d_i = \sum_j w_{ij}}$ and $\boldsymbol{D = \mathrm{diag}(d_i)}$



Only exploit graph structures
Unable to jointly model
graph structures and data features

Revisiting Label Propagation (LP) in the Context of Graph Signal Processing

The closed-form solution:

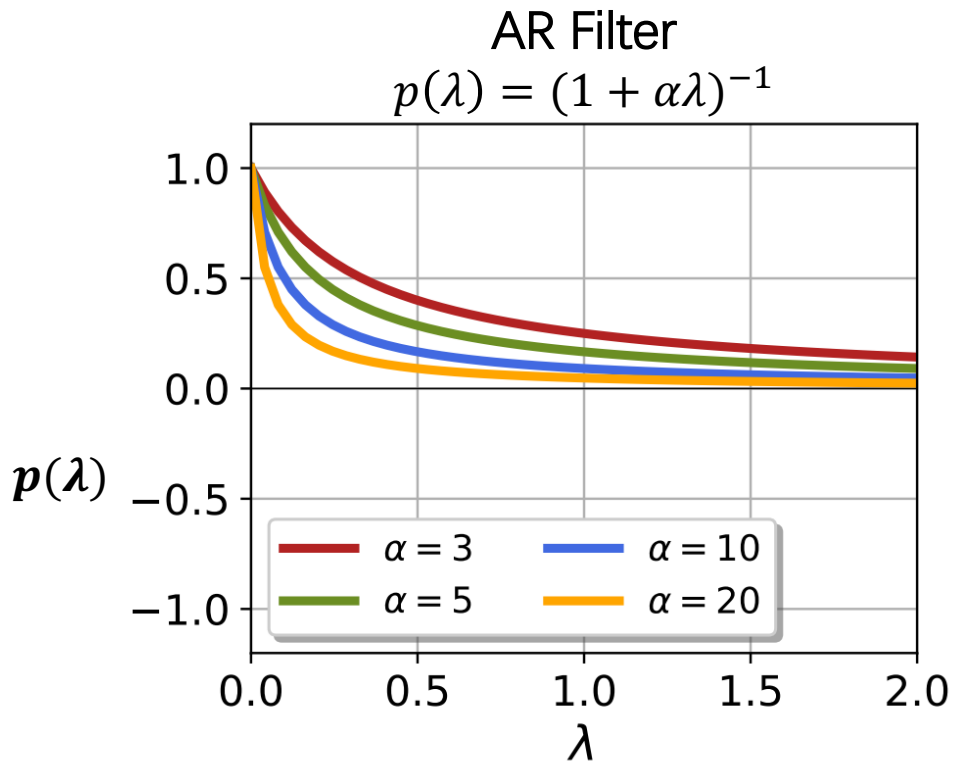
label matrix

$$Z = \underbrace{(I + \alpha L)^{-1}}_{\text{low-pass filter}} Y$$

Three components of LP:

1) The graph signals, Y
2) The low-pass filter $(I + \alpha L)^{-1}$
3) The classifier:

$$l_i = argmax_j \, Z_{ij}$$



Generalized Label Propagation (GLP)

