Graph Neural Networks

姜成翰

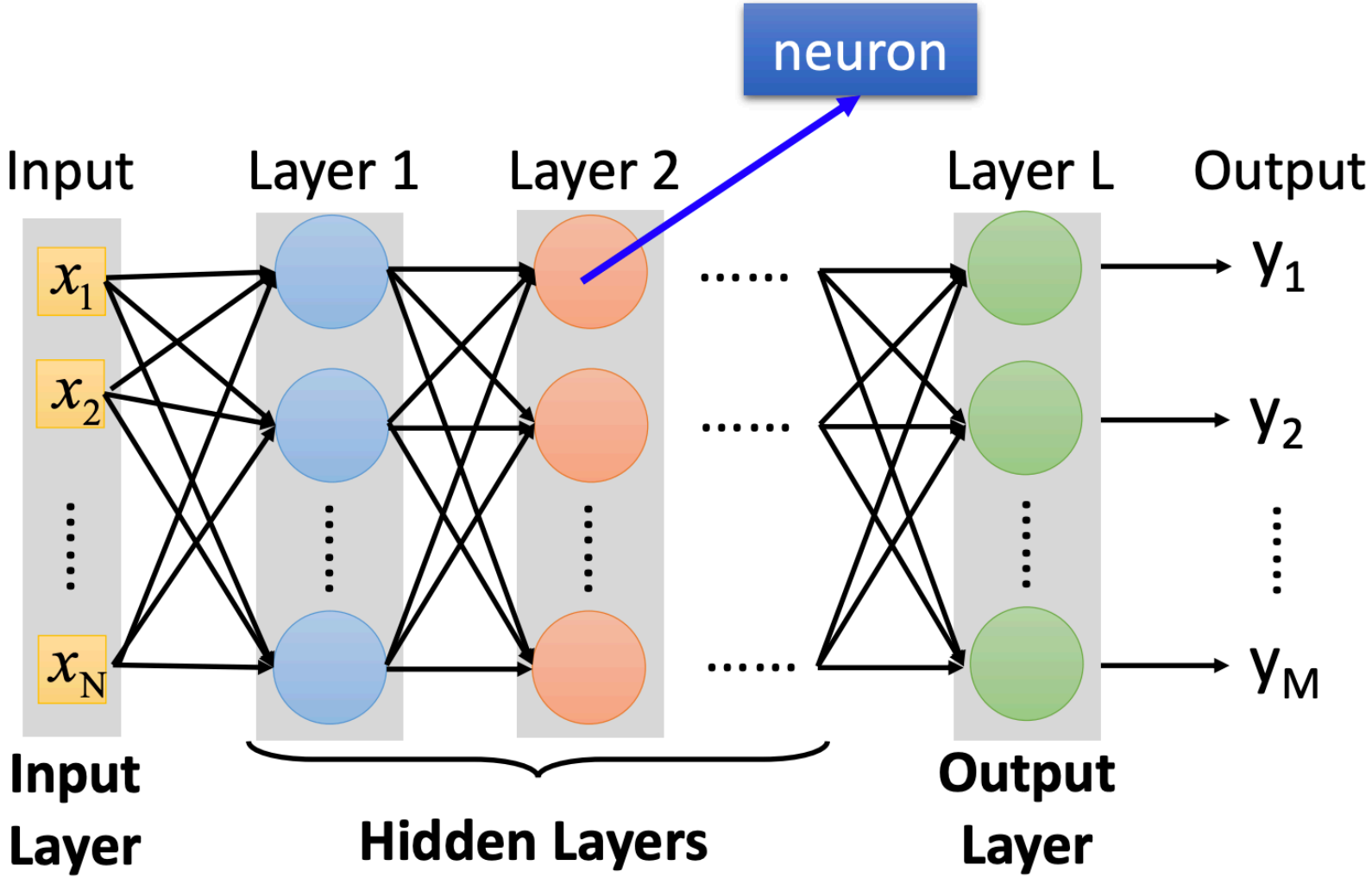
2020.03.26

Outline

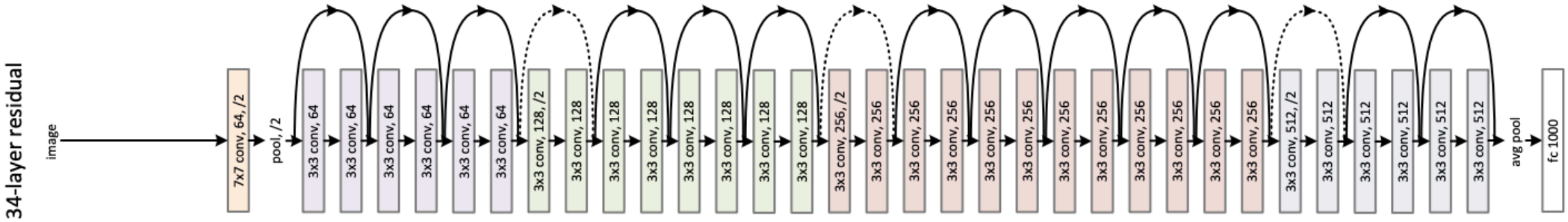
- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

Graph Neural Networks

Neural Network

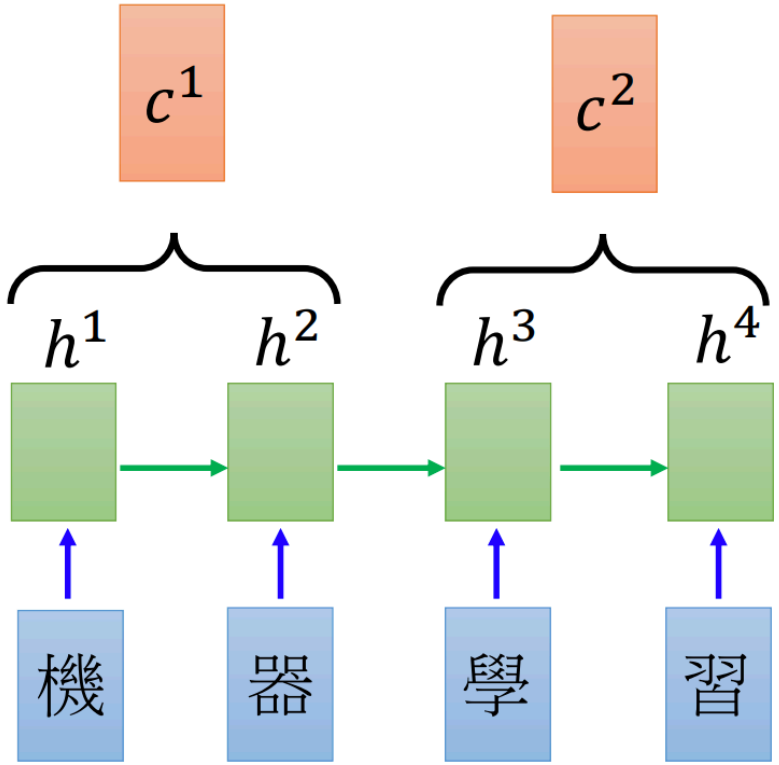


CNN

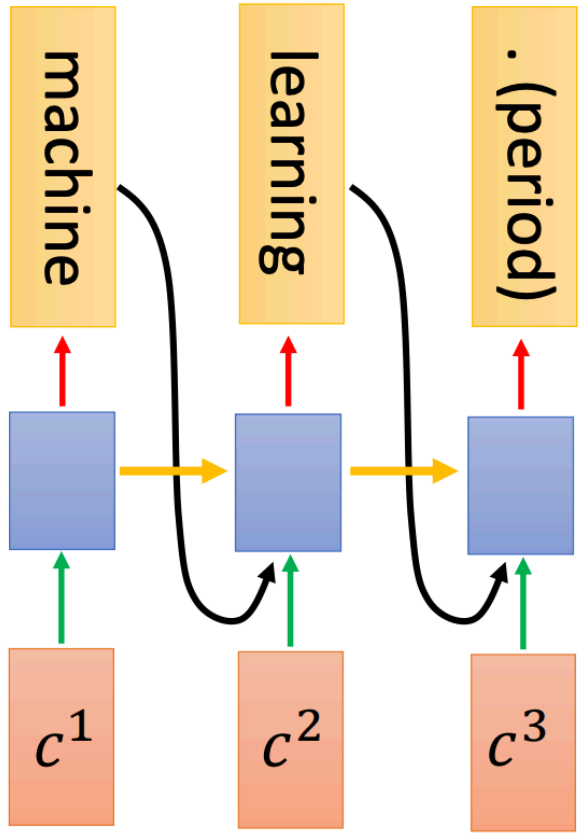


https://arxiv.org/pdf/1512.03385.pdf

RNN

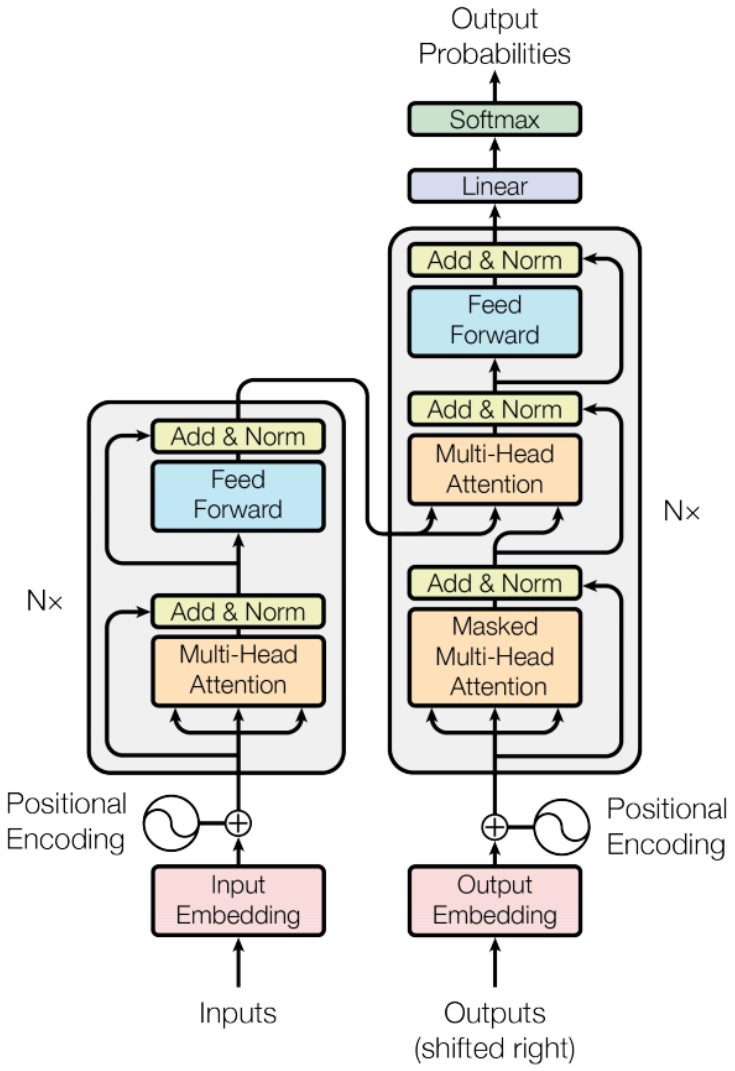


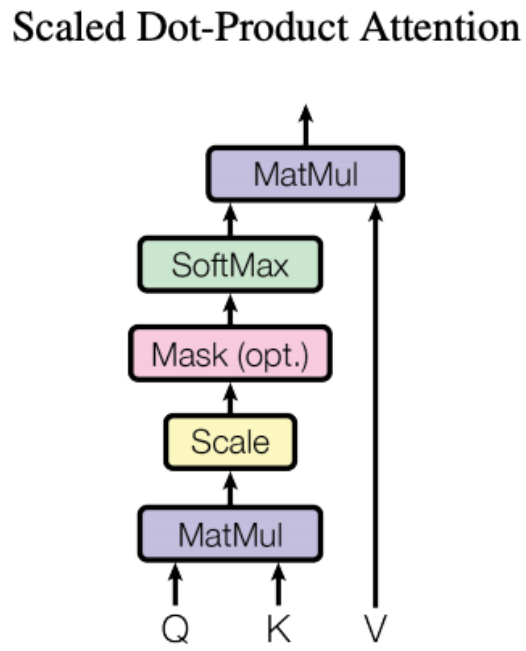
Encoder



Decoder

Transformer

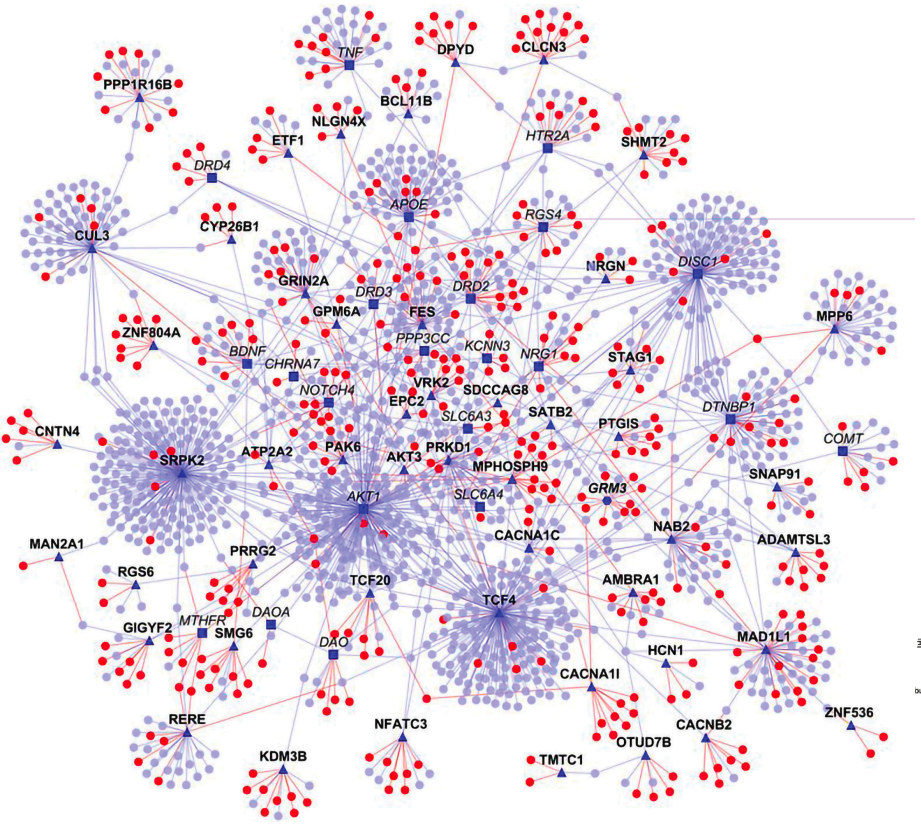




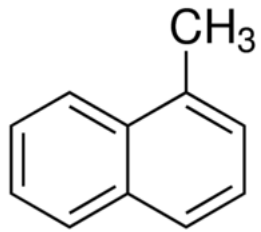
https://arxiv.org/pdf/1706.03762.pdf

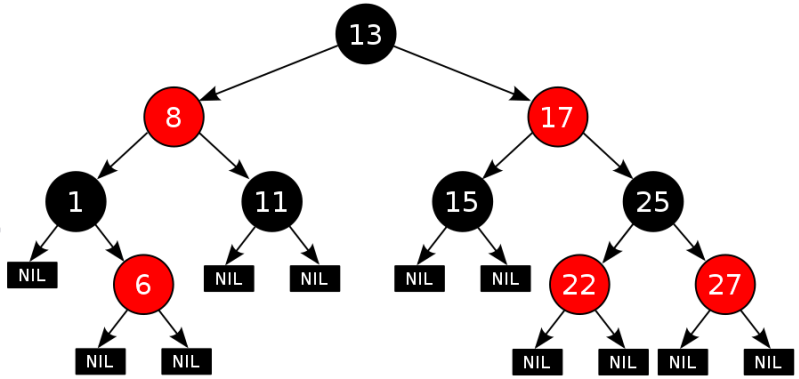
http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2019/Lecture/Transformer%20(v5).pdf

Graph

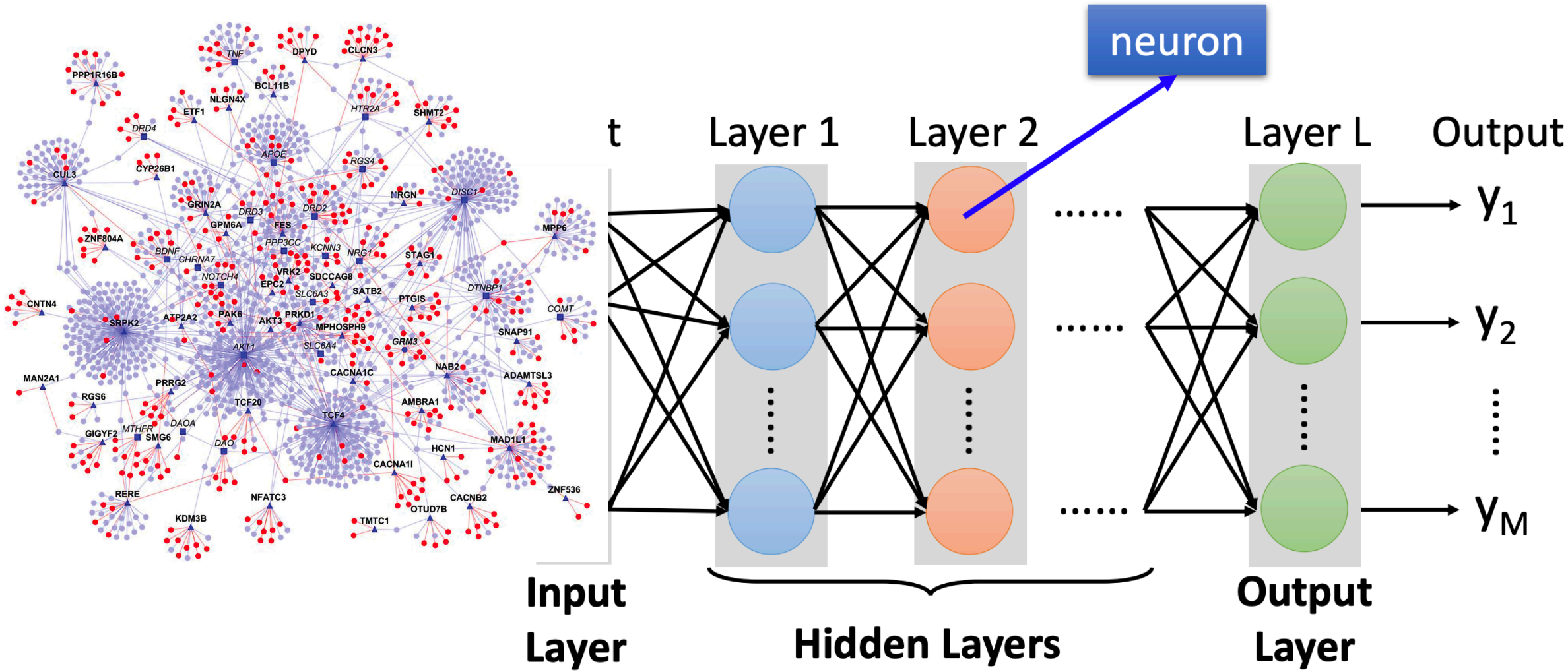

TNF
DPYD
CLCN3
PPP1R16B
BCL11B
NLGN4X
ETF1
HTR2A
SHMT2
DRD4
APOE
RGS4
CYP26B1
CUL3
NRGN
DISC1
GRIN2A
DRD3
DRD2
GPM6A
FES
MPP6
ZNF804A
BDNF
PPP3CC
KCNN3
NRG1
STAG1
CHRNA7
NOTCH4
VRK2
EPC2
SDCCAG8
SLC6A3
SATB2
DTNBP1
CNTN4
PTGIS
COMT
SRPK2
ATP2A2
PAK6
AKT3
PRKD1
MPHOSPH9
SNAP91
AKT1
SLC6A4
GRM3
CACNA1C
NAB2
ADAMTSL3
MAN2A1
PRRG2
RGS6
TCF20
AMBRA1
TCF4
GIGYF2
MTHFR
DAOA
SMG6
DAO
MAD1L1
HCN1
CACNA1I
RERE
NFATC3
CACNB2
ZNF536
KDM3B
TMTC1
OTUD7B


圖例 Legend
BR 文湖線 Wenhu Line
R 淡水信義線 Tamsui-Xinyi Line
G 松山新店線 Songshan-Xindian Line
O 中和新蘆線 Zhonghe-Xinlu Line
BL 板南線 Bannan Line
Y 環狀線 Circular Line
一般車站 Regular Station
轉乘站 Transfer Station
端點站 Terminal Station
機場 Airport
高鐵 HSR
台鐵 TRA
桃園機場捷運 Taoyuan Airport MRT
V 淡海輕軌 Danhai LRT
連通道 Passage
站外轉乘 Out of Station Transfer
溫泉 Hot Spring
淡水 Tamsui
紅樹林 Hongshulin
竹圍 Zhuwei
關渡 Guandu
忠義 Zhongyi
復興崗 Fuxinggang
北投 Beitou
奇岩 Qiyan
唭哩岸 Qilian
石牌 Shipai
明德 Mingde
芝山 Zhishan
士林 Shilin
劍潭 Jiantan
圓山 Yuanshan
民權西路 Minquan W. Rd.
中山國小 Zhongshan Elementary School
行天宮 Xingtian Temple
松江南京 Songjiang Nanjing
南京復興 Nanjing Fuxing
南京三民 Nanjing Sanmin
松山 Songshan
中山 Zhongshan
台北車站 Taipei Main Station
忠孝新生 Zhongxiao Xinsheng
忠孝復興 Zhongxiao Fuxing
國父紀念館 Sun Yat-Sen Memorial Hall
忠孝敦化 Zhongxiao Dunhua
市政府 Taipei City Hall
永春 Yongchun
後山埤 Houshanpi
南港 Nangang
昆陽 Kunyang
南港展覽館 Taipei Nangang Exhibition Center
南港軟體園區 Nangang Software Park
東湖 Donghu
大直 Dazhi
松山機場 Songshan Airport
中山國中 Zhongshan Junior High School
劍南路 Jiannan Rd.
西湖 Xihu
港墘 Gangqian
文德 Wende
內湖 Neihu
大湖公園 Dahu Park
葫洲 Huzhou
台北小巨蛋 Taipei Arena
南京三民 Nanjing Sanmin
松山 Songshan
南港 Nangang
昆陽 Kunyang
後山埤 Houshanpi
永春 Yongchun
市政府 Taipei City Hall
忠孝敦化 Zhongxiao Dunhua
國父紀念館 Sun Yat-Sen Memorial Hall
忠孝復興 Zhongxiao Fuxing
忠孝新生 Zhongxiao Xinsheng
台北車站 Taipei Main Station
台北101/世貿 Taipei 101/ World Trade Center
象山 Xiangshan
辛亥 Xinhai
麟光 Linguang
六張犁 Liuzhangli
科技大樓 Technology Building
大安 Dan
信義安和 Xinyi Anhe
大安森林公園 Daan Park
東門 Dongmen
古亭 Guting
台電大樓 Taipower Building
公館 Gongguan
萬隆 Wanlong
景美 Jingmei
萬芳醫院 Wanfang Hospital
萬芳社區 Wanfang Community
木柵 Muzha
動物園 Taipei Zoo
動物園站 Taipei Zoo Station
動物園南站 Taipei Zoo South Station
指南宮站 Zhinan Temple Station
貓空站 Maokong Station
貓空纜車 Maokong Gondola
大坪林 Dapinglin
七張 Qizhang
新店區公所 Xindian District Office
新店 Xindian
十四張 Shisizhang
小碧潭 Xiaobitan
秀朗橋 Xiulang Bridge
景平 Jingping
景安 Jingan
中和 Zhonghe
橋和 Qiaohe
南勢角 Nanshijiao
中原 Zhongyuan
永安市場 Yongan Market
頂溪 Dingxi
板新 Banxin
板橋 Banqiao
新埔 Xinpu
江子翠 Jiangzicui
龍山寺 Longshan Temple
西門 Ximen
北門 Beimen
小南門 Xiaonanmen
中正紀念堂 Chiang Kai-Shek Memorial Hall
台大醫院 NTU Hospital
善導寺 Shandao Temple
雙連 Shuanglian
台北橋 Taipei Bridge
大橋頭 Daqiaotou
三重國小 Sanchong Elementary School
三和國中 Sanhe Junior High School
徐匯中學 St. Ignatius High School
三民高中 Sanmin Senior High School
蘆洲 Luzhou
菜寮 Cailiao
三重 Sanchong
先嗇宮 Xianse Temple
幸福 Xinfu
頭前庄 buqianzhuang
新莊 Xinzhuang
輔大 Fu Jen University
丹鳳 Danfeng
迴龍 Huilong
新埔民生 Xinpu Minsheng
板橋 Banqiao
府中 Fuzhong
亞東醫院 Far Eastern Hospital
海山 Haishan
土城 Tucheng
永寧 Yongning
頂埔 Dingpu
新莊 Xinzhuang
三重 Sanchong
新北產業園區 New Taipei Industrial Park
臺北大眾捷運股份有限公司 版權所有
Copyright © Taipei Rapid Transit Corporation
各運具路線僅為示意
非實際地理位置
Routes are depicted schematically and do not reflect the exact location.