Extend Signals, Filters, and Classifiers

LP: $Z = (I + \alpha L)^{-1} Y$

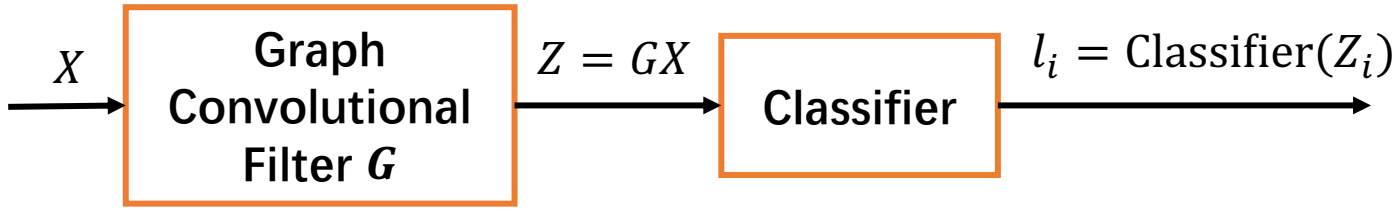
Label matrix

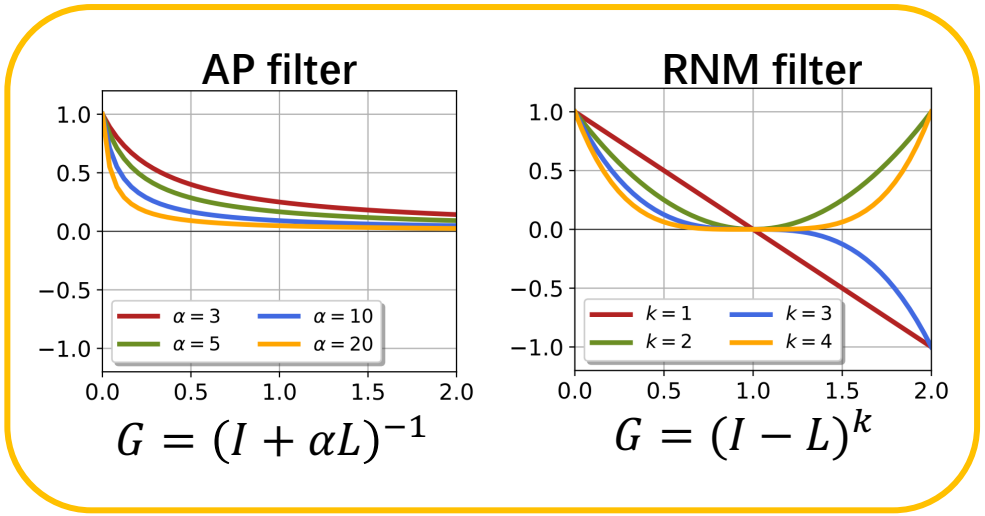
fixed filter

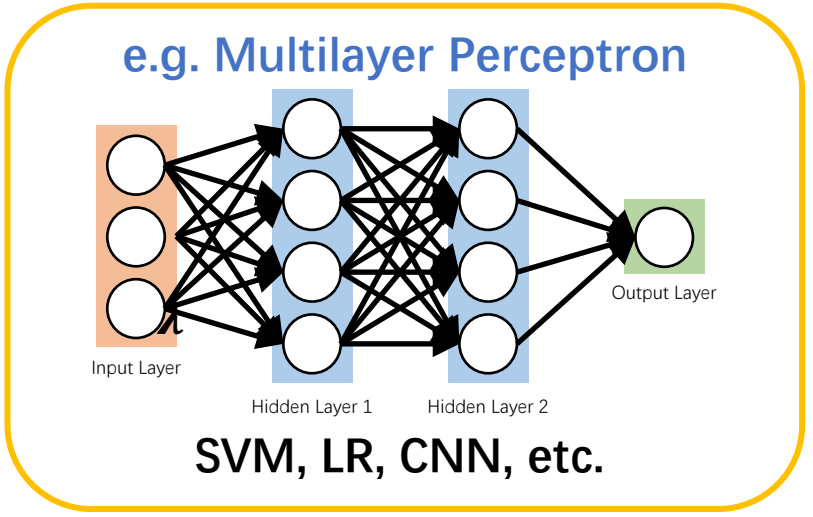
GLP: $Z = G X$

features

general filter







Benefits of GLP

1) Effectively **combines vertex features with graph structures**, whereas LP overlooks rich information in vertex features.

2) A flexible framework that allows **adopting efficient low-pass filters and task-specific classifiers**.

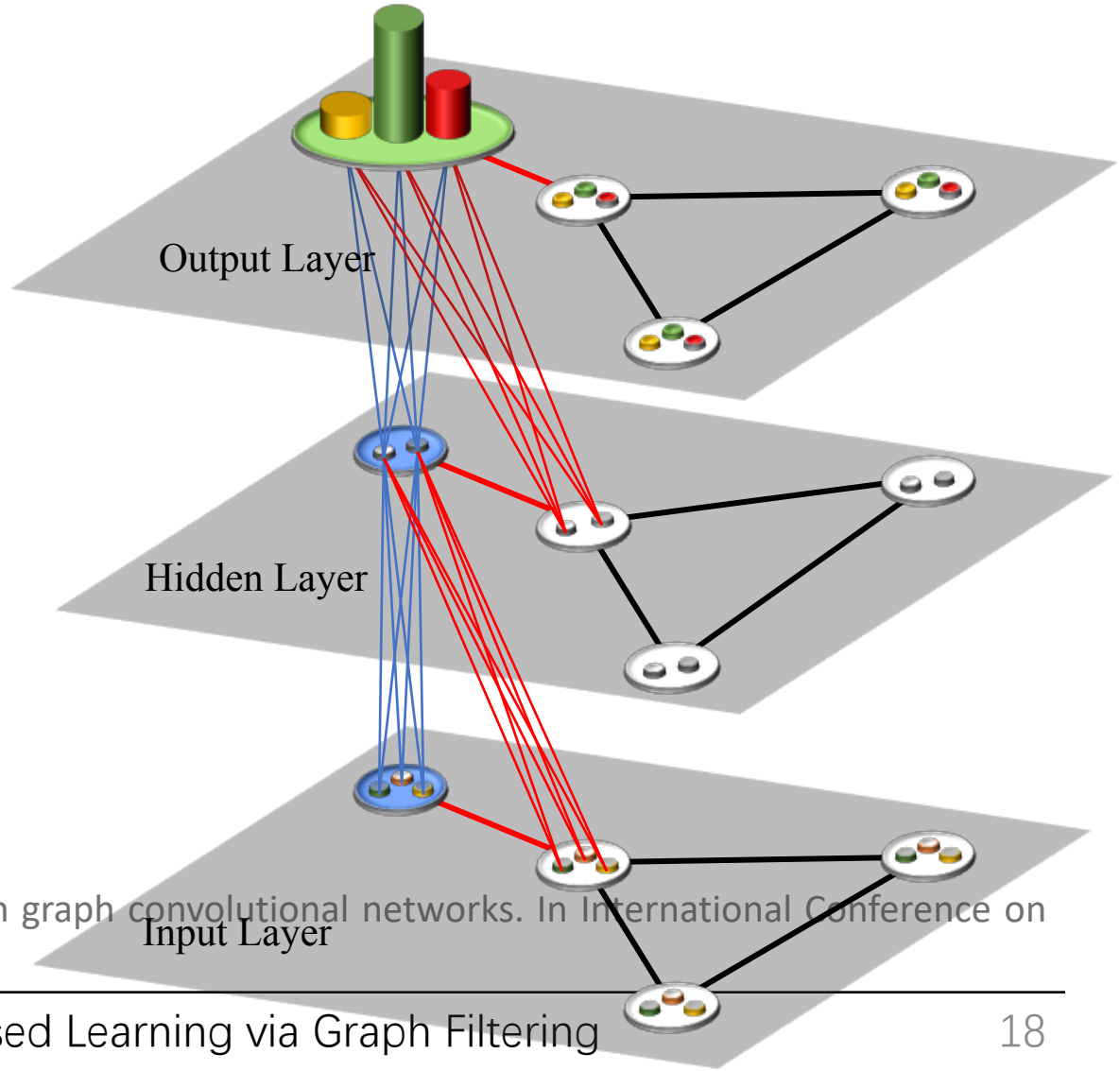
3) Significantly **improve training efficiency**.

Graph Convolutional Network (GCN)

$$Z = \text{softmax}\left(\widetilde{W}_s ReLU\left(\widetilde{W}_s X \Theta^{(0)}\right)\Theta^{(1)}\right)$$

$\widetilde{W}_s$ is the Normalized Adjacent Matrix

Θ is the Projection Layer Weight



T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations, 2017.