CH3

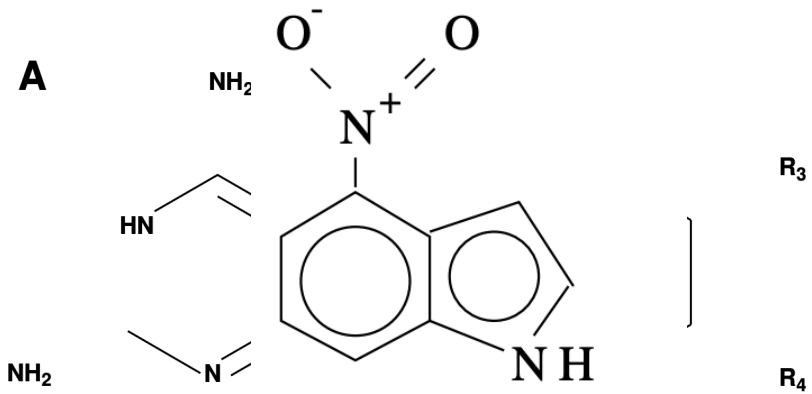

13
8
17
1
11
15
25
NIL
6
22
27

GNN


neuron
t
Layer 1
Layer 2
Layer L
Output
y_1
y_2
y_M
Input Layer
Hidden Layers
Output Layer

GNN: Why?

- ✓ Why do we need GNN?
  - ➢ Classification



4-nitroindole

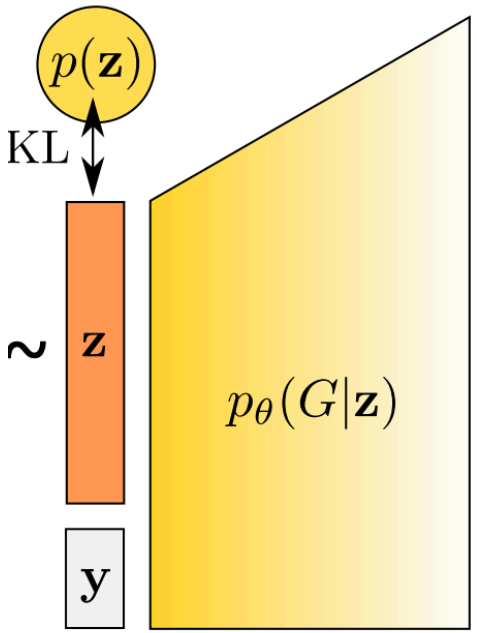
會不會導致突變

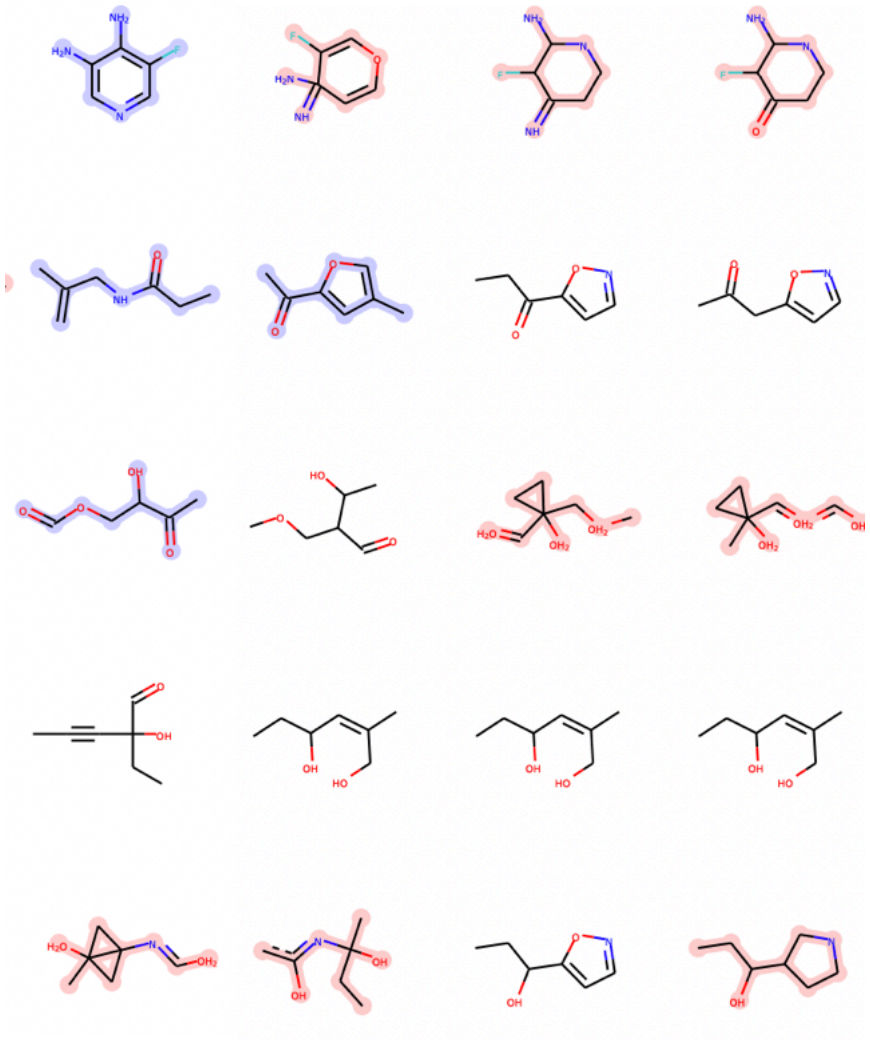
1
2

https://persagen.com/files/misc/scarselli2009graph.pdf

GNN: Why?

- ✓ Why do we need GNN?
  - ➢ Generation





GraphVAE: https://arxiv.org/pdf/1802.03480.pdf

GNN: Why?

✓ Example: テセウスの船

Name

Feature

Who is the murderer?

GNN: Why?

✓ We can train a classifier

$f($ 

$) = False$

$f($ 

$) = False$

$f($ 

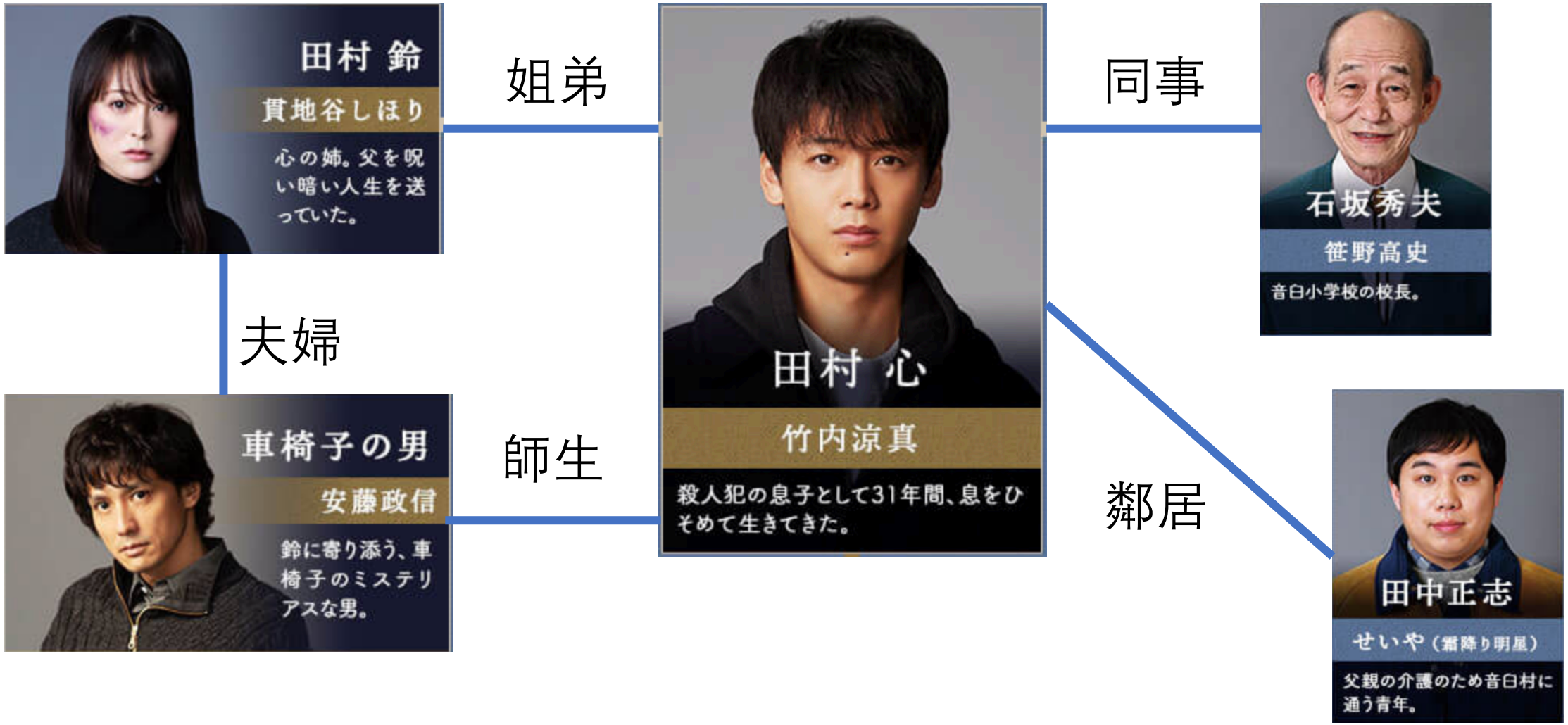
$) = False$

$f($ 

$) = ?$

GNN: Why?

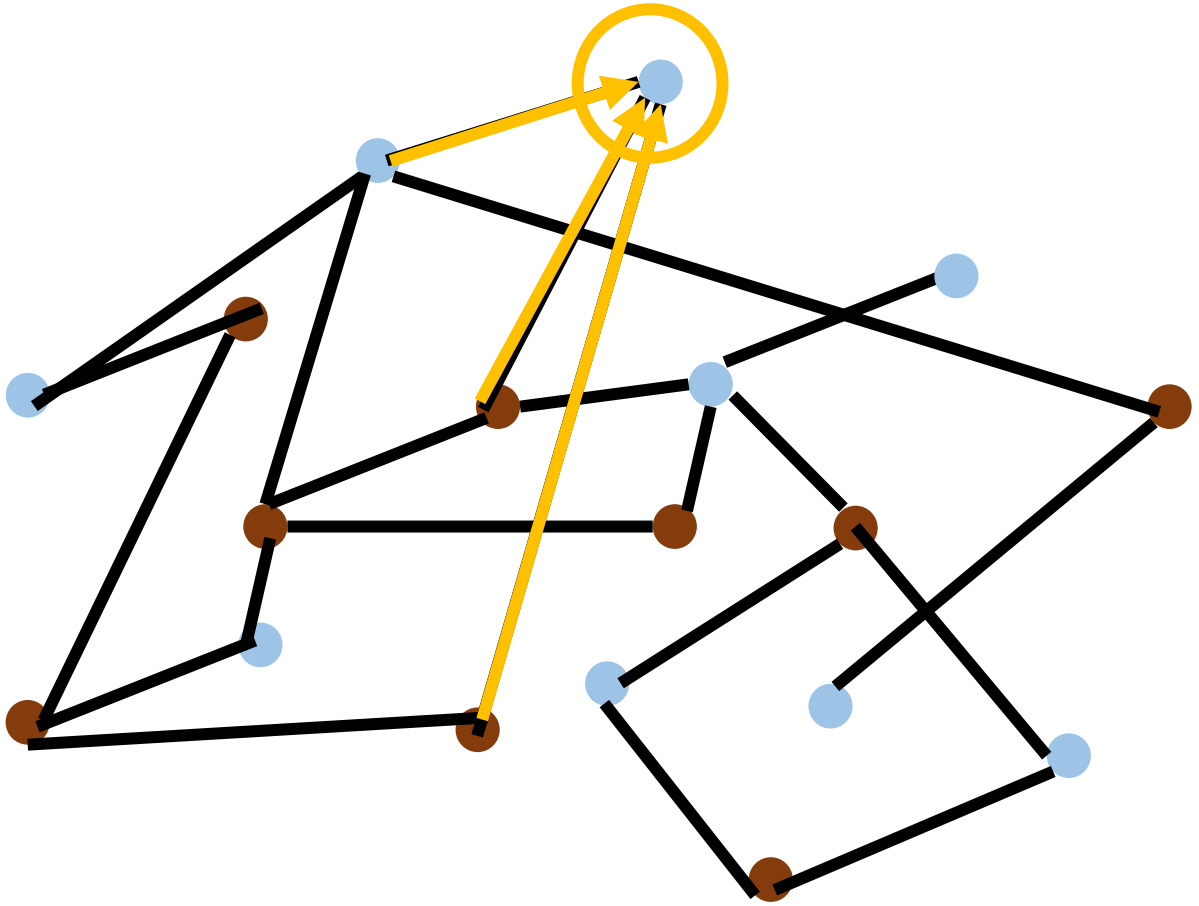
- ✓ The data may have underlying structure and relationship



GNN: Why?

- ✓ How do we utilize the structures and relationship to help our model?
- ✓ What if the graph is larger, like 20k nodes?
- ✓ What if we don‘t have the all the labels?

GNN: Why



Labeled Node

∧

Unlabeled Node

Semi-Supervised Learning

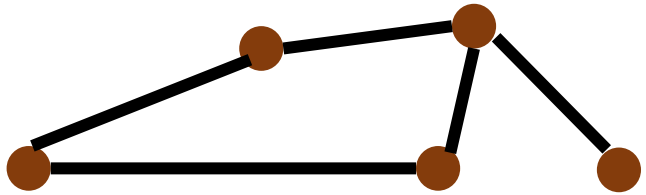
A node can learn the structure from its neighbors, but how?

GNN: How?

✓ Convolution?

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image with
3 x 3 kernel



GNN: How?

- ✓ How to embed node into a feature space using convolution?
- ✓ Solution 1: Generalize the concept of convolution (corelation) to graph >> Spatial-based convolution
- ✓ Solution 2: Back to the definition of convolution in signal processing >> Spectral-based convolution

Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

GNN Roadmap

Theoretical analysis: GIN, GCN

Convolution

Spatial-based

Aggregation	Method
Sum	NN4G
Mean	DCNN, DGC, GraphSAGE
Weighted sum	MoNET, GAT, GIN
LSTM	GraphSAGE
Max Pooling	GraphSAGE

Spectral-based

ChebNet → GCN → HyperGCN

Tasks

- Supervised classification
- Semi-Supervised Learning
- Representation learning: Graph InfoMax
- Generation: GraphVAE, MolGAN, etc.

~~Application: Natural Language Processing~~

Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

Tasks, Dataset, and Benchmark

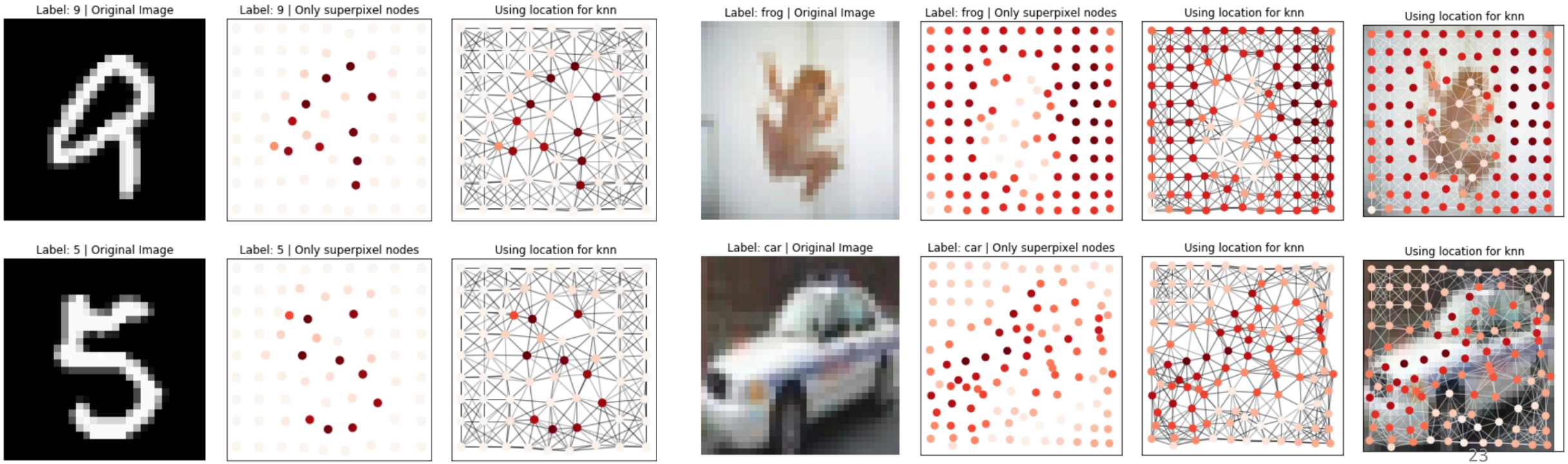
- ✓ Tasks
  - ➢ Semi-supervised node classification
  - ➢ Regression
  - ➢ Graph classification
  - ➢ Graph representation learning
  - ➢ Link prediction
- ✓ Common dataset
  - ➢ CORA: citation network. 2.7k nodes and 5.4k links
  - ➢ TU-MUTAG: 188 molecules with 18 nodes on average

https://arxiv.org/pdf/2003.00982.pdf

Benchmark tasks

✓ Graph Classification: SuperPixel MNIST and CIFAR10

Domain/Construction	Dataset	# graphs	# nodes
Computer Vision/ Graphs constructed with super-pixels	MNIST	70K	40-75
	CIFAR10	60K	85-150



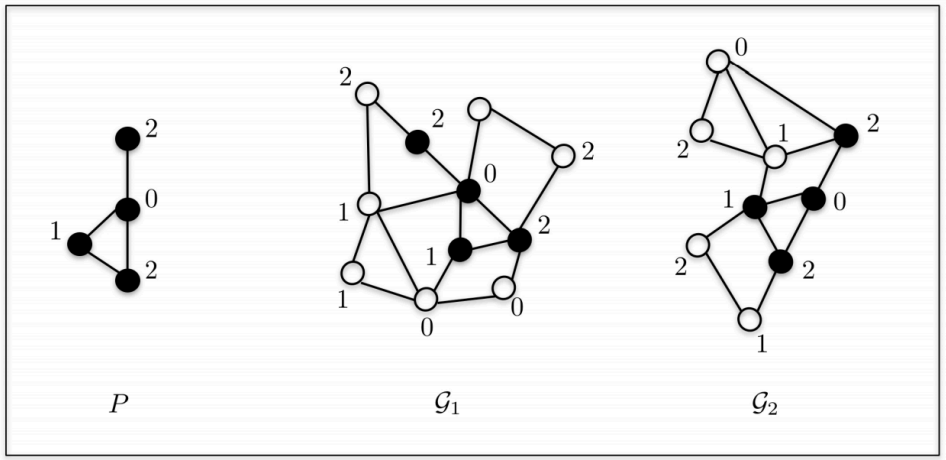
https://arxiv.org/pdf/1711.07553.pdf

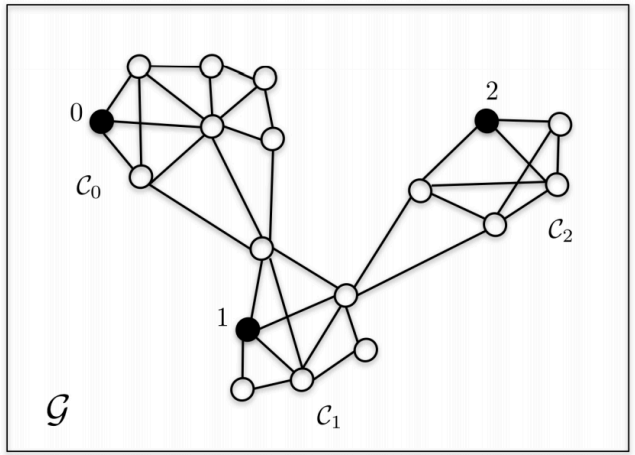
Benchmark tasks

- ✓ Regression: ZINC molecule graphs dataset

Domain/Construction	Dataset	# graphs	# nodes
Chemistry/ Real-world molecular graphs	ZINC	12K	9-37
Artificial/ Graphs generated from Stochastic Block Model	PATTERN	14K	50-180
	CLUSTER	12K	40-190

- ✓ Node classification:Stochastic Block Model dataset
  - ➢ graph pattern recognition and semi-supervised graph clustering

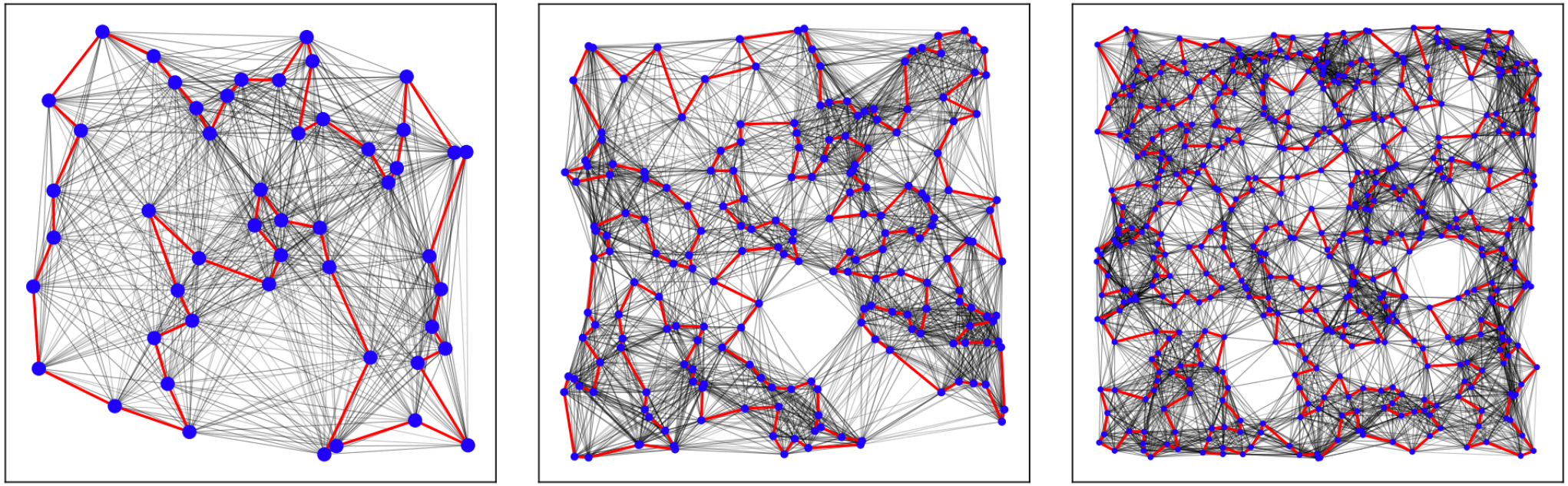




Benchmark tasks

✓ Edge classification: Traveling Salesman Problem

Domain/Construction	Dataset	# graphs	# nodes
Artificial/ Graphs generated from uniform distribution	TSP	12K	50-500