Revisit Graph Convolutional Network

$$Z = \text{softmax}\left(\widetilde{W}_s ReLU\left(\widetilde{W}_s X \Theta^{(0)}\right)\Theta^{(1)}\right)$$

$$Z = \text{softmax}\left(\text{Re}LU\left(\widetilde{W}_s^2 X \Theta^{(0)}\right)\Theta^{(1)}\right)$$

- After exchanging the adjacent matrix $\widetilde{W}_s$ in the second layer with the internal ReLU function, GCN becomes **a special case of GLP** with the following three components:
  1. **Signal**: feature matrix, X
  2. **Filter**: $\widetilde{W}_s^{\,2} = \left(I - \tilde{L}_s\right)^2$, RNM filter with $k = 2$
  3. **Classifier**: 2-layer MLP

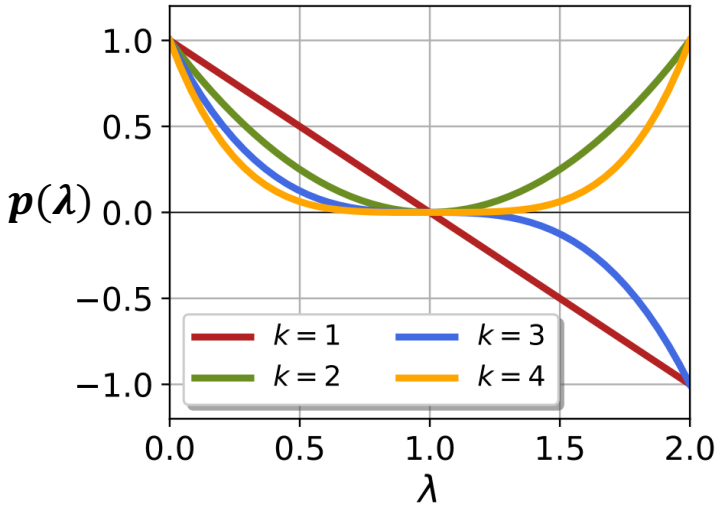
Understand GCN from the Perspective of GSP

$$Z = \mathrm{softmax}\left(\tilde{W}_s ReLU\left(\tilde{W}_s X \Theta^{(0)}\right)\Theta^{(1)}\right)$$

- **Why the Normalized Graph Laplacian ?**
  - Restrict eigenvalues within [0, 2], so the filter is close to a **low-pass** filter.
- **Why Two Convolutional Layers ?**
  - Become more **low-pass**.