(a) TSP50 (b) TSP200 (c) TSP500

Results

✓ SuperPixel

Dataset	Model	#Param	No Residual	
			Acc	Epoch/Total
MNIST	MLP	104044	94.46±0.28	21.82s/1.02hr
	MLP (Gated)	105717	95.18±0.18	22.43s/0.73hr
	GCN	101365	**89.05±0.21**	79.18s/1.76hr
	GraphSage	102691	**97.20±0.17**	76.80s/1.42hr
	GIN	105434	**93.96±1.30**	34.61s/0.74hr
	DiffPool	106538	**94.66±0.48**	171.38s/4.45hr
	GAT	110400	**95.56±0.16**	377.06s/6.35hr
	MoNet	104049	**89.73±0.48**	567.12s/12.05hr
	GatedGCN	104217	**97.36±0.12**	127.15s/2.13hr
	GatedGCN-E*	104217	**97.47±0.13**	127.86s/2.15hr
CIFAR10	MLP	104044	56.01±0.90	21.82s/1.02hr
	MLP (Gated)	106017	56.78±0.12	27.85s/0.68hr
	GCN	101657	51.64±0.45	100.30s/2.44hr
	GraphSage	102907	**66.08±0.24**	96.00s/1.79hr
	GIN	105654	47.66±0.47	44.30s/0.93hr
	DiffPool	108042	56.84±0.37	299.64s/10.42hr
	GAT	110704	**65.48±0.33**	386.14s/7.75hr
	MoNet	104229	50.99±0.17	869.90s/21.79hr
	GatedGCN	104357	**68.92±0.38**	145.14s/2.49hr
	GatedGCN-E*	104357	**69.37±0.48**	145.66s/2.43hr

Results

✓ Regression

Model	#Param	No Residual	
		Acc/MAE	Epoch/Total
MLP	108975	0.710±0.001	1.19s/0.02hr
MLP (Gated)	106970	0.681±0.005	1.16s/0.03hr
GCN	103077	0.525±0.007	2.97s/0.09hr
GraphSage	105031	**0.410±0.005**	3.20s/0.10hr
GIN	103079	**0.408±0.008**	2.50s/0.06hr
DiffPool	110561	0.514±0.007	12.36s/0.38hr
GAT	102385	0.496±0.004	21.03s/0.62hr
MoNet	106002	0.444±0.024	11.75s/0.34hr
GatedGCN	105735	**0.422±0.006**	6.12s/0.17hr
GatedGCN-E*	105875	**0.365±0.009**	6.17s/0.17hr

Results

✓ SBM

Dataset	Model	#Param	No Residual	
			Acc	Epoch/Total
PATTERN	MLP	105263	50.13±0.00	8.68s/0.10hr
	MLP (Gated)	103629	50.13±0.00	9.78s/0.12hr
	GCN	100923	55.22±0.17	97.46s/2.30hr
	GraphSage	98607	**81.25±3.84**	79.43s/2.14hr
	GIN	100884	**98.25±0.38**	14.11s/0.37hr
	GAT	109936	88.91±4.48	229.65s/8.78hr
	MoNet	103775	**97.89±0.89**	870.05s/24.86hr
	GatedGCN	104003	**97.24±1.19**	115.03s/2.59hr
CLUSTER	MLP	106015	20.97±0.01	6.54s/0.08hr
	MLP (Gated)	104305	20.97±0.01	7.37s/0.09hr
	GCN	101655	34.85±0.65	66.81s/1.21hr
	GraphSage	99139	**53.90±4.12**	54.40s/1.19hr
	GIN	103544	**52.54±1.03**	11.57s/0.27hr
	GAT	110700	**54.12±1.21**	158.46s/4.53hr
	MoNet	104227	39.48±2.21	600.04s/11.18hr
	GatedGCN	104355	50.18±3.03	80.66s/2.07hr

Results

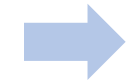
✓ TSP

Model	#Param	No Residual	
		F1	Epoch/Total
MLP	94394	0.548±0.003	53.92s/2.85hr
MLP (Gated)	115274	0.548±0.001	54.39s/2.44hr
GCN	108738	0.547±0.003	164.41s/10.28hr
GraphSage	98450	**0.657±0.002**	147.22s/14.33hr
GIN	118574	**0.657±0.001**	74.71s/5.60h
GAT	109250	0.567±0.003	360.74s/20.55hr
MoNet	94274	0.569±0.002	1472.65s/42.44hr
GatedGCN	94946	**0.791±0.003**	202.12s/15.20hr
GatedGCN-E*	94946	0.794±0.003	201.32s/15.05hr

Results

✓ TSP

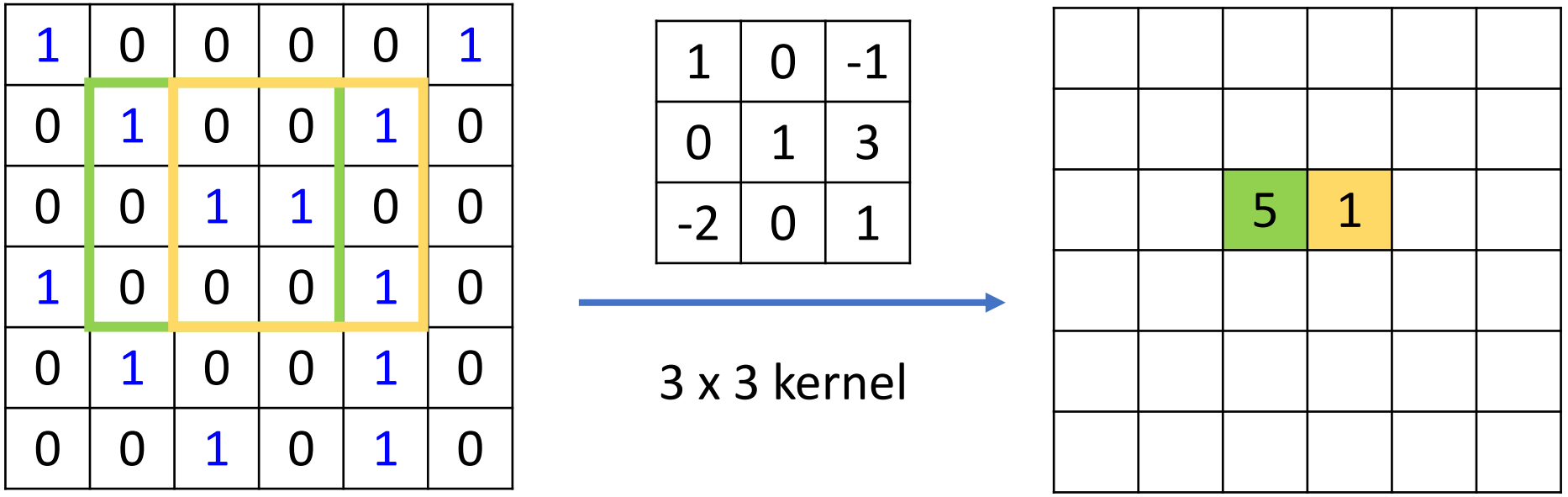
Model	L	#Param	Residual		No Residual	
			F1	Epoch/Total	F1	Epoch/Total
GCN	4	108738	**0.628**	165.15s/5.69hr	0.552	168.72s/6.14hr
	8	175810	**0.639**	279.33s/9.86hr	0.568	281.56s/14.16hr
	16	309954	**0.651**	502.59s/21.37hr	0.532	507.35s/10.72hr
	32	578242	**0.666**	1042.46s/28.96hr	0.361	1031.62s/15.19hr
GIN	4	118574	**0.653**	71.41s/2.50hr	**0.653**	75.63s/3.34hr
	8	223866	**0.675**	93.95s/4.26hr	0.674	93.41s/5.19hr
	16	434450	**0.681**	146.09s/5.68hr	0.642	144.52s/2.89hr
	32	855618	**0.669**	274.5s/3.81hr	0.6063	282.97s/4.40hr
GatedGCN	4	94946	**0.792**	214.67s/6.50hr	**0.787**	212.67s/9.75hr
	8	179170	**0.817**	374.39s/14.56hr	0.807	367.68s/19.72hr
	16	347618	**0.833**	685.41s/22.85hr	0.810	678.76s/22.07hr
	32	684514	**0.843**	1760.56s/48.00hr	0.722	1760.55s/33.27hr



Outline

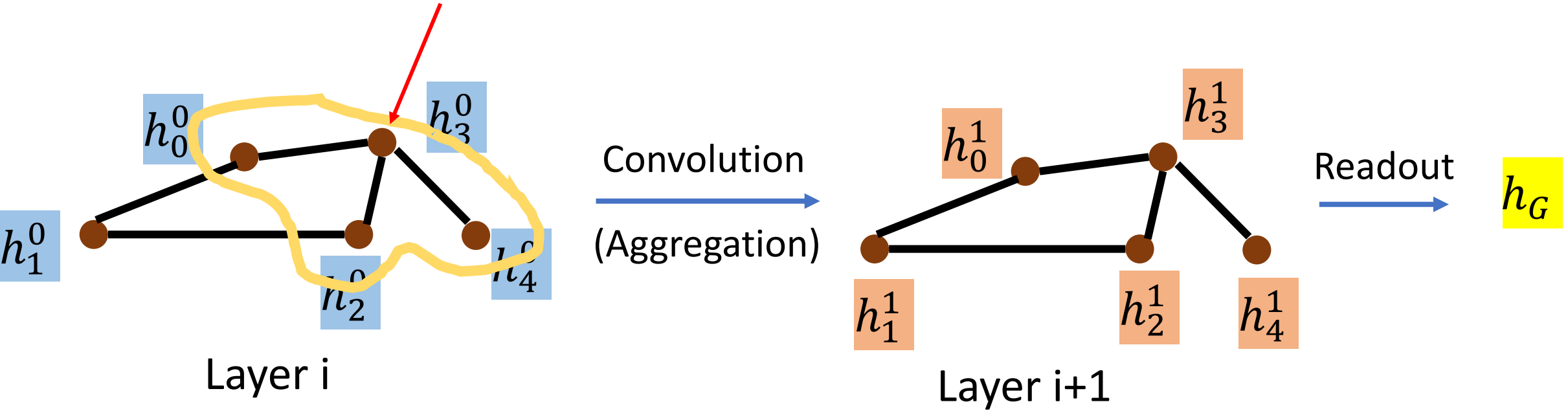
- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

Review: Convolution



Spatial-based Convolution

- ✓ Terminology:
  - ➢ Aggregate: 用 neighbor feature update 下一層的 hidden state
  - ➢ Readout: 把所有 nodes 的 feature 集合起來代表整個 graph



https://ieeexplore.ieee.org/document/4773279

NN4G (Neural Networks for Graph)

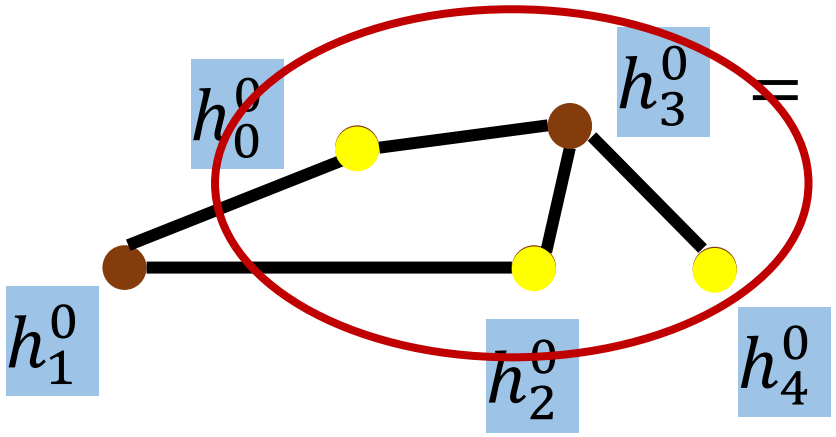
Hidden layer 1:

$h_3^1 = \hat{w}_{1,0}\,(h_0^0 + h_2^0 + h_4^0)$

$+$

$\bar{w}_1$

Hidden layer 0:

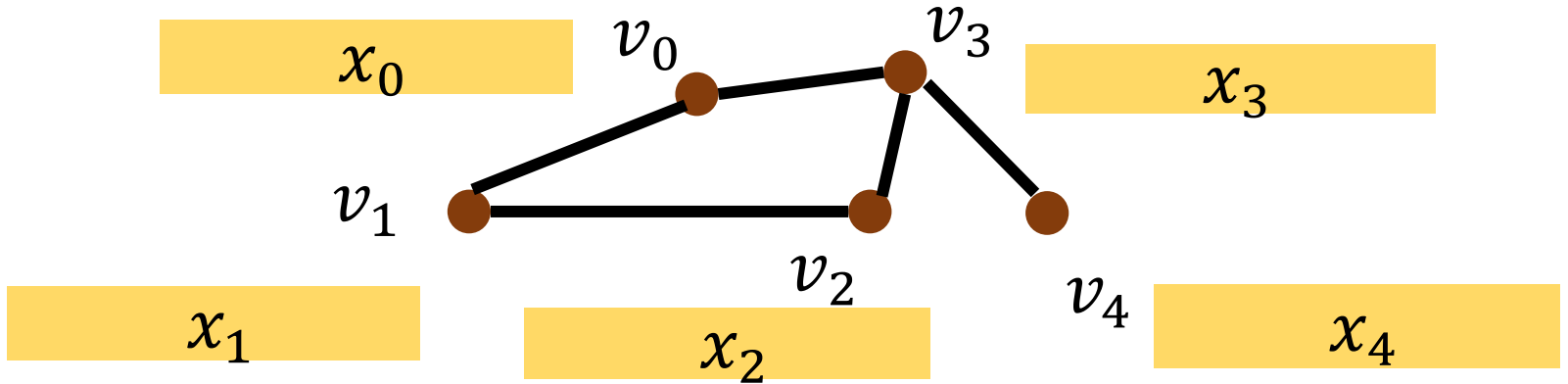


$\bar{w}_0 \cdot x_3$

x_3

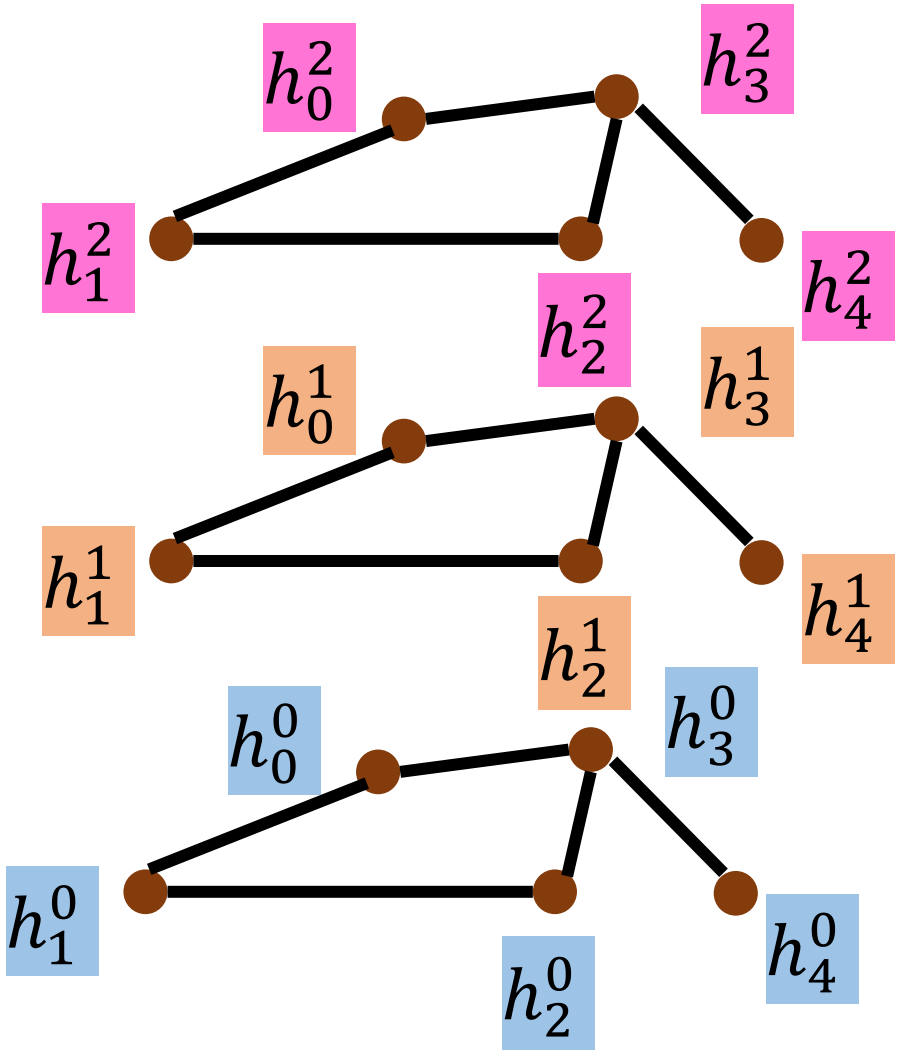
h_{node}^{layer}

Input layer



NN4G (Neural Networks for Graph)

✓ Readout:



$$X_2 = MEAN(\ h^2\ )$$

$$X_1 = MEAN(\ h^1\ )$$

$$X_0 = MEAN(\ h^0\ )$$

w_2

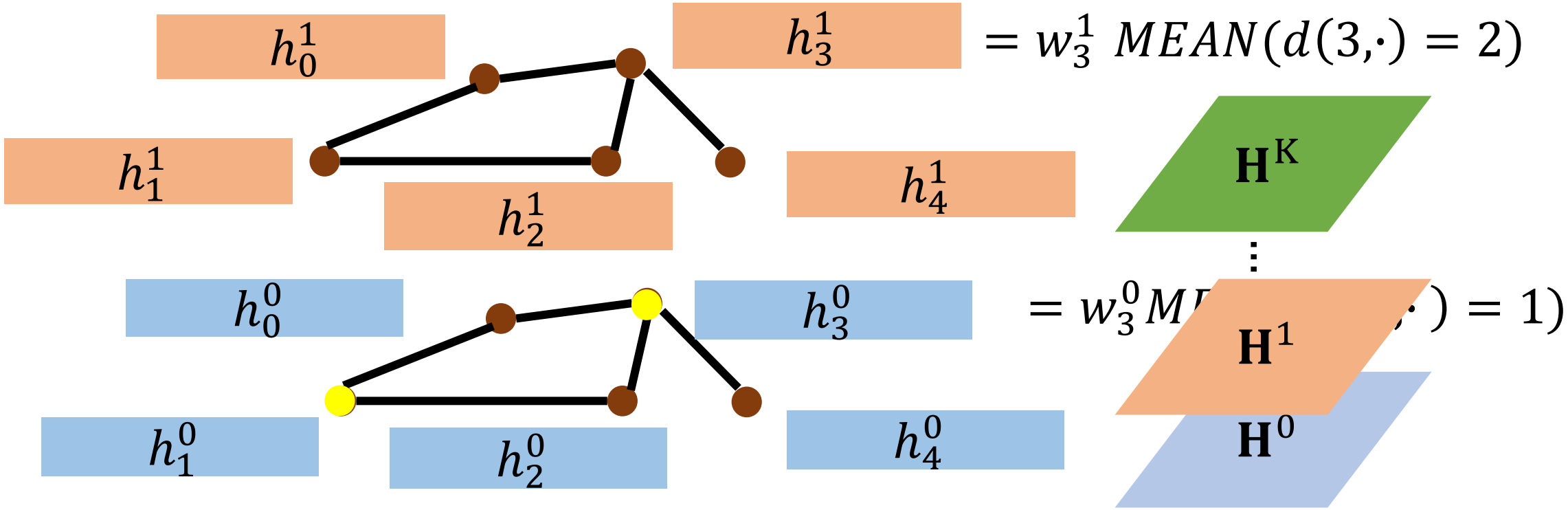
w_1

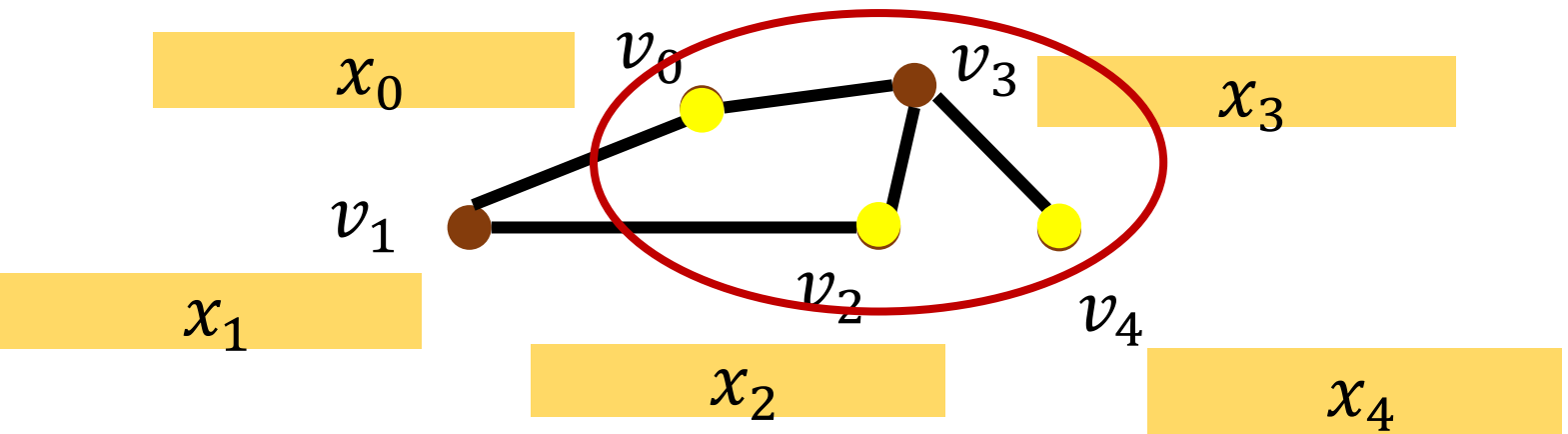
w_0

$+$

y

DCNN (Diffusion-Convolution Neural Network)