RNM filter

$$p(\lambda) = (1-\lambda)^k$$



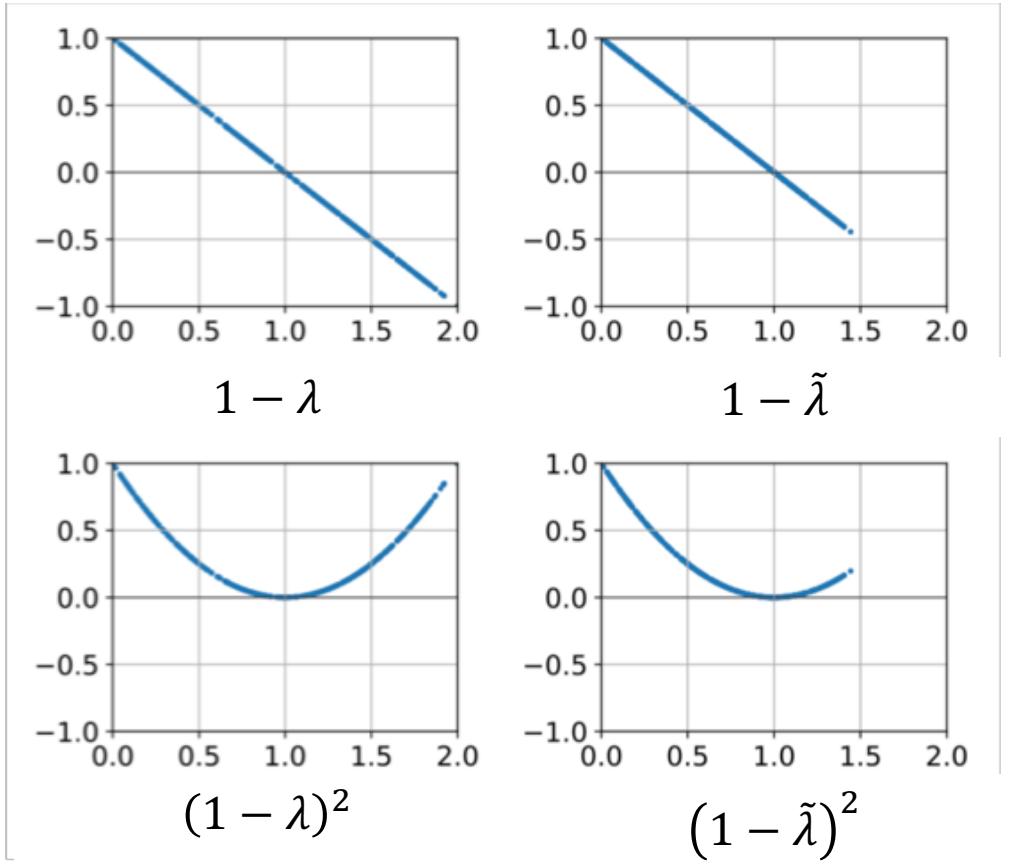
$$G = (I - L)^k$$

Understand GCN's Mechanism from the Perspective of GSP

$$Z = \mathrm{softmax}\left(\tilde{W}_s ReLU\left(\tilde{W}_s X \Theta^{(0)}\right)\Theta^{(1)}\right)$$

- **Why the Renormalization Trick ?**
  - Adding self-loops (the renormalization trick) shrinks the eigenvalues of $\tilde{L}_s$ from $[0, \lambda_m]$ to $\left[0, \frac{(d_m)}{(d_m+1)}\lambda_m\right]$. It **compresses the range of eigenvalues** and makes the filter **more low-pass**.
  - **Cora citation network**: The range of eigenvalues shrinks from [0, 2] to [0, 1.5]. It avoids amplifying eigenvalues near 2 and reduces noise.

The frequency responses on the eigenvalues of L_s and $\tilde{L}_s$ on the Cora citation network



Improved Graph Convolutional Networks (IGCN)
Use *k*-Order Filters

GCN: $Z = \mathrm{softmax}\left(\tilde{W}_s ReLU\left(\tilde{W}_s X\Theta^{(0)}\right)\Theta^{(1)}\right)$