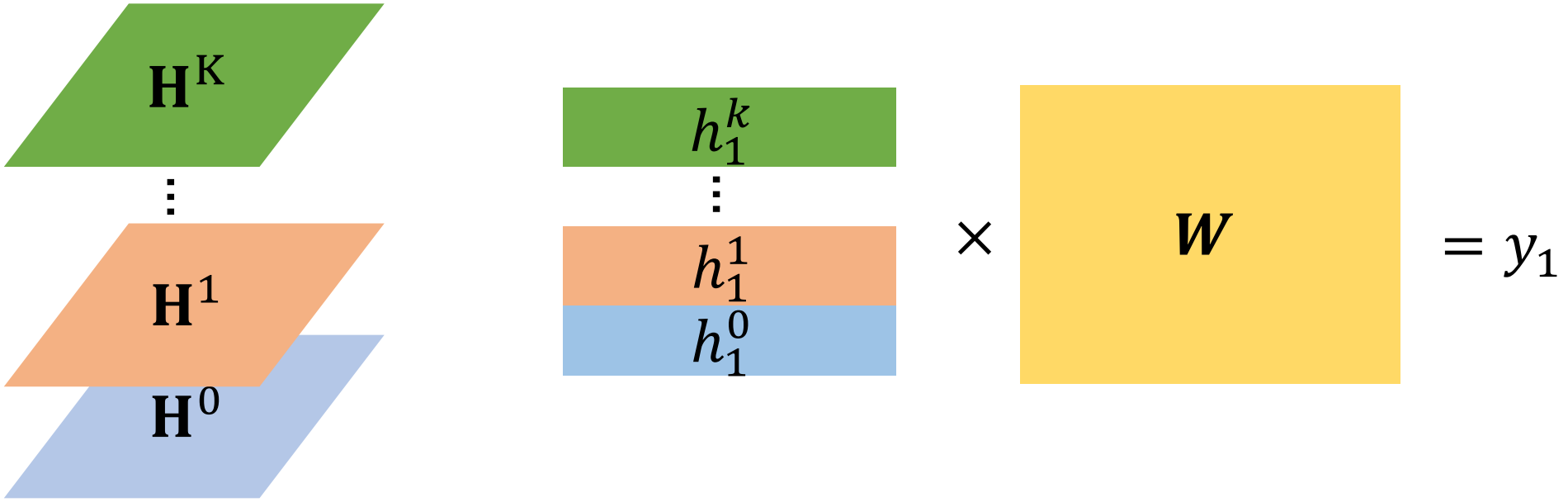




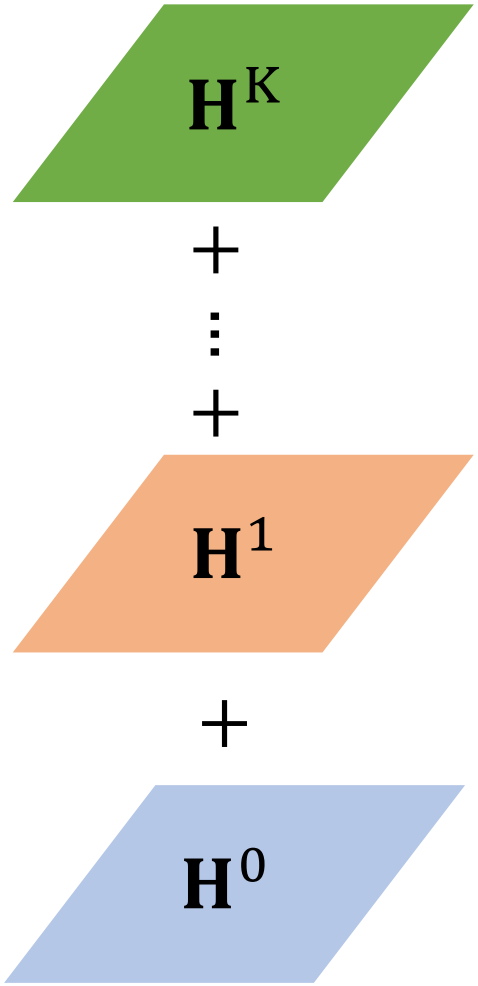
https://arxiv.org/abs/1511.02136

DCNN (Diffusion-Convolution Neural Network)

✓ Node features



DGC (Diffusion Graph Convolution)

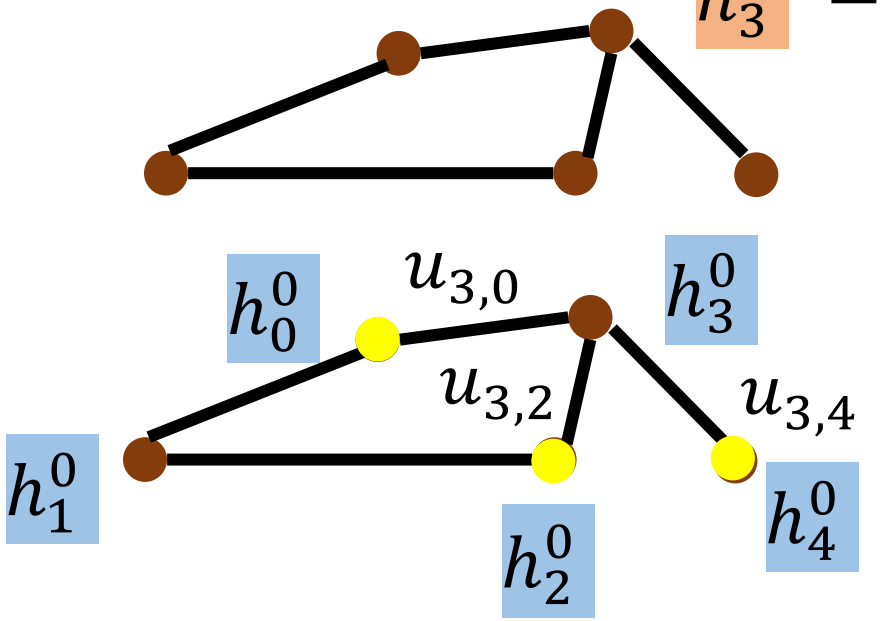


Published as a conference paper at ICLR 2018

https://arxiv.org/pdf/1707.01926.pdf

MoNET (Mixture Model Networks)

- ✓ Define a measure on node ‘distances’
- ✓ Use weighted sum (mean) instead of simply summing up (averaging) neighbor features.



$$h_3^1 = w(\hat{u}_{3,0}) \times h_0^0 + w(\hat{u}_{3,2}) \times h_2^0 + w(\hat{u}_{3,2}) \times h_4^0$$

$w(\cdot)$ is a NN, $\hat{u}$ is a transform from u

$$\mathbf{u}(x,y) = \left(\frac{1}{\sqrt{\deg(x)}}, \frac{1}{\sqrt{\deg(y)}}\right)^\top$$

$$\sum_{y \in \mathcal{N}(x)} e^{-\frac{1}{2}(\tilde{\mathbf{u}}(x,y)-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\tilde{\mathbf{u}}(x,y)-\boldsymbol{\mu}_j)} f_l(y)$$

https://arxiv.org/pdf/1611.08402.pdf

MoNET

- ✓ Experiment

Method	Cora	PubMed
DCNN [2]	76.80 ± 0.60%	73.00 ± 0.52%
GCN [26]	81.59 ± 0.42%	78.72 ± 0.25%
MoNet	**81.69** ± 0.48%	**78.81** ± 0.44%

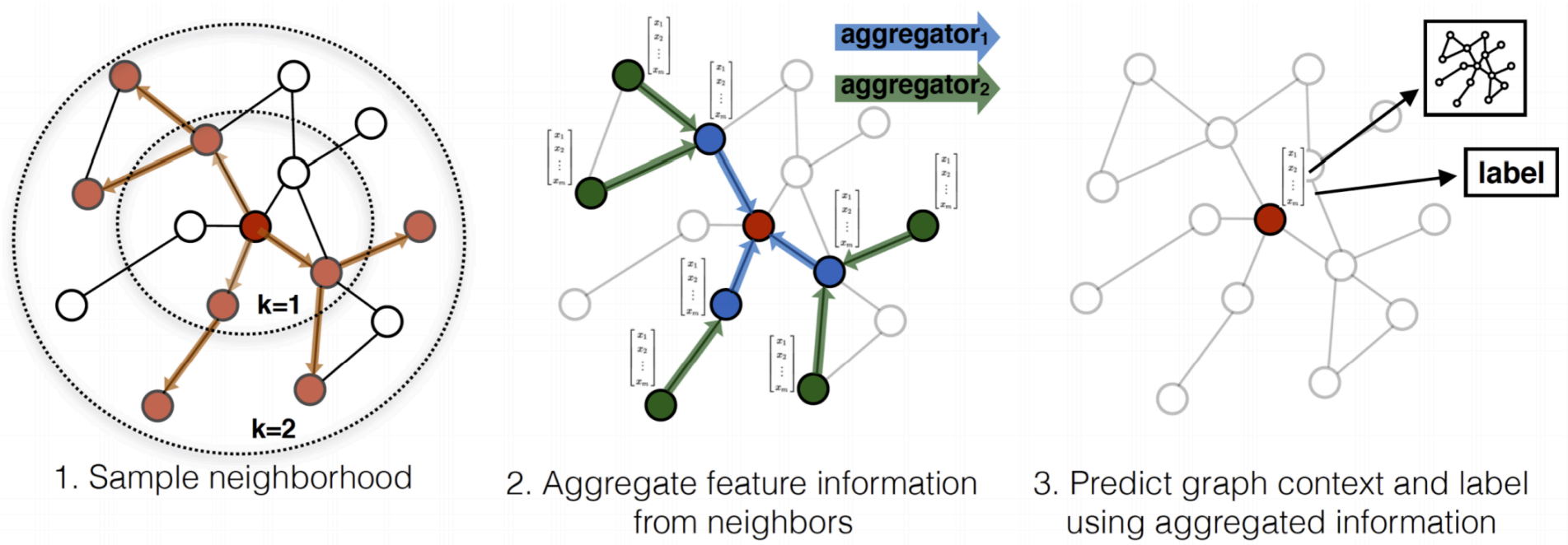
GraphSAGE

- ✓ SAmple and aggreGatE
- ✓ Can work on both transductive and inductive setting
- ✓ GraphSAGE learns how to embed node features from neighbors

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

GraphSAGE

✓ AGGREGATION: mean, max-pooling, or LSTM



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

https://arxiv.org/pdf/1706.02216.pdf

GraphSAGE

✓ Result

	Citation		Reddit		PPI	
Name	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1
GraphSAGE-GCN	0.742	0.772	**0.908**	0.930	0.465	0.500
GraphSAGE-mean	0.778	0.820	0.897	0.950	0.486	0.598
GraphSAGE-LSTM	0.788	0.832	**0.907**	**0.954**	0.482	**0.612**
GraphSAGE-pool	**0.798**	**0.839**	0.892	0.948	**0.502**	0.600
% gain over feat.	39%	46%	55%	63%	19%	45%

GAT (Graph Attention Networks)

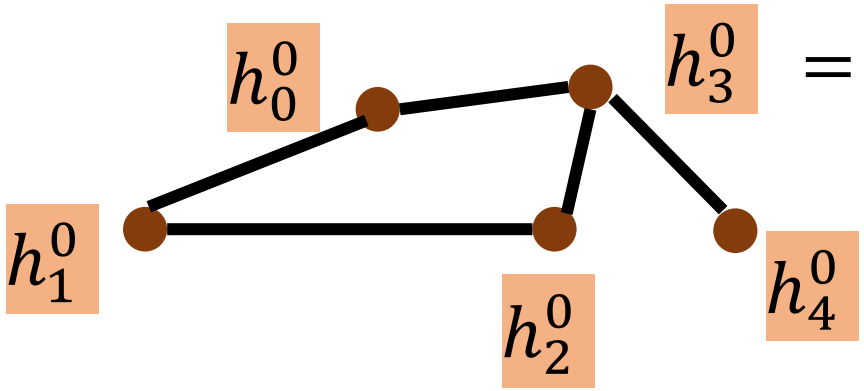
- ✓ Input: node features $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$
- ✓ Calculate energy: $e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$
- ✓ Attention score (over the neighbors)

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

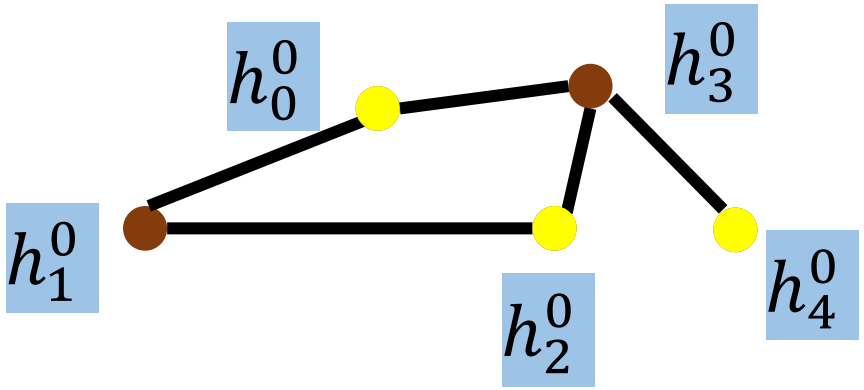
https://arxiv.org/pdf/1710.10903.pdf

Published as a conference paper at ICLR 2018

GAT (Graph Attention Networks)



$$h_3^0 = e_{3,0} \cdot h_0^0 + e_{3,2} \cdot h_2^0 + e_{3,4} \cdot h_4^0$$



$$f(h_3^0, h_0^0) = e_{3,0}$$

$$f(h_3^0, h_2^0) = e_{3,2}$$

$$f(h_3^0, h_4^0) = e_{3,4}$$

GAT (Graph Attention Networks)

✓ Experiment

Transductive

Method	Cora	Citeseer	Pubmed
MLP	55.1%	46.5%	71.4%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	81.7 ± 0.5%	—	78.8 ± 0.3%
GCN-64*	81.4 ± 0.5%	70.9 ± 0.5%	**79.0** ± 0.3%
GAT (ours)	**83.0** ± 0.7%	**72.5** ± 0.7%	**79.0** ± 0.3%

Inductive

Method	PPI
Random	0.396
MLP	0.422
GraphSAGE*	0.768
Const-GAT (ours)	0.934 ± 0.006
GAT (ours)	**0.973** ± 0.002

https://openreview.net/forum?id=ryGs6iA5Km

GIN (Graph Isomorphism Network)

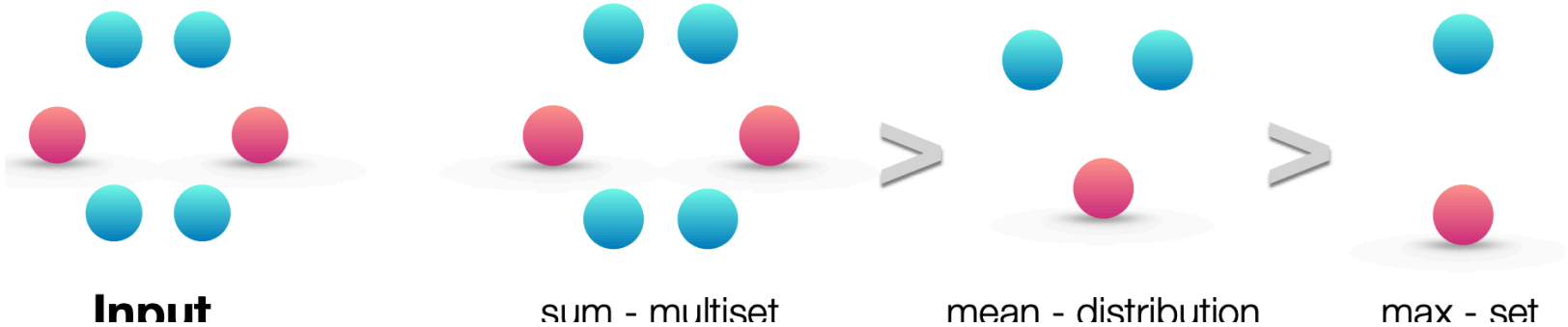
- ✓ A GNN can be at most as powerful as WL isomorphic test
- ✓ Theoretical proofs were provided

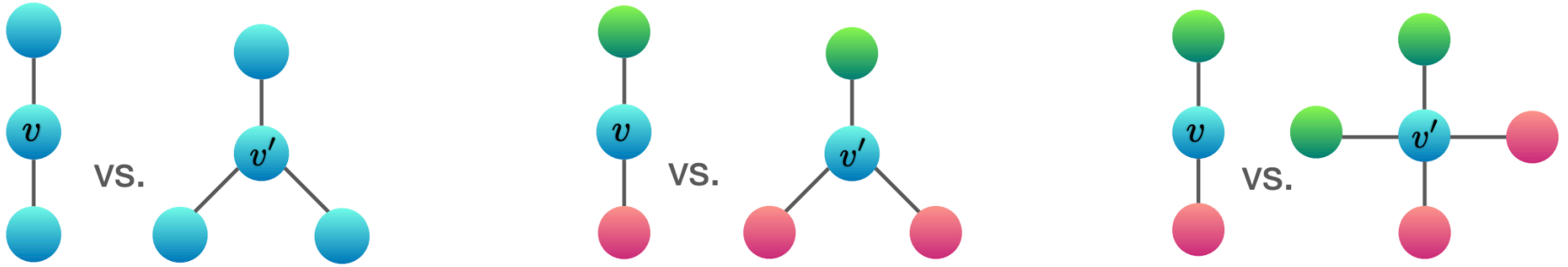
GIN (Graph Isomorphism Network)

✓ $h_v^{(k)} = \boxed{\mathrm{MLP}^{(k)}}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \boxed{\sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}}\right)$

✓ Sum instead of mean or max

✓ MLP instead of 1-layer





(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

GIN (Graph Isomorphism Network)

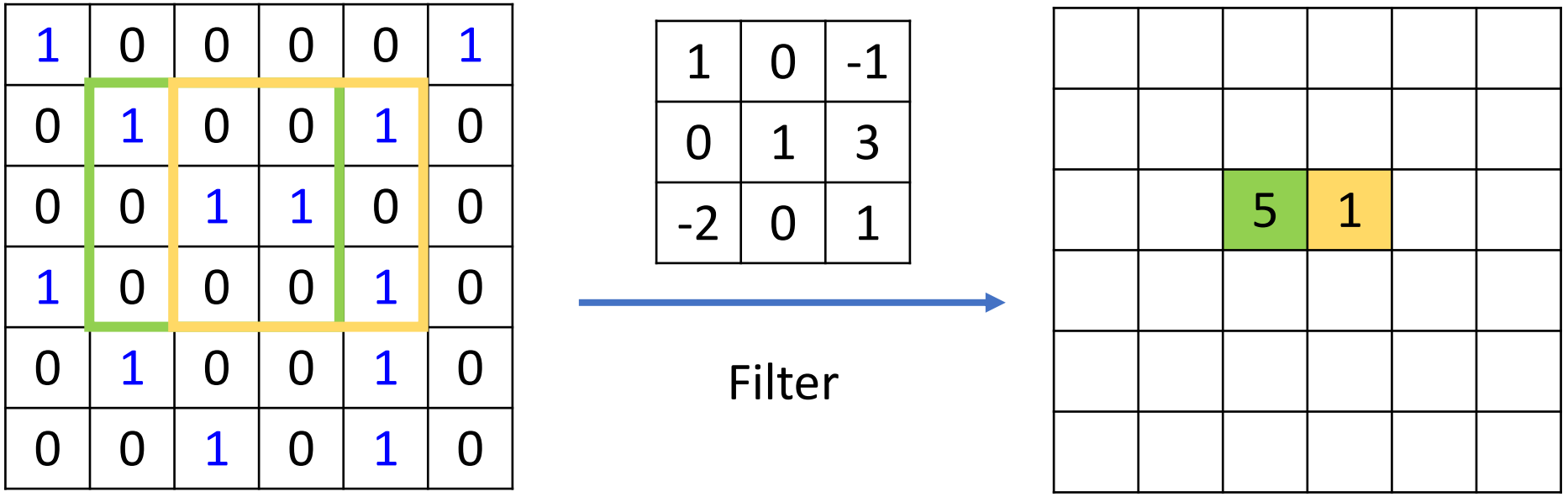
✓ Experiment

	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
GNN variants	SUM–MLP (**GIN-0**)	**75.1 ± 5.1**	**52.3 ± 2.8**	**92.4 ± 2.5**	**57.5 ± 1.5**	**80.2 ± 1.9**	**89.4 ± 5.6**	**76.2 ± 2.8**	**64.6 ± 7.0**	**82.7 ± 1.7**
	SUM–MLP (**GIN-ϵ**)	**74.3 ± 5.1**	**52.1 ± 3.6**	**92.2 ± 2.3**	**57.0 ± 1.7**	**80.1 ± 1.9**	**89.0 ± 6.0**	**75.9 ± 3.8**	63.7 ± 8.2	**82.7 ± 1.6**
	SUM–1-LAYER	74.1 ± 5.0	**52.2 ± 2.4**	90.0 ± 2.7	55.1 ± 1.6	**80.6 ± 1.9**	**90.0 ± 8.8**	**76.2 ± 2.6**	63.1 ± 5.7	82.0 ± 1.5
	MEAN–MLP	73.7 ± 3.7	**52.3 ± 3.1**	50.0 ± 0.0	20.0 ± 0.0	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	**66.6 ± 6.9**	80.9 ± 1.8
	MEAN–1-LAYER (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	MAX–MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	MAX–1-LAYER (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

Review: Convolution

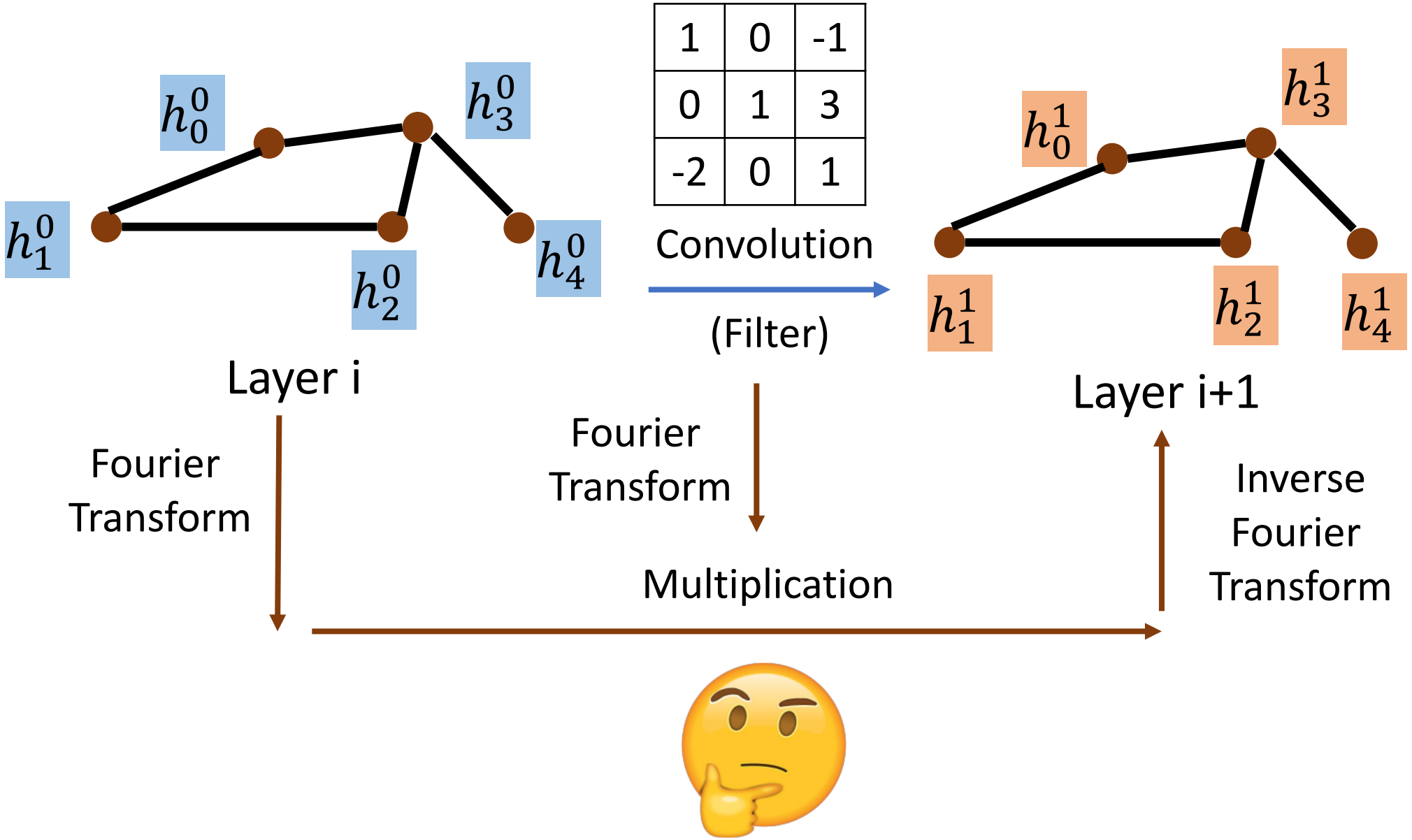


6 x 6 image with
3 x 3 kernel

Layer i

Layer i+1

Spectral-Based Convolution



Warning
of
~~Math (?)~~
Signal And System

Some of the following slides are from
Signal and System lectured by Prof. Lin Shan, Lee

N-dim Vector Space (P.31 of 1.0)

$$\vec{A} = \sum_{k=1}^{N} a_k \, \hat{v}_k \qquad \text{(合成)}$$

$$a_j = \vec{A} \cdot \hat{v}_j \qquad \text{(分析)}$$

$$\hat{v}_i \cdot \hat{v}_j = \delta_{ij}$$

Fourier Series Representation

- Fourier Series

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} a_k \phi_k(t)$$

$a_j \phi_j(t)$: j-th harmonic components

$-x(t)$ real

$$a_k^* = a_{-k}$$

$$x(t) = a_0 + 2\sum_{k=1}^{\infty} A_k \cos(k\omega_0 t + \theta_k), a_k = A_k e^{j\theta_k}$$

$$= a_0 + 2\sum_{k=1}^{\infty} [B_k \cos k\omega_0 t - C_k \sin k\omega_0 t], a_k = B_k + jC_k$$

Determination of a_k

$$\vec{A} \cdot \hat{v}_n = \left(\sum_k a_k \hat{v}_k\right) \cdot \hat{v}_n$$

$$\hat{v}_k \cdot \hat{v}_n = \begin{cases} T, k = n \\ 0, k \neq n \end{cases}$$

Not unit vector
orthogonal

$$\vec{A} \cdot \hat{v}_n = T a_n$$

$$a_n = \frac{1}{T}(\vec{A} \cdot \hat{v}_n) \quad \text{(分析)}$$

Signal Representation in Two Domains

Time Domain Basis

$$\vec{A} = \sum_i a_i \ \vec{v}_i$$

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)\,d\tau$$

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\,d\omega$$

$$= \sum_k \underset{b_k}{(\cdots)}\ \vec{u}_k$$

Frequency Domain Basis

$$\vec{A} = \sum_k b_k \ \vec{u}_k$$

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\,d\omega$$

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)\,d\tau$$

$$= \sum_i \underset{a_i}{(\cdots)}\ \vec{v}_i$$

Fourier transform

Basis (Eigenfuntion)

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$: spectrum, frequency domain

Fourier Transform

Inner-Product (分析)

Spectral Graph Theory

- ✓ Graph: $G = (V, E)$, $N = |V|$
- ✓ $A \in \mathbb{R}^{N \times N}$, adjacency matrix (weight matrix).

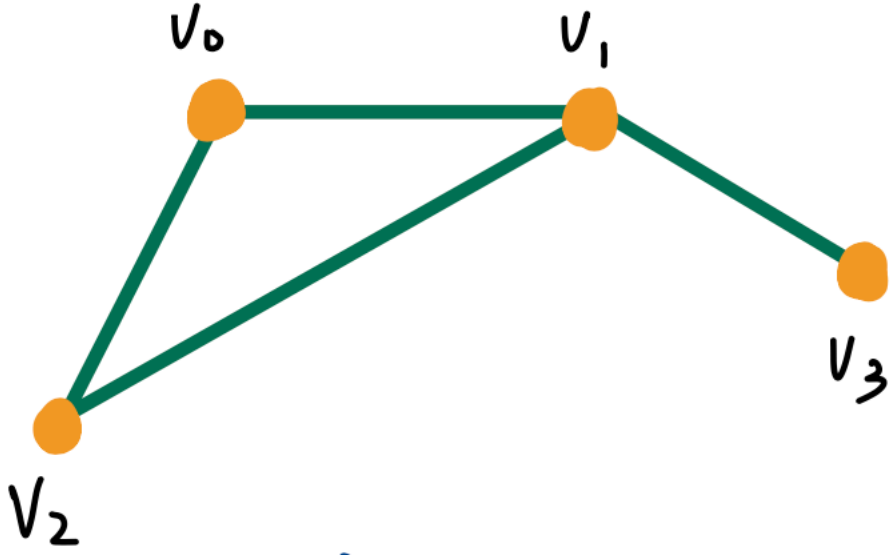
$$A_{i,j} = 0 \; if \; e_{i,j} \notin E, else \; A_{i,j} = w(i,j)$$

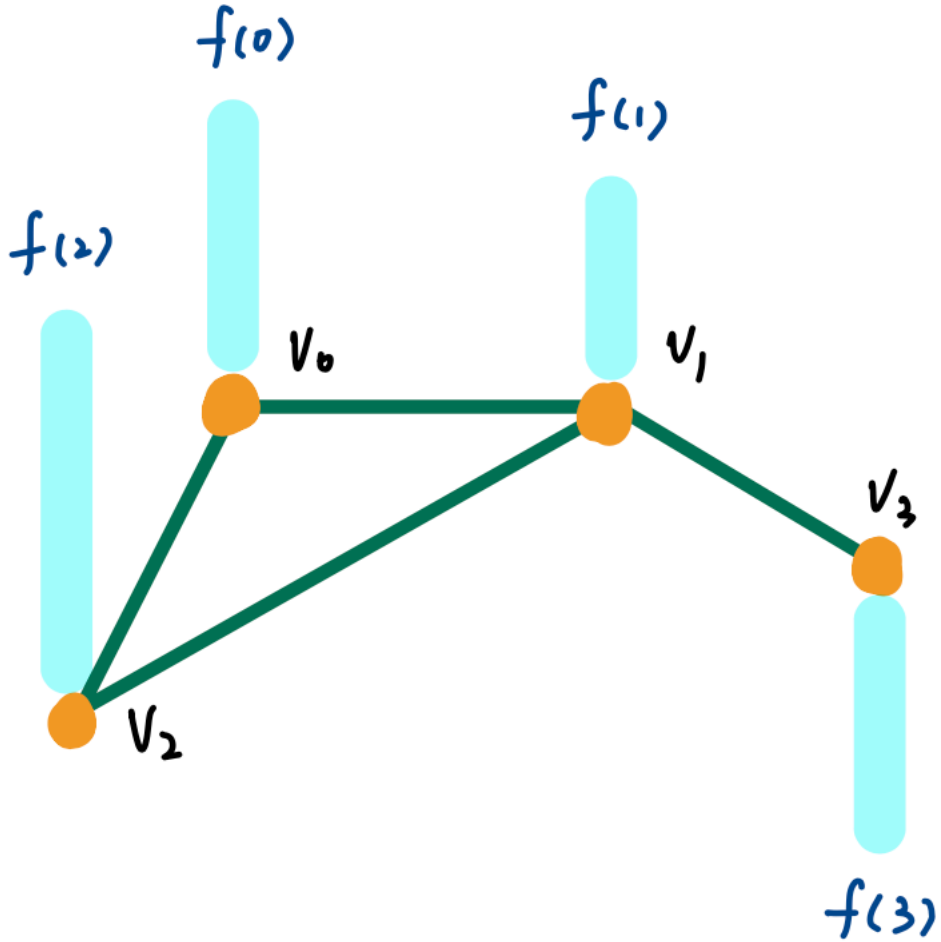
- ✓ We only consider undirected graph
- ✓ $D \in \mathbb{R}^{N \times N}$, degree matrix

$$D_{i,j} = \begin{cases} d(i) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$ (Sum of row i in A)

- ✓ $f: V \to \mathbb{R}^N$, signal on graph (vertices). $f(i)$ denotes the signal on vertex i

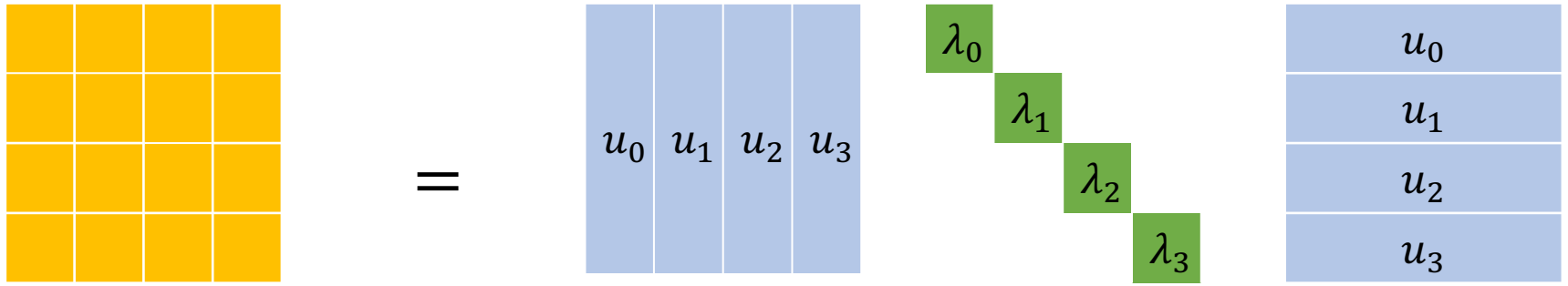
Spectral Graph Theory





Spectral Graph Theory

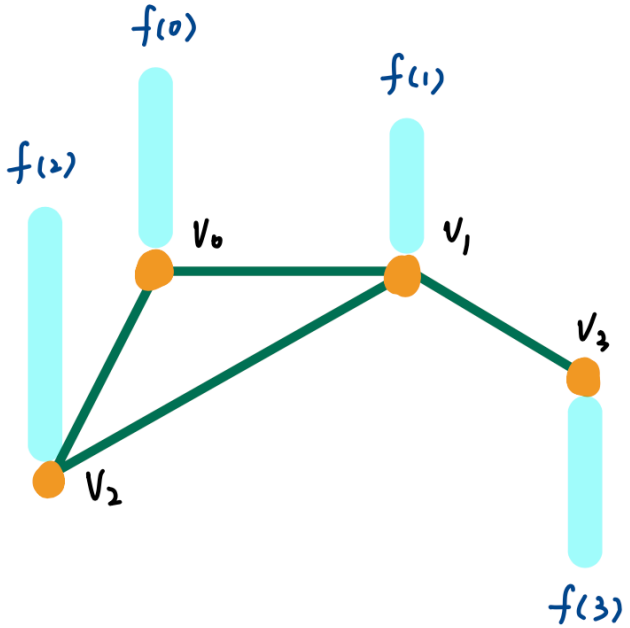
- ✓ Graph Laplacian $L = D - A,\ L \succcurlyeq 0$ (Positive semidefinite)
- ✓ L is symmetric (for undirected graph)
- ✓ $L = U\Lambda U^{\mathrm{T}}$ (spectral decomposition)
- ✓ $\Lambda = \mathrm{diag}(\lambda_0, \dots, \lambda_{N-1}) \in \mathbb{R}^{N\times N}$
- ✓ $U = [u_0, \dots, u_{N-1}] \in \mathbb{R}^{N\times N}$, orthonormal
- ✓ λ_l is the frequency, u_l is the basis corresponding to λ_l



$L = U \Lambda U^{\mathrm{T}}$

Spectral Graph Theory

✓ Vertex domain signal



$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

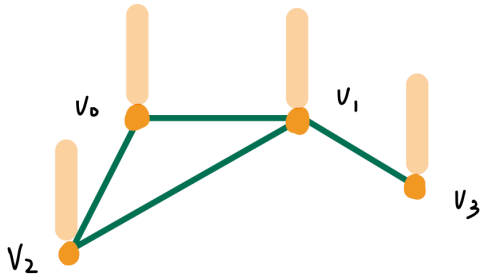
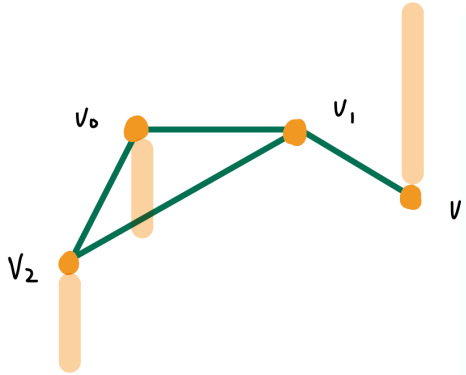
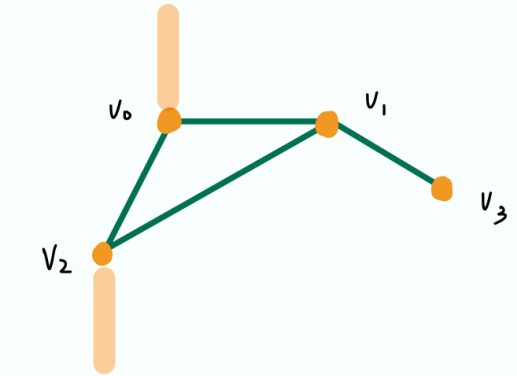
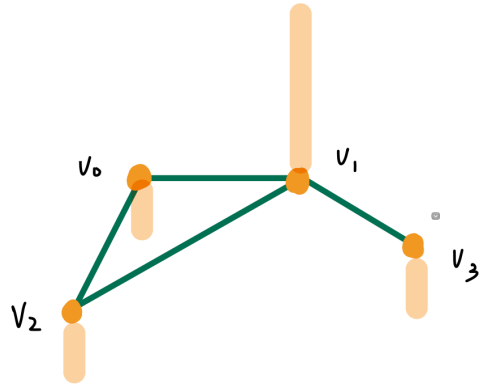
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$f = \begin{bmatrix} 4 \\ 2 \\ 4 \\ -3 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.5 & -0.41 & 0.71 & -0.29 \\ 0.5 & 0 & 0 & 0.87 \\ 0.5 & -0.41 & -0.71 & -0.29 \\ 0.5 & 0.82 & 0 & -0.29 \end{bmatrix}$$

Spectral Graph Theory

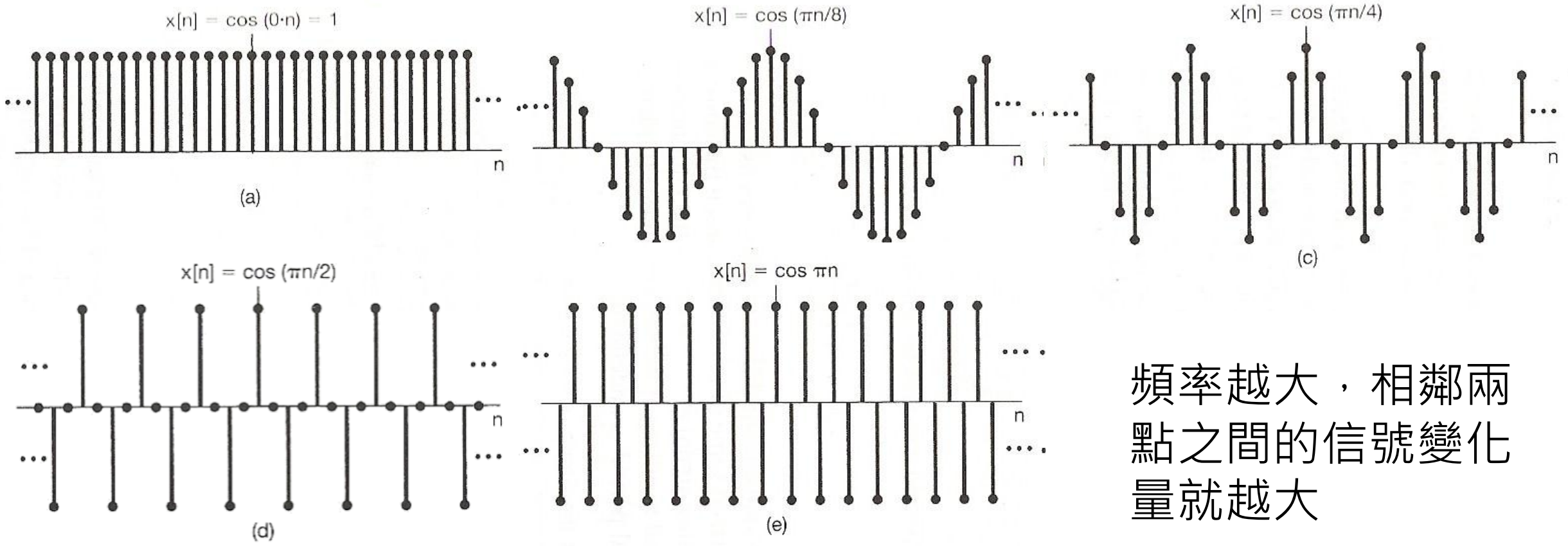
Low frequency → High frequency

λ	0	1	3	4
u	$[0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T$	$[-0.41 \quad 0 \quad -0.41 \quad 0.82]^T$	$[0.71 \quad 0 \quad -0.71 \quad 0]^T$	$[-0.29 \quad 0.87 \quad -0.29 \quad -0.29]^T$
				

(DC component)

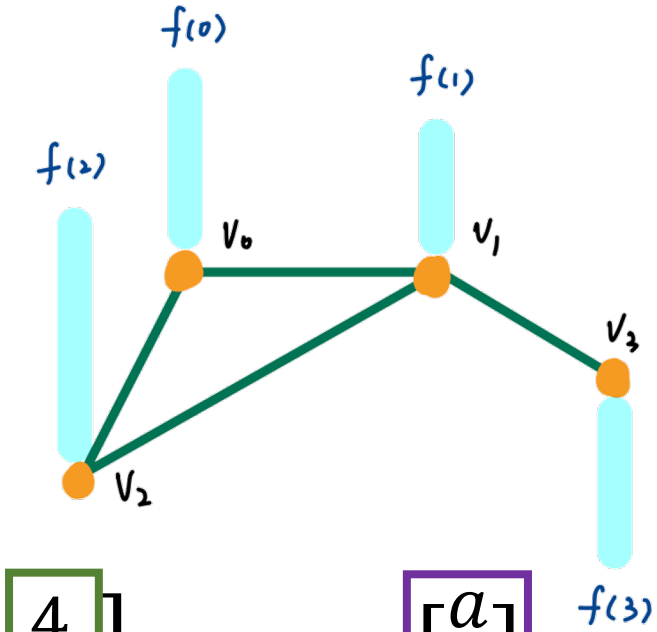
Spectral Graph Theory

- ✓ Discrete time Fourier basis



頻率越大，相鄰兩點之間的信號變化量就越大

Spectral Graph Theory



- ✓ Interpreting vertex frequency
  - ➢ L as an operator on graph
  - ➢ Given a graph signal f, what does Lf mean?
  - ➢ $Lf = (D - A)f = Df - Af$

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad f = \begin{bmatrix} 4 \\ 2 \\ 4 \\ -3 \end{bmatrix} \quad Lf = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

  - ➢ Lets focus on the first row of Lf
  - ➢ $a = [2\ 0\ 0\ 0] \cdot [4\ 2\ 4\ -3] - [0\ 1\ 1\ 0] \cdot [4\ 2\ 4\ -3]$

# of neighbors of v_0 — Signal on v_0's neighbors

$$= 2 \times 4 - 2 - 4 = (4 - 2) + (4 - 4) = 2$$

Signal on v_0 — Sum of difference between v_0 and its neighbors

Spectral Graph Theory

- ✓ $(Lf)(v_i) = \sum_{v_j \in V} w_{i,j}(f(v_i) - f(v_j))$, where $w_{i,j}$ is the $(i,j)^{th}$ entry of A

$$f^T Lf = \sum_{v_i \in V} f(v_i) \sum_{v_j \in V} w_{i,j}(f(v_i) - f(v_j))$$

$$= \sum_{v_i \in V} \sum_{v_j \in V} w_{i,j}(f^2(v_i) - f(v_i)f(v_j))$$

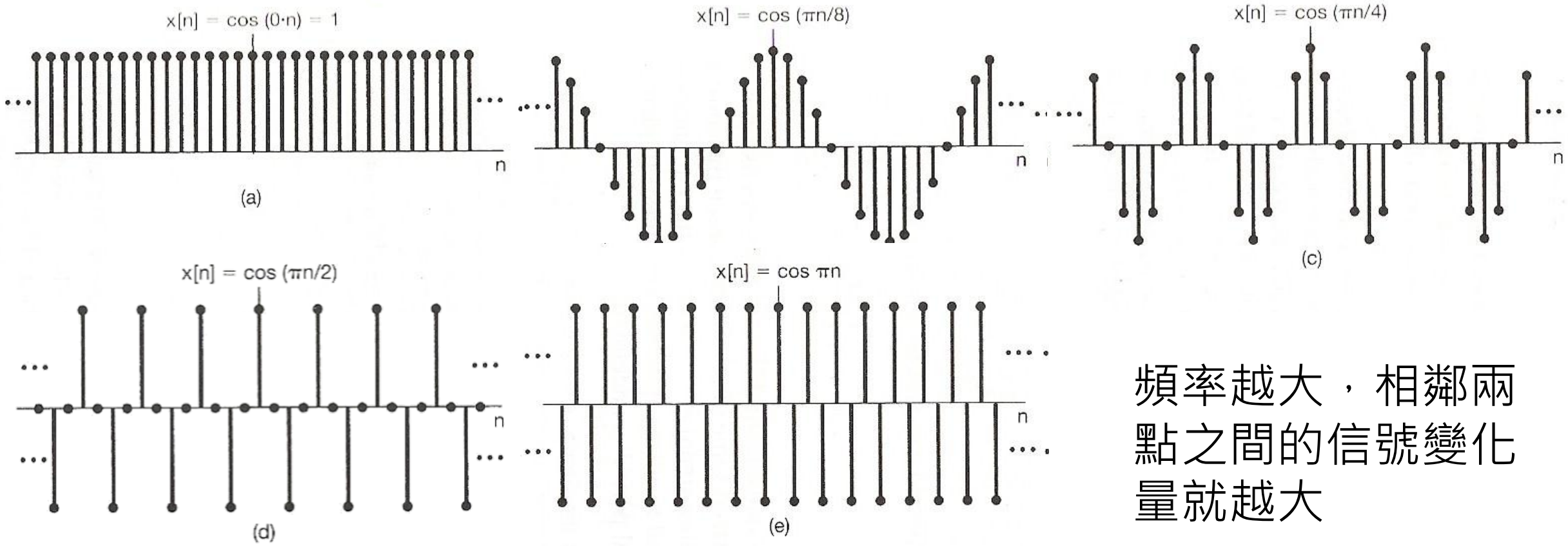
$$= \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in V} w_{i,j}(f^2(v_i) - f(v_i)f(v_j) + f^2(j) - f(v_j)f(v_i))$$

$$= \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in V} w_{i,j} \left(f(v_i) - f(v_j)\right)^2$$

"Power" of signal variation between nodes, i.e., smoothness of graph signal

Spectral Graph Theory

✓ Discrete time Fourier basis



頻率越大，相鄰兩點之間的信號變化量就越大

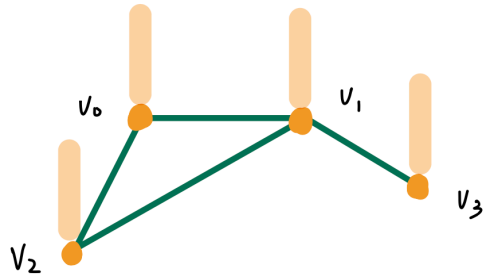
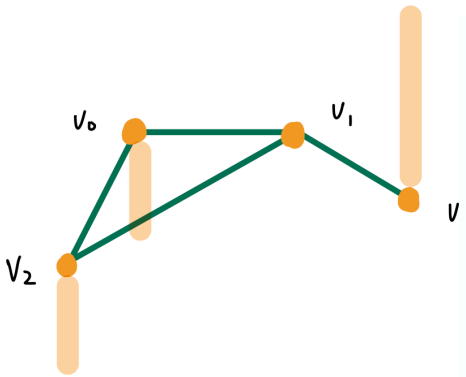
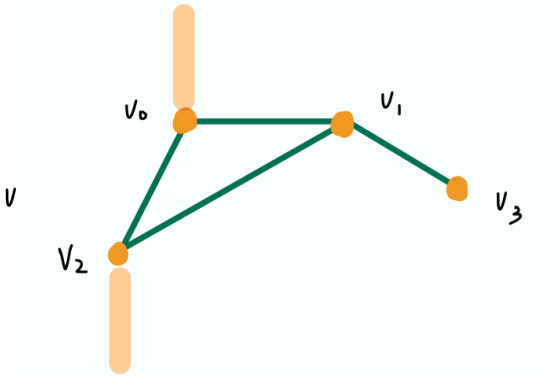
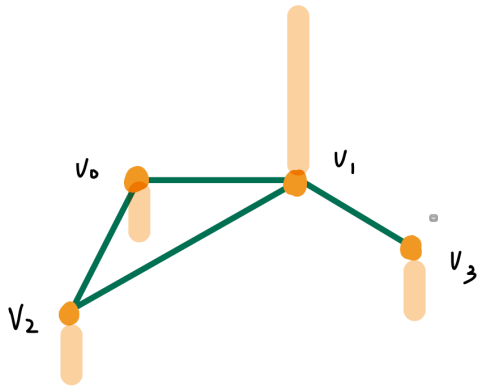
✓ $f^T L f$ represents "power" of signal variation between nodes

Frequency of f?