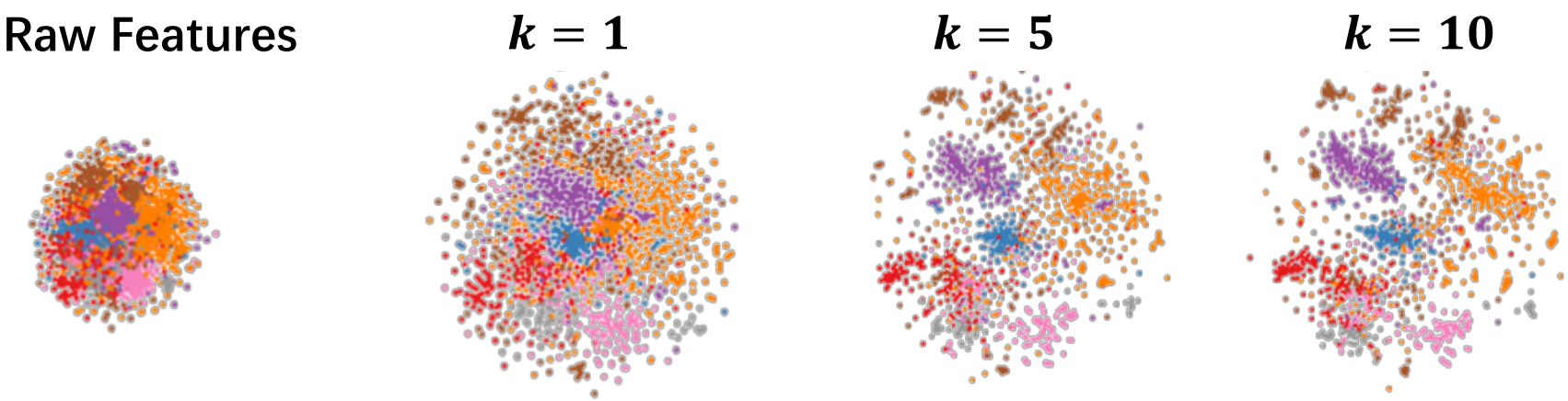
Renormalization (RNM) filter
$\tilde{W}_s = \boldsymbol{I} - \tilde{L}_s = \Phi\left(I - \tilde{\Lambda}\right)\Phi^{-1}$

IGCN: $Z = \mathrm{softmax}(\tilde{W}_s^k ReLU(\tilde{W}_s^k X\Theta^{(0)})\Theta^{(1)})$

$\tilde{W}_s^k$ **- *k*-order Renormalization (RNM) filter**

Design Filters via Selecting k

Small label rate	**Increase smoothing strength** (larger k)	Make features more similar
Large label rate	**Reduce smoothing strength** (smaller k)	Preserve feature diversity to avoid over-smoothing



Visualization of raw and filtered Cora features (by the RNM filter with different k)

Benefits of IGCN

- Easy to **achieve label efficiency** and **reduce model complexity** by conveniently adjusting k.
  - Modify filter strength to adapt to varying label rate.
  - Avoid stacking many layers as in GCN, making training much easier.

Semi-Supervised Classification on Graph
Experimental Setup

Dataset	Type	Vertices	Edges	Classes	Features	Label Rate
Cora	Citation Networks	3327	4732	6	3703	**20/4 labels per class**
CiteSeer		2708	5429	7	1433	
PubMed		19717	44338	3	500	
Large Cora		11881	64898	10	3780	
NELL	Knowledge Graph	65755	266144	210	5414	**10%, 1%, 0.1%**