Spectral Graph Theory

- ✓ $u_i^T L u_i = u_i^T \lambda_i u_i = \lambda u_i^T u_i = \lambda_i$
- ✓ The eigenvectors corresponding to small λ belong to the low-pass part of a graph signal

Low frequency → High frequency

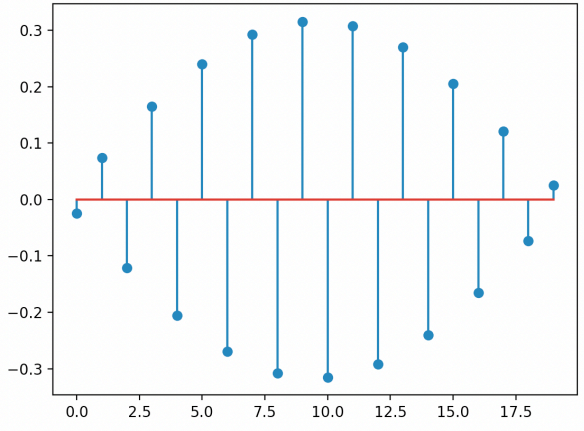
λ	0	1	3	4
u	$[0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T$	$[-0.41 \quad 0 \quad -0.41 \quad 0.82]^T$	$[0.71 \quad 0 \quad -0.71 \quad 0]^T$	$[-0.29 \quad 0.87 \quad -0.29 \quad -0.29]^T$
				

(DC component)

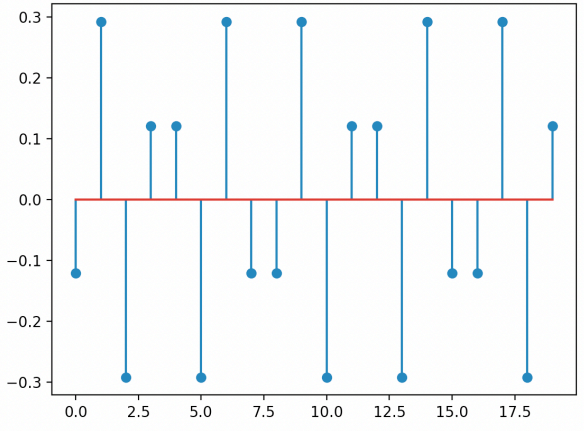
Spectral Graph Theory

- ✓ A special example: a line graph with 20 nodes

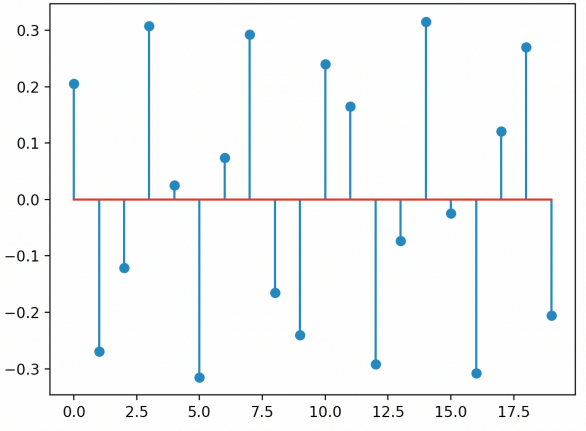
$\lambda = 3.97$



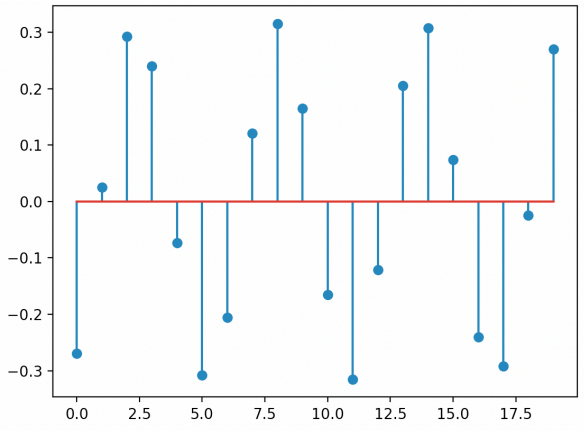
$\lambda = 3.41$



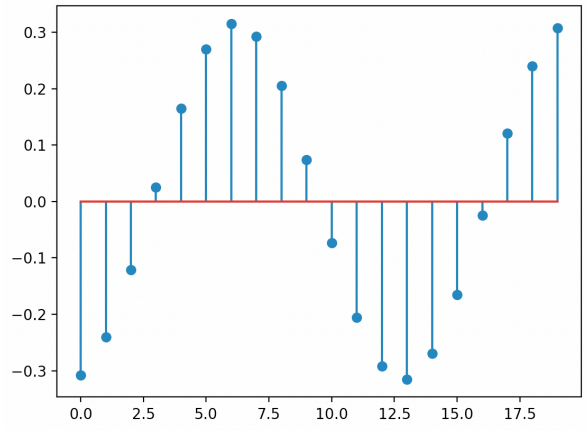
$\lambda = 2.31$



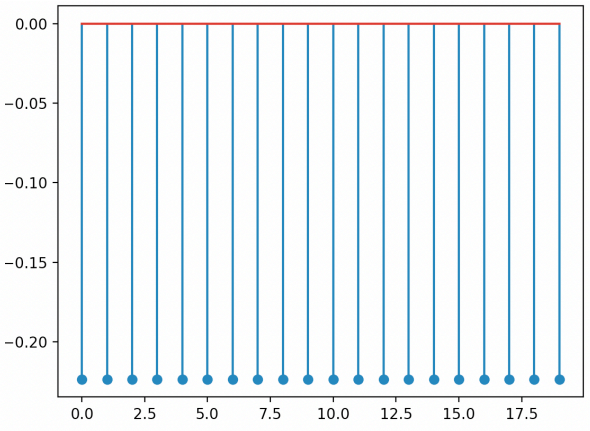
$\lambda = 1.10$



$\lambda = 0.22$

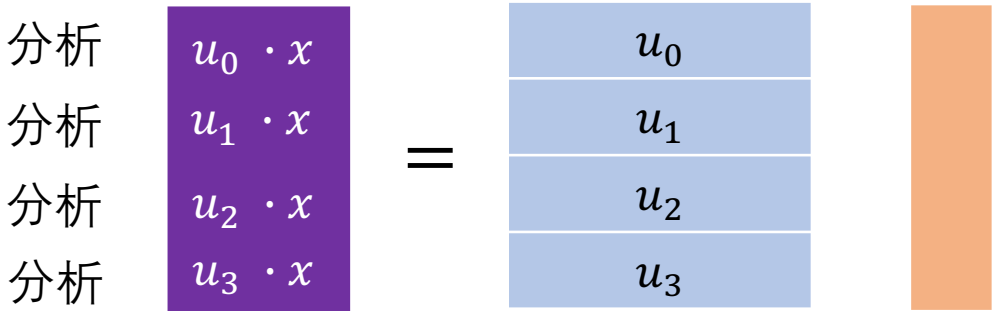


$\lambda = 0$



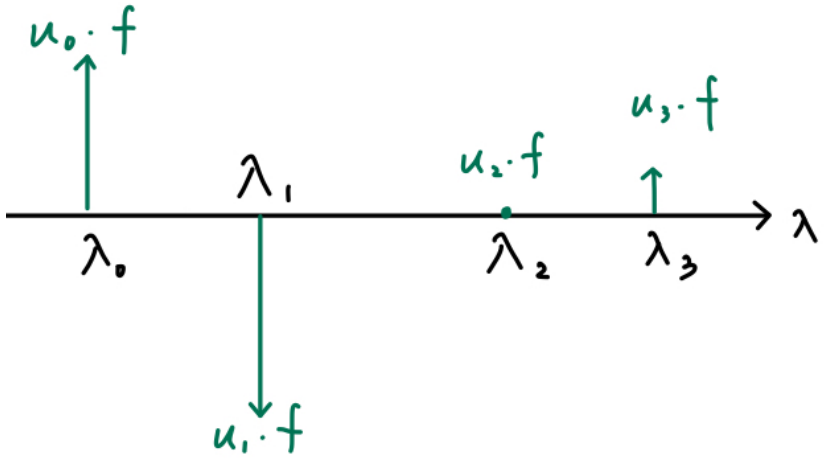
Spectral Graph Theory

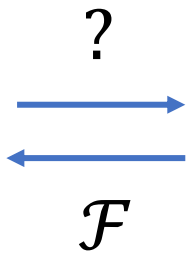
✓ Graph Fourier Transform of signal x: $\hat{x} = U^T x$

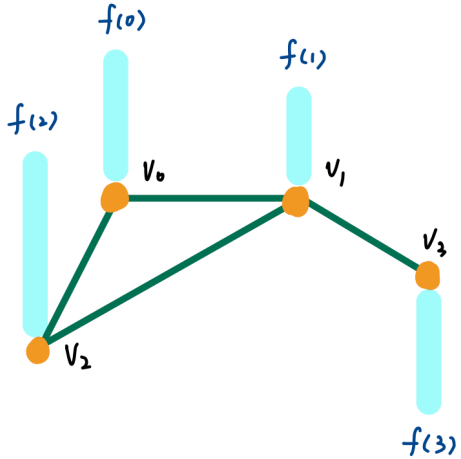


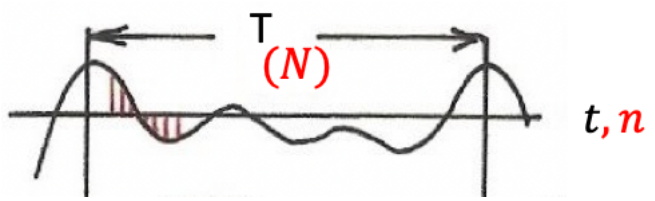
$\hat{x}$ = U^{T} x 合成

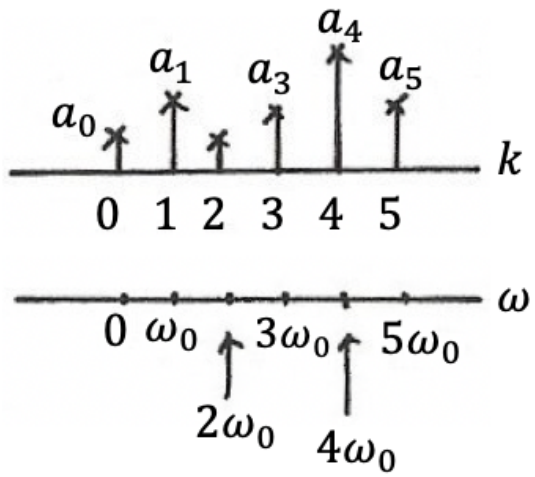
Spectral domain Vertex domain





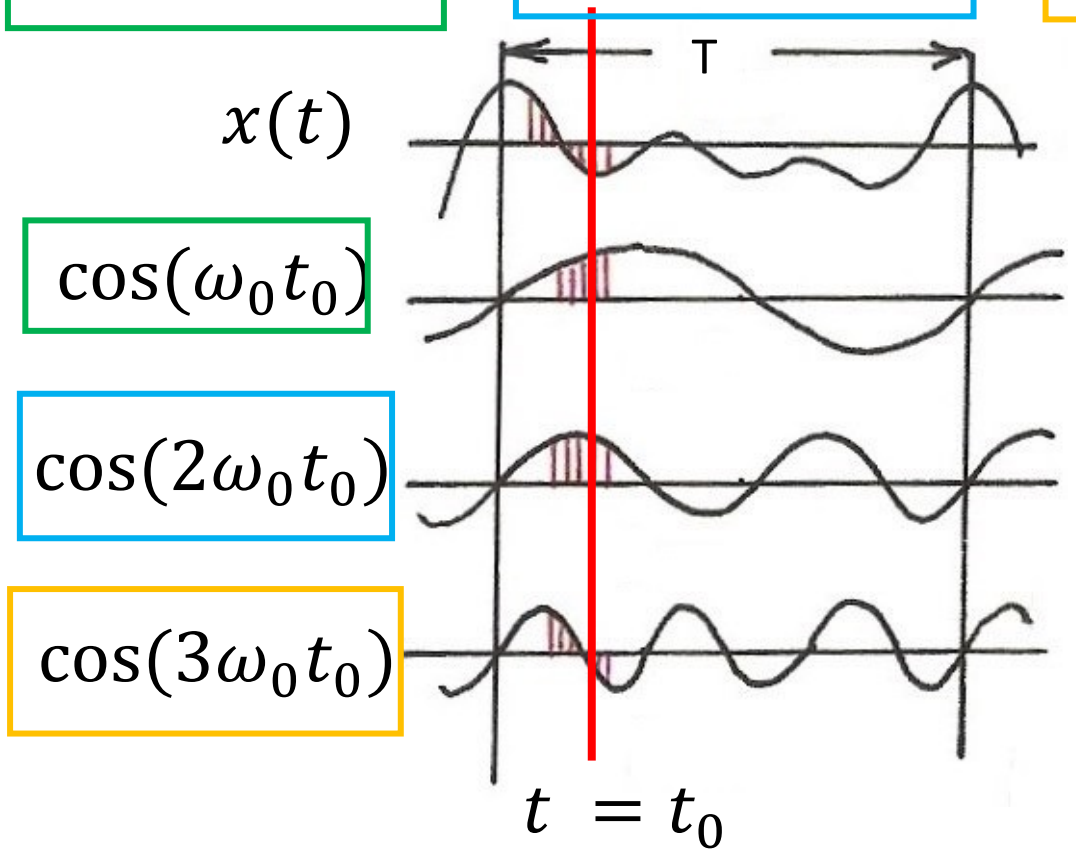






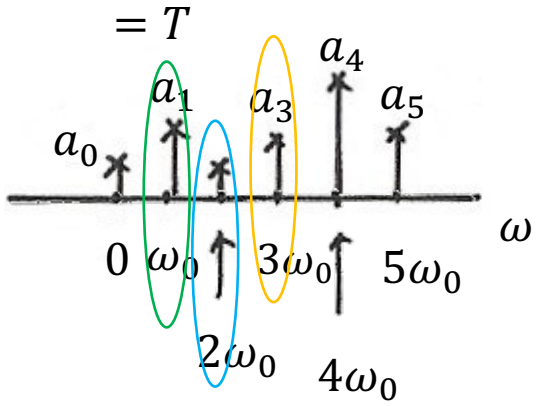
Harmonically Related Exponentials for Periodic Signals

$a_1 \cos(\omega_0 t_0) + a_2 \cos(2\omega_0 t_0) + a_3 \cos(3\omega_0 t_0) + \ldots$



$V = \{x(t) | x(t)$ periodic, fundamental period $= T\}$

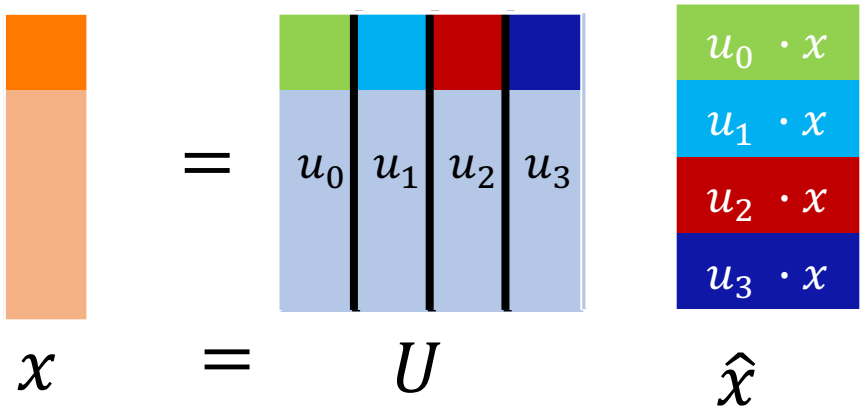
$\omega_0 = \frac{2\pi}{T}$



- All with period T: integer multiples of ω_0
- Discrete in frequency domain

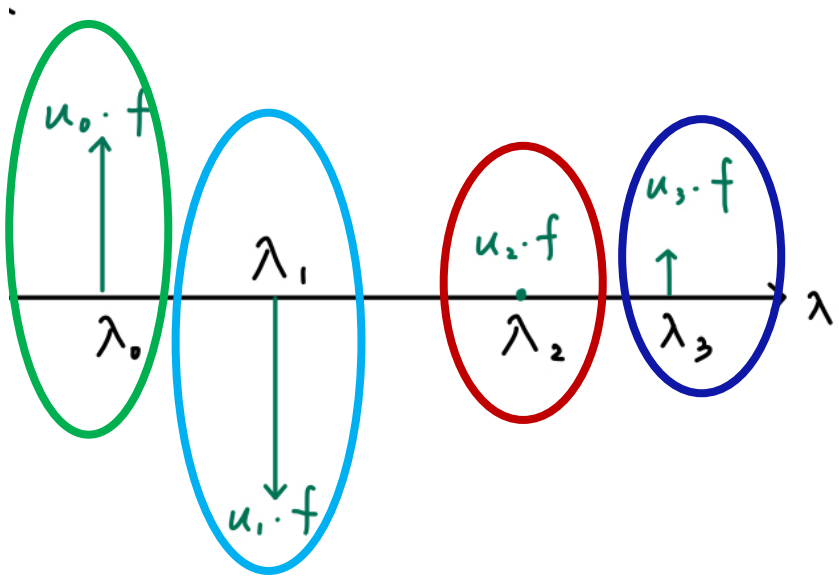
Spectral Graph Theory

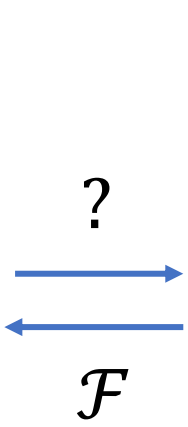
✓ Inverse Graph Fourier Transform of signal $\hat{x}$: $x = U\hat{x}$

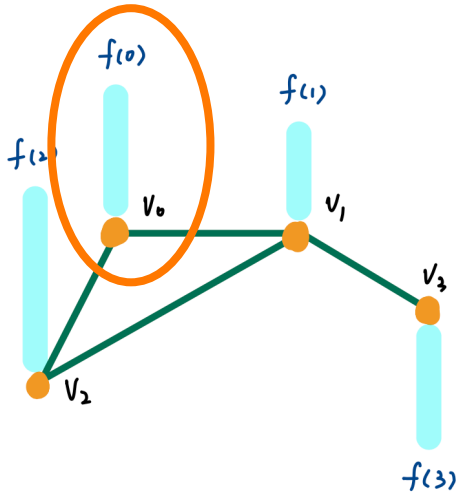


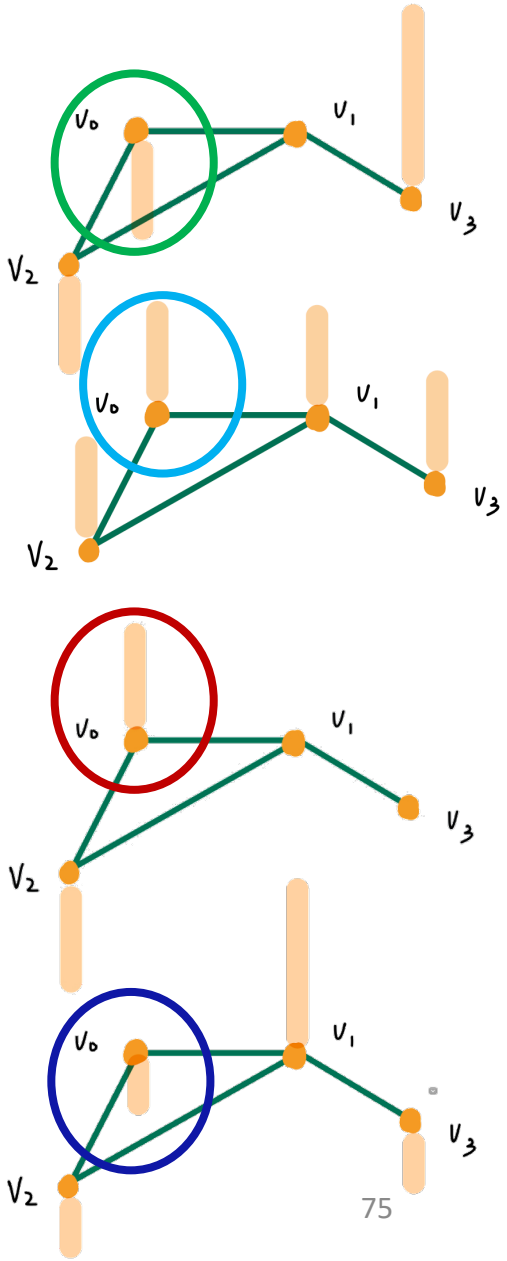
x = U $\hat{x}$ 合成

Vertex domain Spectral domain





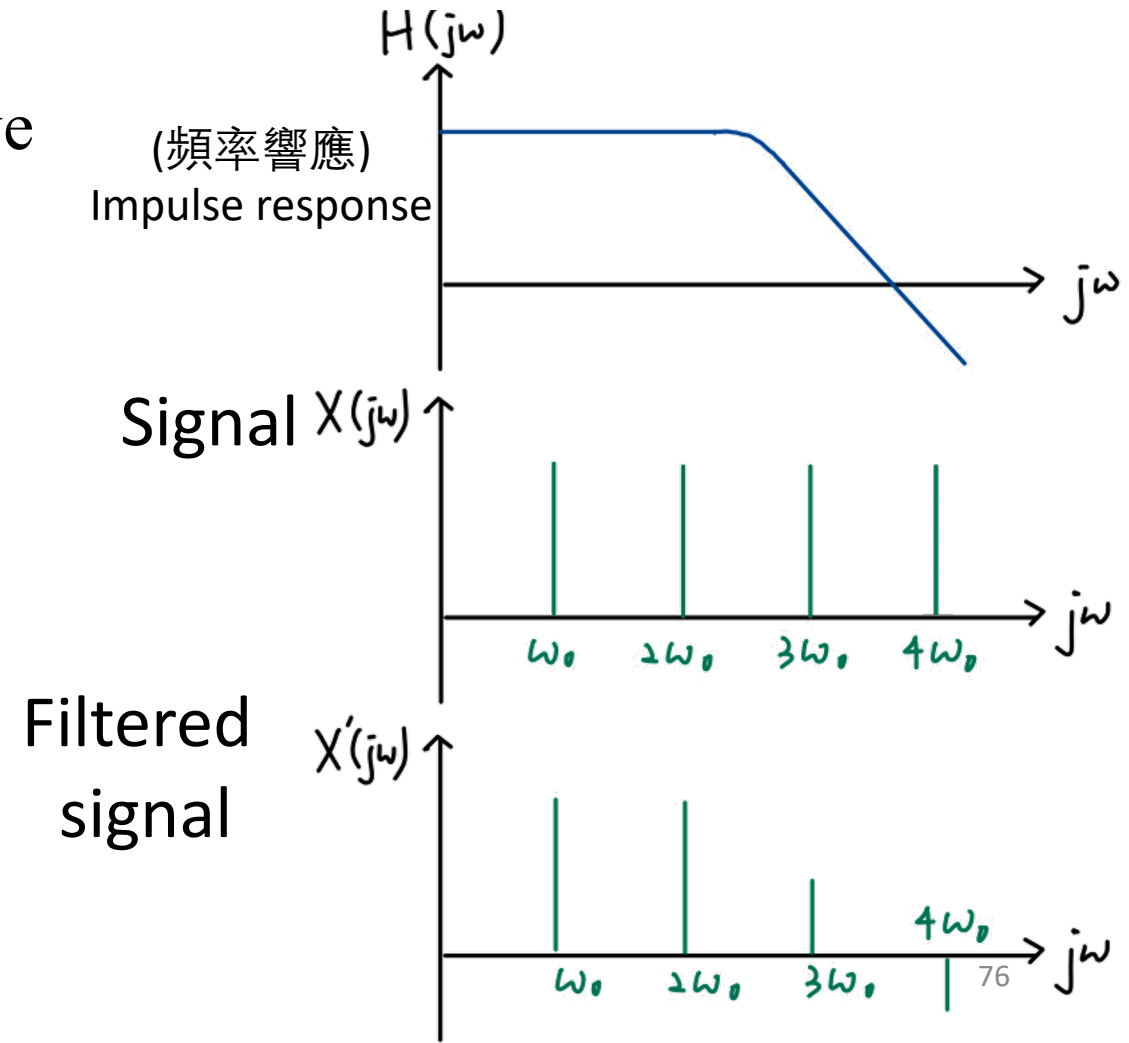




Filtering

modifying the amplitude/ phase of the different frequency components in a signal, including eliminating some frequency components entirely