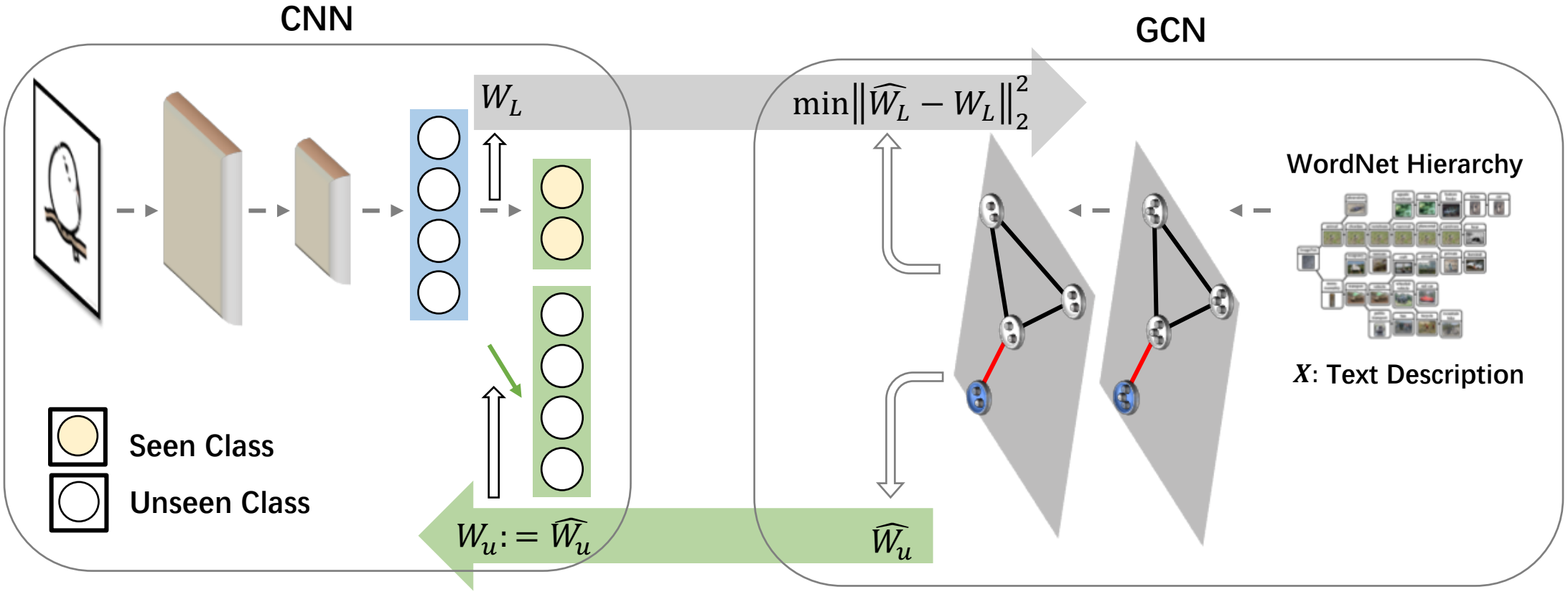
- Classifier for GLP: MLP
- Experiment with both RNM and AR filter
- When label rate $\leq 1\%, k = 10(RNM), \alpha = 20(AR)$, otherwise, $k = 5(RNM), \alpha = 10(AR)$
- Two layer MLP with 16 hidden units, 0.01 learning rate, 0.5 dropout rate, 5×10^{-4} L2 regularization

Semi-Supervised Classification on Graph

Table 2: Classification accuracy and running time on citation networks and NELL.

Label rate	20 labels per class				4 labels per class				10%	1%	0.1%
	Cora	CiteSeer	PubMed	Large Cora	Cora	CiteSeer	PubMed	Large Cora	NELL		
ManiReg	59.5	60.1	70.7	-	-	-	-	-	63.4	41.3	21.8
SemiEmb	59.0	59.6	71.7	-	-	-	-	-	65.4	43.8	26.7
DeepWalk	67.2	43.2	65.3	-	-	-	-	-	79.5	72.5	58.1
ICA	75.1	69.1	73.9	-	62.2	49.6	57.4	-	-	-	-
Planetoid	75.7	64.7	77.2	-	43.2	47.8	64.0	-	84.5	75.7	61.9
GAT	79.5	68.2	76.2	67.4	66.6	55.0	64.6	46.4	-	-	-
MLP	55.1 (0.6s)	55.4 (0.6s)	69.5 (0.6s)	48.0 (0.8s)	36.4 (0.6s)	38.0 (0.5s)	57.0 (0.6s)	30.8 (0.6s)	63.6 (2.1s)	41.6 (1.1s)	16.7 (1.0s)
LP	68.8 (0.1s)	48.0 (0.1s)	72.6 (0.1s)	52.5 (0.1s)	56.6 (0.1s)	39.5 (0.1s)	61.0 (0.1s)	37.0 (0.1s)	84.5 (0.7s)	75.1 (1.8s)	65.9 (1.9s)
GCN	79.9 (1.3s)	68.6 (1.7s)	77.6 (9.6s)	67.7 (7.5s)	65.2 (1.3s)	55.5 (1.7s)	67.7 (9.8s)	48.3 (7.4s)	81.6 (33.5s)	63.9 (33.5s)	40.7 (33.2s)
IGCN(RNM)	80.9 (1.2s)	69.0 (1.7s)	77.3 (10.0s)	68.9 (7.9s)	70.3 (1.3s)	57.4 (1.7s)	69.3 (10.3s)	52.1 (8.1s)	85.9 (42.4s)	76.7 (44.0s)	66.0 (46.6s)
IGCN(AR)	81.1 (2.2s)	69.3 (2.6s)	78.2 (11.9s)	69.2 (11.0s)	70.3 (3.0s)	58.0 (3.4s)	70.1 (13.6s)	52.5 (13.6s)	85.4 (77.9s)	75.7 (116.0s)	67.4 (116.0s)
GLP(RNM)	80.3 (0.9s)	68.8 (1.0s)	77.1 (0.6s)	68.4 (1.8s)	68.0 (0.7s)	56.7 (0.8s)	68.7 (0.6s)	51.1 (1.1s)	86.0 (35.9s)	76.1 (37.3s)	65.4 (38.5s)
GLP(AR)	80.8 (1.0s)	69.3 (1.2s)	78.1 (0.7s)	69.0 (2.4s)	67.5 (0.8s)	57.3 (1.1s)	69.7 (0.8s)	51.6 (2.3s)	80.3 (57.4s)	67.4 (76.6s)	55.2 (78.6s)