– frequency shaping, frequency selective



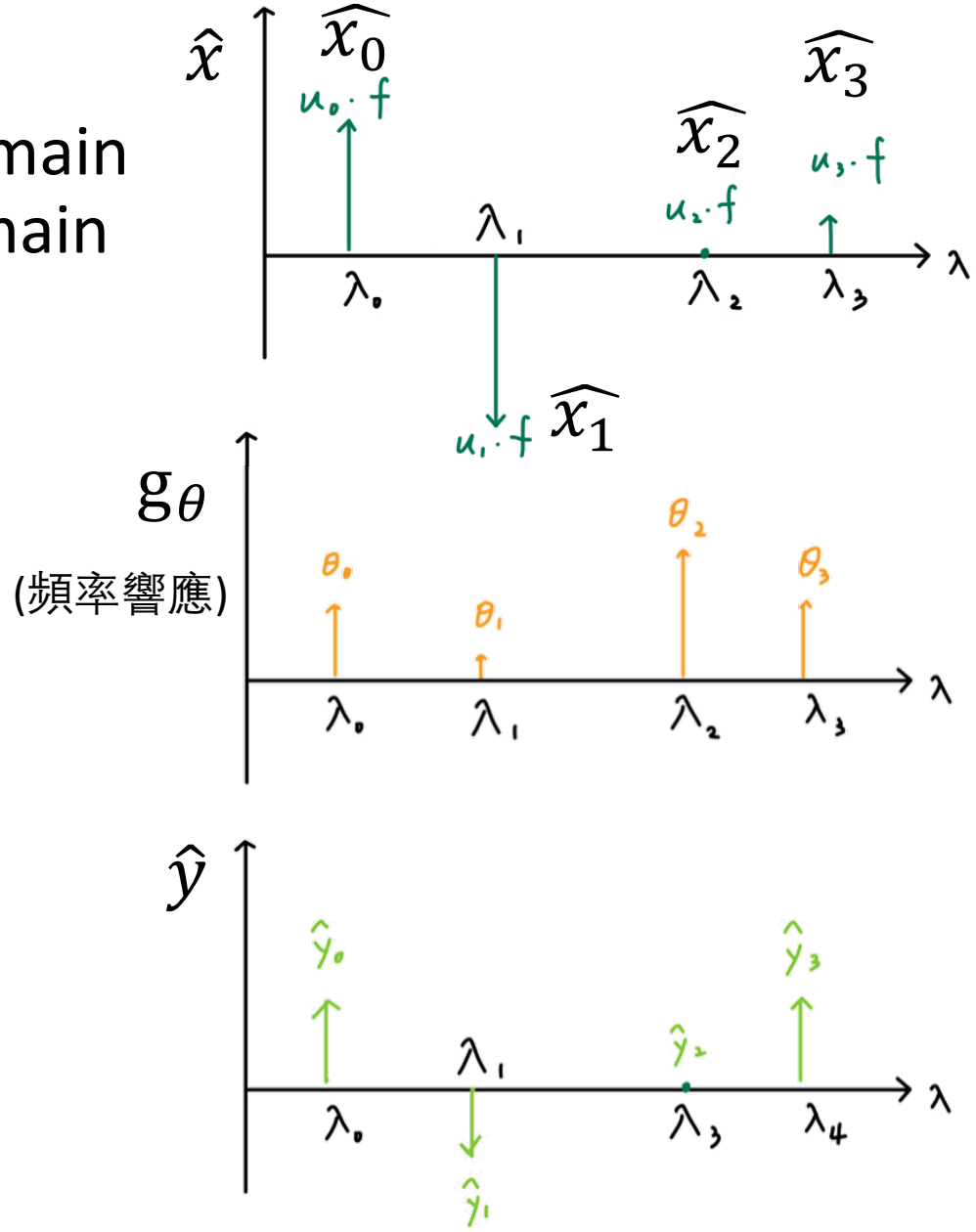
Spectral Graph Theory

- ✓ Filtering: Convolution in time domain is multiplication in frequency domain
- ✓ $\hat{y} = g_\theta(\Lambda)\,\hat{x}$

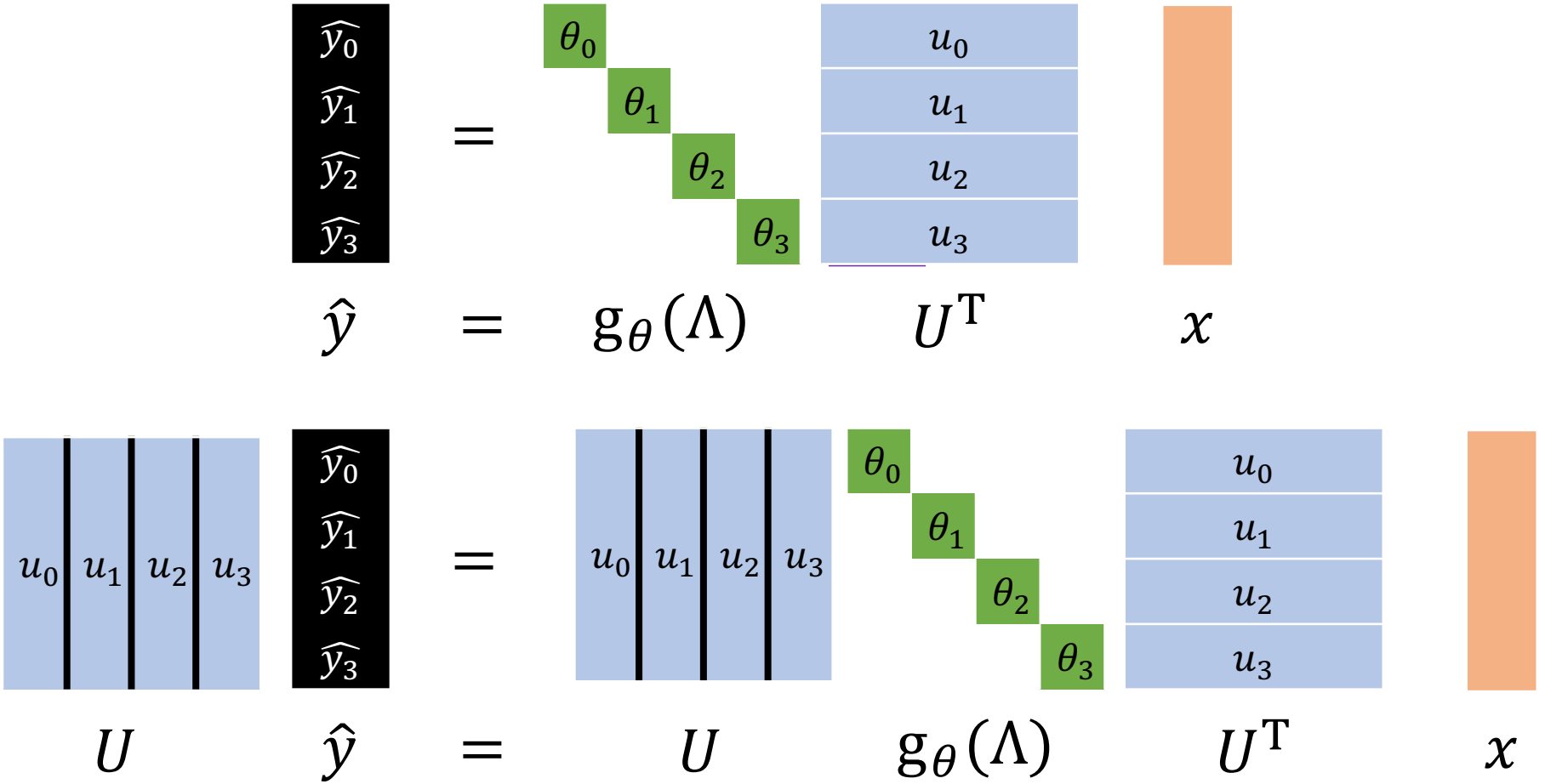
$$\begin{bmatrix} \widehat{y_0} \\ \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \end{bmatrix} = \begin{bmatrix} \theta_0 & & & \\ & \theta_1 & & \\ & & \theta_2 & \\ & & & \theta_3 \end{bmatrix} \begin{bmatrix} \widehat{x_0} \\ \widehat{x_1} \\ \widehat{x_2} \\ \widehat{x_3} \end{bmatrix}$$

$$\hat{y} = g_\theta(\Lambda)\,\hat{x}$$

$$g_\theta(\lambda_i) = \theta_i$$

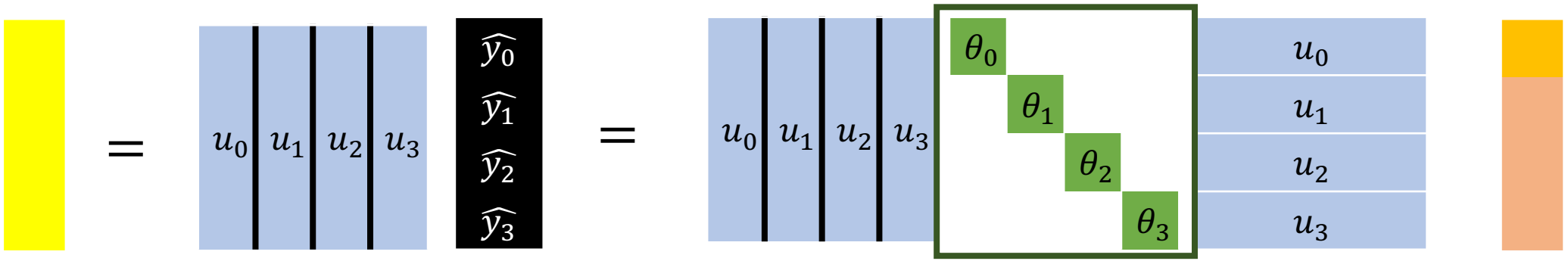


Spectral Graph Theory



Inverse Fourier Transform

Spectral Graph Theory



y U $\hat{y}$ $= U \; \mathrm{g}_\theta(\Lambda) \; U^{\mathrm{T}} x$

$= \mathrm{g}_\theta\left(U \Lambda U^{\mathrm{T}}\right) x$

$= \mathrm{g}_\theta(L) x$

$y = \mathrm{g}_\theta(L) x$

Parameters to be learnt

$\mathrm{g}_\theta(\cdot)$ can be any function. For example,

$$\mathrm{g}_\theta(L) = \log(\mathrm{I} + L) = L - \frac{L^2}{2} + \frac{L^3}{3} \ldots, \lambda_{max} < 1$$

Problem 1: Learning complexity is $O(N)$!!!

Spectral Graph Theory

✓ How about $g_\theta(L) = cos(L) = I - \frac{L^2}{2!} + \frac{L^4}{4!} \dots$

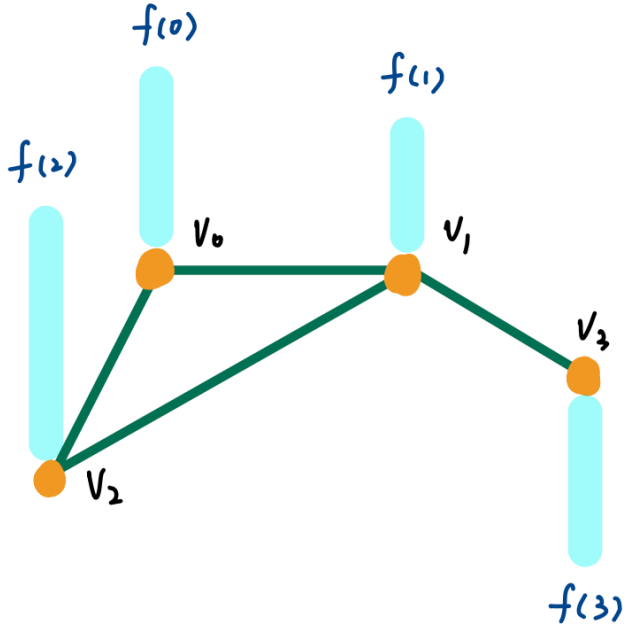
$$y = g_\theta(L)x$$

✓ What does this mean?

If $g_\theta(L) = L^2$, then $y = L^2 x$

$$\begin{bmatrix} 1 \\ 4 \\ 1 \\ -6 \end{bmatrix} = \begin{bmatrix} 6 & -4 & -3 & 1 \\ -4 & 12 & -4 & -4 \\ -3 & -4 & 6 & 1 \\ 1 & -4 & 1 & 2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 4 \\ -3 \end{bmatrix}$$

$$y = L^2 x$$



Lemma 3 *Let G be a weighted graph, and $\mathscr{L}$ the graph Laplacian of G. Fix an integer $s > 0$, and pick vertices m and n. Then $(\mathscr{L}^s)_{m,n} = 0$ whenever $d_G(m, n) > s$.*

https://hal.inria.fr/inria-00541855/document

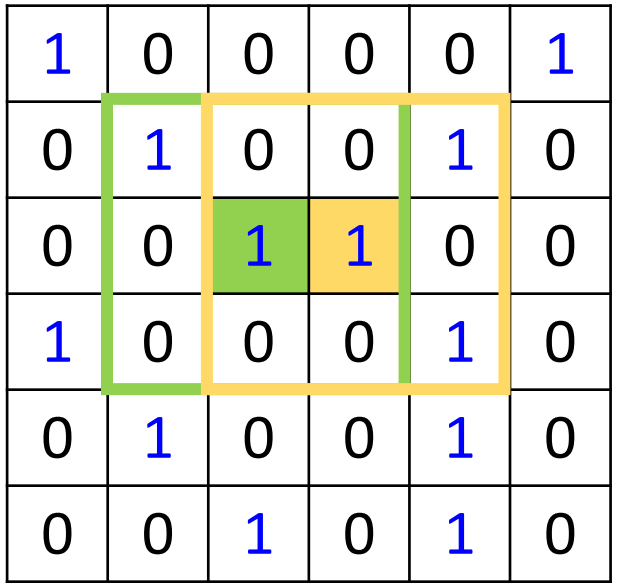
Spectral Graph Theory

- ✓ If we select $g_\theta(L) = cos(L) = I - \frac{L^2}{2!} + \frac{L^4}{4!} \ldots$
- ✓ If a connected graph has N nodes, then L^N will make the all nodes be able to share their signals with each other

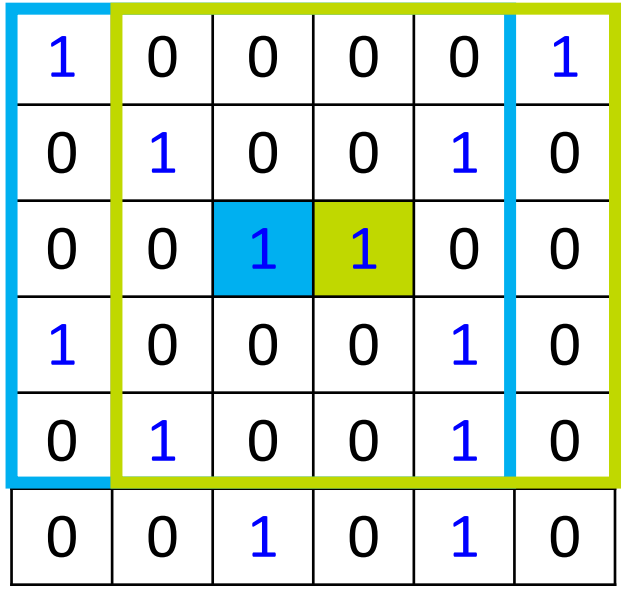
Problems 2: Not localize!!!

Review CNN

- ✓ Selecting different kernel size mean different receptive field
- ✓ We only consider local information within the receptive field



6 x 6 image with
3 x 3 kernel



6 x 6 image with
5 x 5 kernel

ChebNet

✓ Solution to Problem 1 and 2:

➢Use polynomial to parametrize $g_\theta(L)$

$$g_\theta(L) = \sum_{k=0}^{K} \theta_k L^k$$

Now it is K-localized

Parameters to be learnt: $O(K)$

$$g_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k$$

Problem 3:

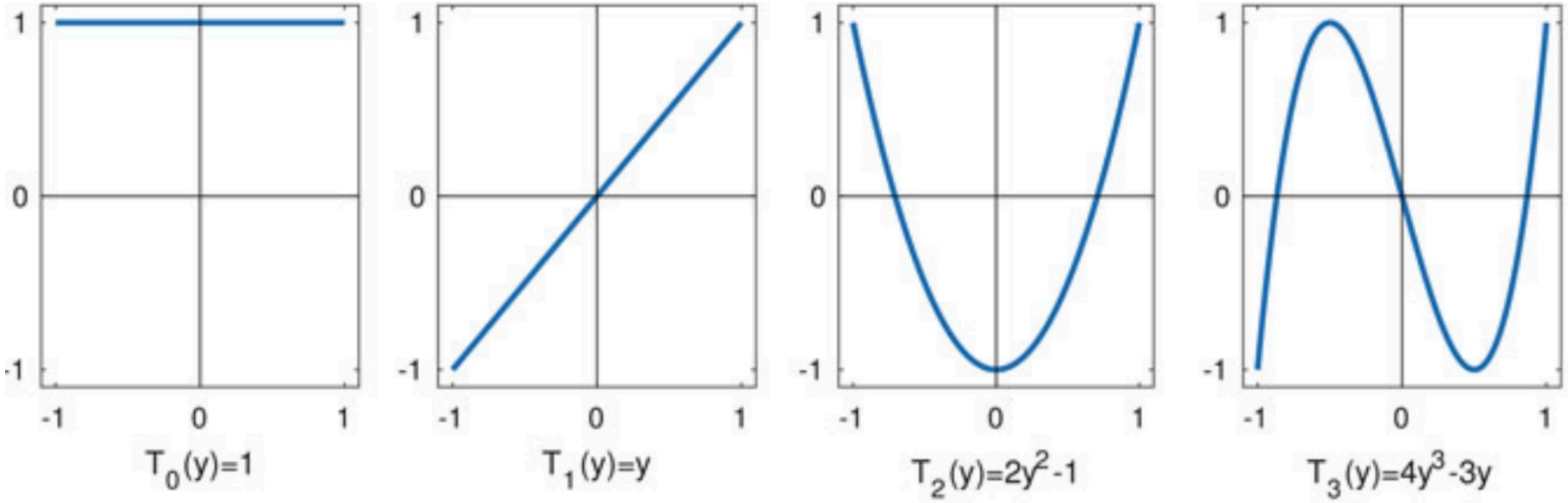
Time complexity: $O(N^2)$

$$y = U g_\theta(\Lambda) U^T x = U\left(\sum_{k=0}^{K} \theta_k \Lambda^k\right) U^T x$$

https://arxiv.org/pdf/1606.09375.pdf

ChebNet

- ✓ Solution to Problem 3:
  - ➢Use a polynomial function that can be computed recursively from L
- ✓ Chebyshev polynomial
  - ➢$T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), x \in [-1, 1]$



ChebNet

$$T_0(\tilde{\Lambda}) = \mathrm{I}, T_1(\tilde{\Lambda}) = \tilde{\Lambda}, T_k(\tilde{\Lambda}) = 2\tilde{\Lambda}T_{k-1}(\tilde{\Lambda}) - T_{k-2}(\tilde{\Lambda})$$

$$\text{where } \tilde{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - \mathrm{I}, \quad \tilde{\lambda} \in [-1, 1]$$

$$\mathrm{g}_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k \longrightarrow \mathrm{g}_{\theta'}(\tilde{\Lambda}) = \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

$$y = \mathrm{g}_{\theta'}(\tilde{L})x = \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x$$

ChebNet

$$g_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k \longrightarrow g_{\theta'}(\tilde{\Lambda}) = \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

〔例題 4〕（綜合除法的應用）

設多項式 $f(x)=3x^4-7x^3-2x^2+2x+18$

$=a(x-2)^4+b(x-2)^3+c(x-2)^2+d(x-2)+e$，

其中 a,b,c,d,e 皆為實數。

(1) 求 a,b,c,d,e 之值　(2) 求 $f(1.99)$ 的近似值到小數第三位。

~56~

建中數學學資第一冊

ChebNet

$$y = g_{\theta'}(L)x = \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x$$

$$= \theta'_0 T_0(\tilde{L})x + \theta'_1 T_1(\tilde{L})x + \theta'_2 T_2(\tilde{L})x + \cdots + \theta'_K T_K(\tilde{L})x$$

$$T_0(\tilde{L}) = \mathrm{I}, T_1(\tilde{L}) = \tilde{L}, T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L})$$

$$T_0(\tilde{L})x = x, T_1(\tilde{L})x = \tilde{L}x, T_k(\tilde{L})x = 2\tilde{L}T_{k-1}(\tilde{L})x - T_{k-2}(\tilde{L})\,x$$

Define $T_k(\tilde{L})x = \bar{x}_k$

$$\bar{x}_0 = x, \bar{x}_1 = \tilde{L}x, \bar{x}_k = 2\tilde{L}\bar{x}_{k-1} - \bar{x}_{k-2}$$