Semi-Supervised Regression for Zero-Shot Image Recognition



X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In CVPR, pages 6857–6866, 2018.

Zero-Shot Image Recognition
Experimental Setup

Datasets	**AWA2, a subset of ImageNet,** which is an mage database organized according to the WordNet hierarchy. All categories form a graph through "is a kind of" relation.	
Baselines	**Devise**	A. Frome et al. Devise: A deep visual-semantic embedding model. In NeuralPS, pages 2121–2129, 2013.
	GCNZ	X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In CVPR, pages 6857–6866, 2018.
	SYNC	S. Changpinyo et al., Synthesized classifiers for zero-shot learning. In CVPR, pages 5327– 5336, 2016.
	GPM	M. Kampffmeyer et al., Rethinking knowledge graph propagation for zero-shot learning. *arXiv preprint arXiv:1805.11724*, 2018.
	DGPM	
	ADGPM	
Settings	• 1000 training classes. 21K classes in total. • Use a ResNet-50 model that has been pre-trained on the ImageNet 2012. • Two-layer structure with 2048 hidden units.	

Zero-Shot Image Recognition
Results

Results for unseen classes in AWA2			
Method	Accuracy	Method	Accuracy
Devise	59.7	SYNC	46.6
GCNZ	68.0	DGPM	67.2
GPM	77.3	ADGPM	76.0
IGCN (RNM)		GLP (RNM)	
$\mathbf{k = 2}$	**77.9**	$\mathbf{k = 2}$	76.0
$\mathbf{k = 4}$	**77.7**	$\mathbf{k = 4}$	75.0
$\mathbf{k = 6}$	73.1	$\mathbf{k = 6}$	73.0

IGCN	**bat**
GLP	**bat**
GPM	seal
GCNZ	Walrus
ADGPM	seal
DGPM	**bat**

IGCN	**dolphin**
GLP	**dolphin**
GPM	**dolphin**
GCNZ	seal
ADGPM	walrus
DGPM	seal

Summary

- Propose **a unified graph filtering framework** for label-efficient semi-supervised learning.
- Offer **new insights** into existing methods that substantially **improve modeling capability and reduce model complexity**.
- Demonstrate model effectiveness on various semi-supervised **classification and regression** tasks.

Investigate and develop deeper insights into the design of proper filters for various application scenarios.

Future Work

We Are Hiring!

We have multiple positions for PhD and RA (Research Assistant/Associate) , Interested applicants may send their resumes to csxmwu@comp.polyu.edu.hk

- **Research Directions**:
  - **Few-shot Learning**
  - **Semi-supervised Learning**
  - **Meta-learning**
- **Applications**
  - **Computer Vision (CV)**
  - **Natural Language Processing (NLP)**

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

45TH Department of Computing 電子計算學系

Question and Feedback