ChebNet

$$\bar{x}_0 = x, \bar{x}_1 = \tilde{L}x, \bar{x}_k = 2\tilde{L}\bar{x}_{k-1} - \bar{x}_{k-2}$$

$$y = g_{\theta'}(L)x = \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x$$

$$= \theta'_0 T_0(\tilde{L})x + \theta'_1 T_1(\tilde{L})x + \theta'_2 T_2(\tilde{L})x + \cdots + \theta'_K T_K(\tilde{L})x$$

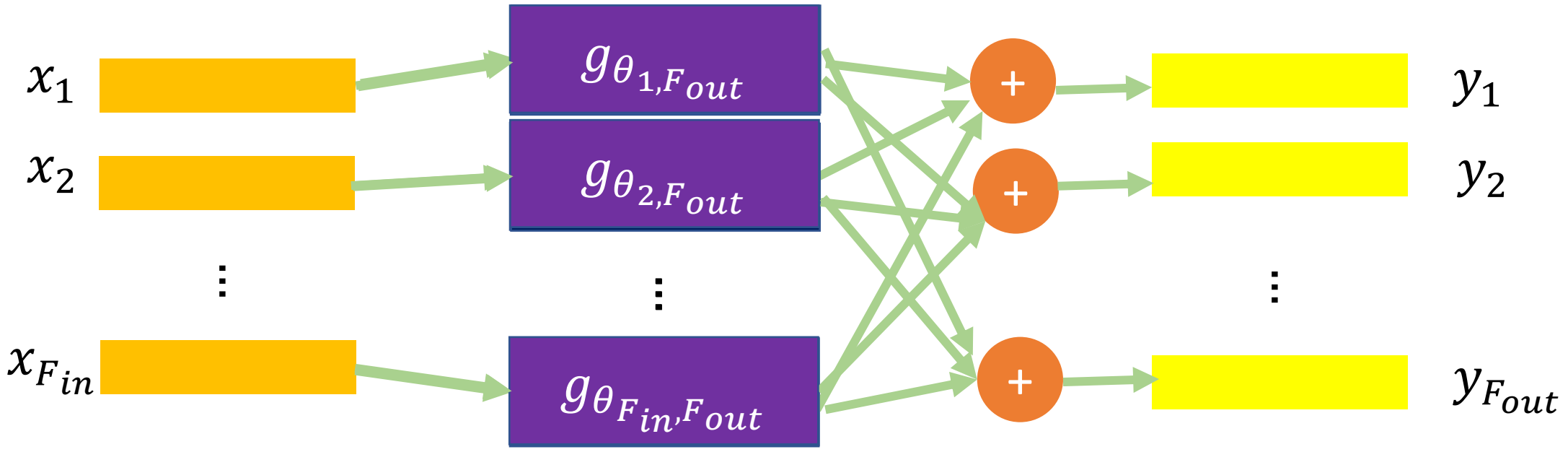
$$= \theta'_0 \bar{x}_0 + \theta'_1 \bar{x}_1 + \theta'_2 \bar{x}_2 + \cdots + \theta'_K \bar{x}_K$$

$$= [\bar{x}_0 \ \ \bar{x}_1 \ \ \ldots \ \ \bar{x}_K][\theta'_0 \ \ \theta'_1 \ \ \ldots \theta'_K]$$

Calculating $\bar{x}_k = 2\tilde{L}\bar{x}_{k-1} - \bar{x}_{k-2}$ cost $O(E)$

Total complexity: $O(KE)$

ChebNet


x_1
x_2
$x_{F_{in}}$
$g_{\theta_{1,F_{out}}}$
$g_{\theta_{2,F_{out}}}$
$g_{\theta_{F_{in},F_{out}}}$
+
+
+
y_1
y_2
$y_{F_{out}}$

GCN

https://openreview.net/pdf?id=SJU4ayYgl

✓ $y = g_{\theta'}(L)x = \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x \ , K = 1$

$$y = g_{\theta'}(L)x = \theta'_0 x + \theta'_1 \tilde{L} x \qquad \because \tilde{L} = \frac{2L}{\lambda_{max}} - I$$

$$= \theta'_0 x + \theta'_1 (\frac{2L}{\lambda_{max}} - I)x \qquad \because \lambda_{max} \approx 2$$

$$= \theta'_0 x + \theta'_1 (L - I)x \qquad \because L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$= \theta'_0 x - \theta'_1 (D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x \qquad \because \theta = \theta'_0 = -\theta'_1$$

$$= \theta (I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$$

renormalization trick: $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right)$$

GCN

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

Can be rewritten as:

$$h_v = f\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} Wx_u + b\right), \quad \forall v \in \mathcal{V}.$$

GCN

- ✓ Experiment: Semi-supervised classification

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

- ✓ Node features:
  - ➢ Citation network: Bag of Word of the document
  - ➢ In the knowledge base, each relation is denoted as a triplet (e_1, r, e_2)

GCN

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Comparison between the above GNNs

- ✓ Benchmark tasks

GRAPH NEURAL NETWORKS EXPONENTIALLY LOSE EXPRESSIVE POWER FOR NODE CLASSIFICATION

Kenta Oono[1, 2], **Taiji Suzuki**[1, 3]
{kenta_oono, taiji}@mist.i.u-tokyo.ac.jp
[1]The University of Tokyo
[2]Preferred Networks, Inc.
[3]RIKEN Center for Advanced Intelligence Project (AIP)

ICLR'20 spotlight (8, 6, 8)

https://arxiv.org/abs/1905.10947

Application to GCN

Theorem 2. *For any initial value* $X^{(0)}$*, the output of* l*-th layer* $X^{(l)}$ *satisfies* $d_{\mathcal{M}}(X^{(l)}) \leq (s\lambda)^l d_{\mathcal{M}}(X^{(0)})$*. In particular,* $d_{\mathcal{M}}(X^{(l)})$ *exponentially converges to* 0 *when* $s\lambda < 1$.

✓ 翻譯：如果$s\lambda < 1$，X 經過多層的 MLP 之後會落在subspace $\mathcal{M}$ 裡面

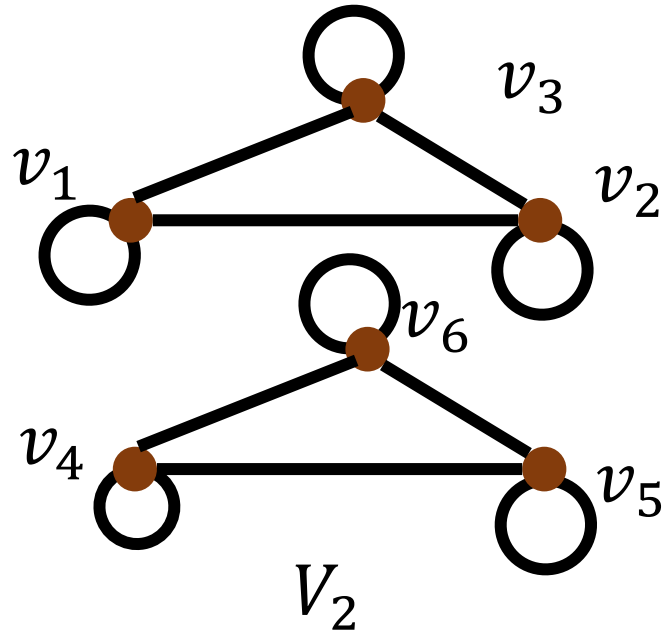
$$X = [\alpha e_1 + \delta e_2 \quad \beta e_1 + \varepsilon e_2 \quad \gamma e_1 + \varphi e_2]$$

Feature dim

$$e_1 = \frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\\0\\0\\0\end{bmatrix} \quad e_2 = \frac{1}{\sqrt{3}}\begin{bmatrix}0\\0\\0\\1\\1\\1\end{bmatrix} \quad X = \frac{1}{\sqrt{3}}\begin{bmatrix}\alpha & \beta & \gamma\\ \alpha & \beta & \gamma\\ \alpha & \beta & \gamma\\ \delta & \varepsilon & \varphi\\ \delta & \varepsilon & \varphi\\ \delta & \varepsilon & \varphi\end{bmatrix} \in \mathcal{M}$$

Node

Application to GCN



$$X = \frac{1}{\sqrt{3}} \begin{bmatrix} \alpha & \beta & \gamma \\ \alpha & \beta & \gamma \\ \alpha & \beta & \gamma \\ \delta & \varepsilon & \varphi \\ \delta & \varepsilon & \varphi \\ \delta & \varepsilon & \varphi \end{bmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{matrix} \} V_1 \\ \\ \\ \} V_2 \\ \\ \end{matrix}$$

$s\lambda < 1 \quad \lambda < 1 \quad$ If $s < 1$

exponential information loss of GCNs in terms of the layer size

If $s \geq 1$

Information loss may still happen

Discussion

- ✓ In reality, graph is usually not too sparse, so GCN is still applicable
- ✓ Find the sweet-spot to make the singular value of the weight not too small and not too big
- ✓ Remedy for over-smoothing: sample edges while training to make the graph sparser

DROPEDGE: TOWARDS DEEP GRAPH CONVOLUTIONAL NETWORKS ON NODE CLASSIFICATION

Yu Rong[1], **Wenbing Huang**[2]*, **Tingyang Xu**[1], **Junzhou Huang**[1]
[1] Tencent AI Lab
[2] Beijing National Research Center for Information Science and Technology (BNRist),
State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University

ICLR'20 poster (6, 3, 3)

https://openreview.net/pdf?id=Hkx1qkrKPr

DropEdge

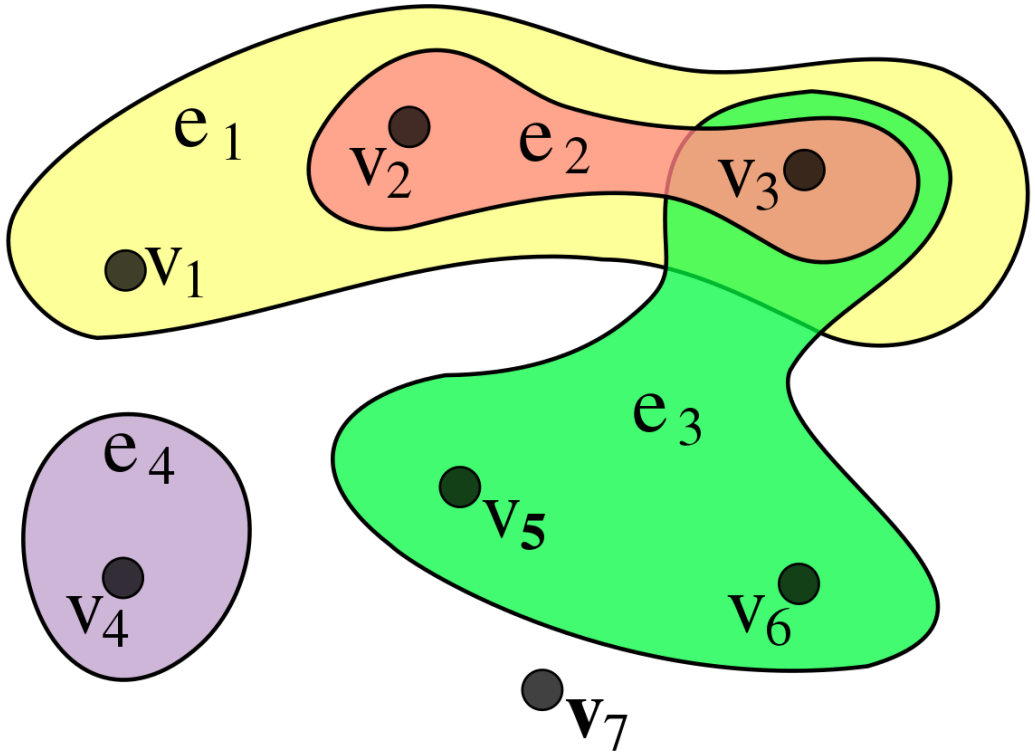
- ✓ Randomly drop the edge of input graph with probability p
  - ➢每一層都用同一個 drop 過的 adjacency matrix
  - ➢每一層都用不同的 drop 過的 adjacency matrix
- ✓ 可以避免 over-smoothing 的問題

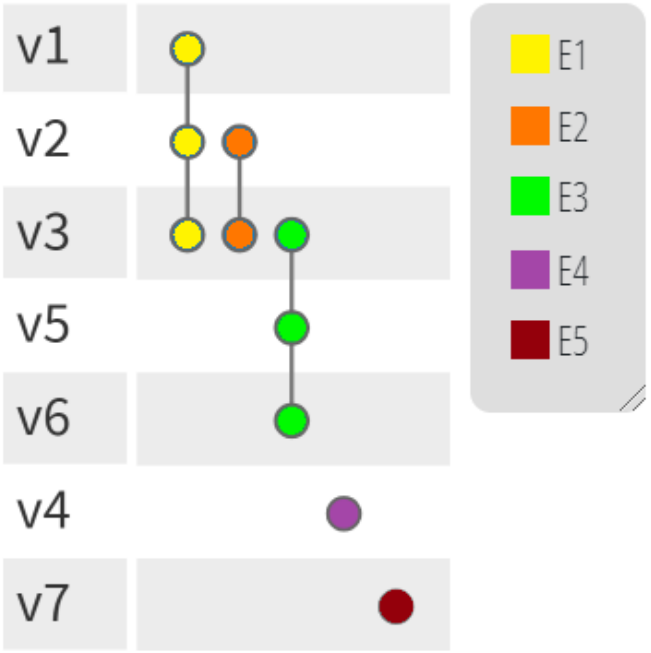
DropEdge

Dataset	Backbone	2 layers		8 layers		32 layers	
		Orignal	DropEdge	Orignal	DropEdge	Orignal	DropEdge
Cora	GCN	86.10	**86.50**	78.70	**85.80**	71.60	**74.60**
	ResGCN	-	-	85.40	**86.90**	85.10	**86.80**
	JKNet	-	-	86.70	**87.80**	87.10	**87.60**
	IncepGCN	-	-	86.70	**88.20**	87.40	**87.70**
	GraphSAGE	87.80	**88.10**	84.30	**87.10**	31.90	**32.20**
Citeseer	GCN	75.90	**78.70**	74.60	**77.20**	59.20	**61.40**
	ResGCN	-	-	77.80	**78.80**	74.40	**77.90**
	JKNet	-	-	79.20	**80.20**	71.70	**80.00**
	IncepGCN	-	-	79.60	**80.50**	72.60	**80.30**
	GraphSAGE	78.40	**80.00**	74.10	**77.10**	37.00	**53.60**
Pubmed	GCN	90.20	**91.20**	90.10	**90.90**	84.60	**86.20**
	ResGCN	-	-	89.60	**90.50**	90.20	**91.10**
	JKNet	-	-	90.60	**91.20**	89.20	**91.30**
	IncepGCN	-	-	90.20	**91.50**	OOM	**90.50**
	GraphSAGE	90.10	**90.70**	90.20	**91.70**	41.30	**47.90**
Reddit	GCN	96.11	**96.13**	96.17	**96.48**	45.55	**50.51**
	ResGCN	-	-	96.37	**96.46**	93.93	**94.27**
	JKNet	-	-	96.82	**97.02**	OOM	OOM
	IncepGCN	-	-	96.43	**96.87**	OOM	OOM
	GraphSAGE	96.22	**96.28**	96.38	**96.42**	96.43	**96.47**

HyperGCN

✓ Hypergraph





https://en.wikipedia.org/wiki/Hypergraph

HyperGCN

- ✓ Hypergraph in real life(?)
- ✓ Co-author network, co-citation network

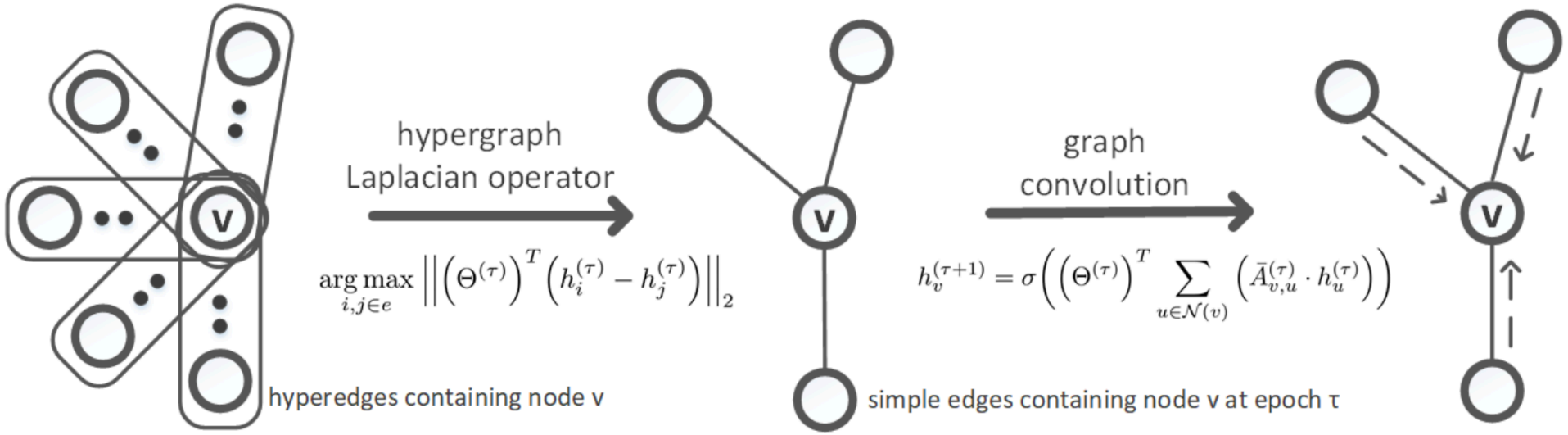


Figure 1: Graph convolution on a hypernode v using HyperGCN.

https://arxiv.org/pdf/1809.02589.pdf

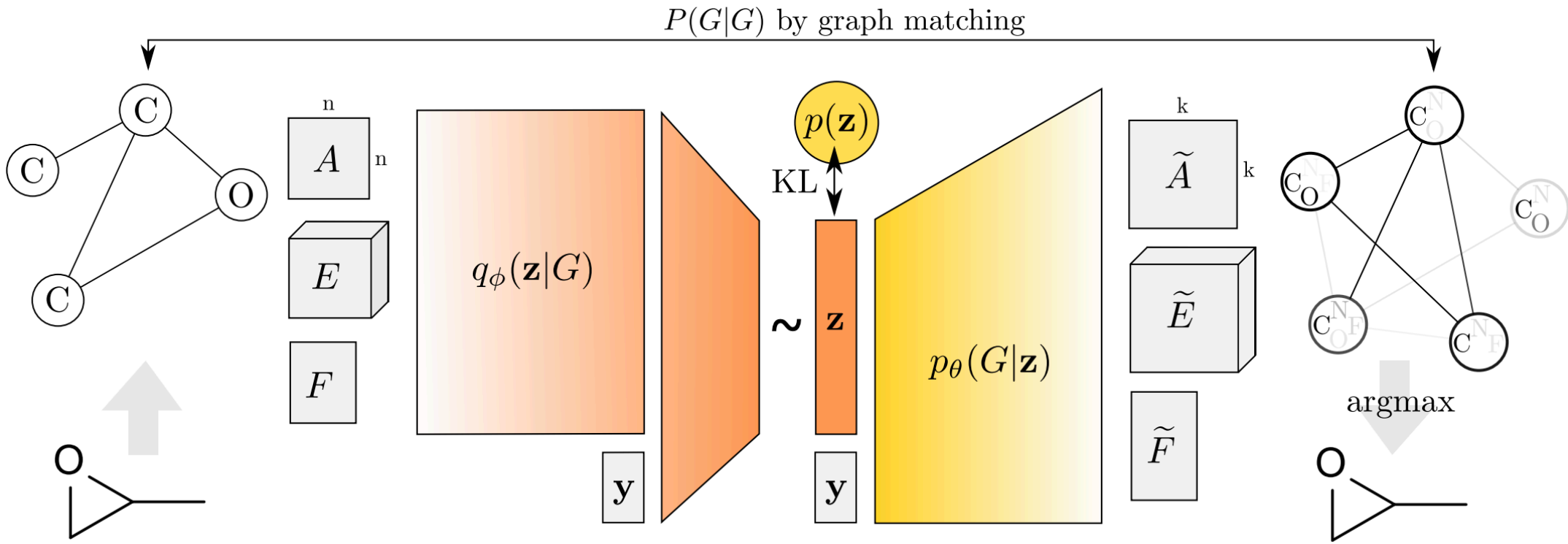
Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

Graph Generation

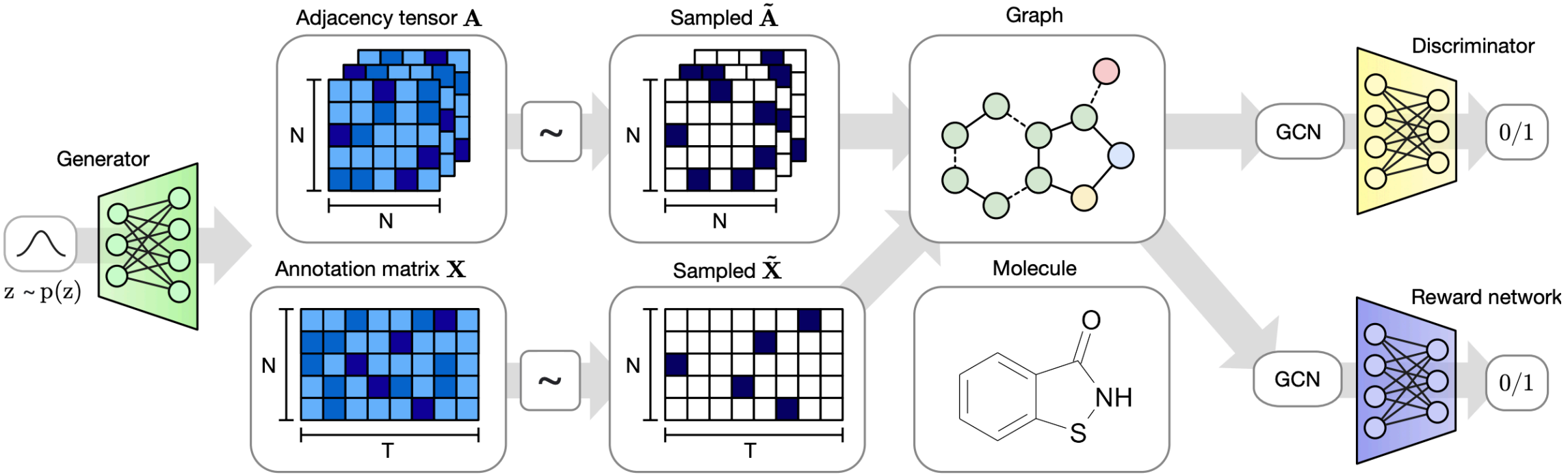
- ✓ VAE-based model: Generate a whole graph in one step
- ✓ GAN-based model: Generate a whole graph in one step
- ✓ Auto-regressive-based model: Generate a node or an edge in one step

VAE based model



https://arxiv.org/pdf/1802.03480.pdf

GAN-based model



https://arxiv.org/pdf/1901.00596.pdf

AR-based model

https://arxiv.org/pdf/1803.03324.pdf

$$s_u = f_s(\mathbf{h}_u^{(T)}, \mathbf{h}_v^{(T)}), \quad \forall u \in V$$

$$f_{addnode}(G) = \text{s}$$

$$f_{addedge}(G, v) =$$

$$f_{addnode}(G) =$$

$$f_{addedge}(G, v) = f_{nodes}(G, v) = \text{softmax}(\mathbf{s})$$

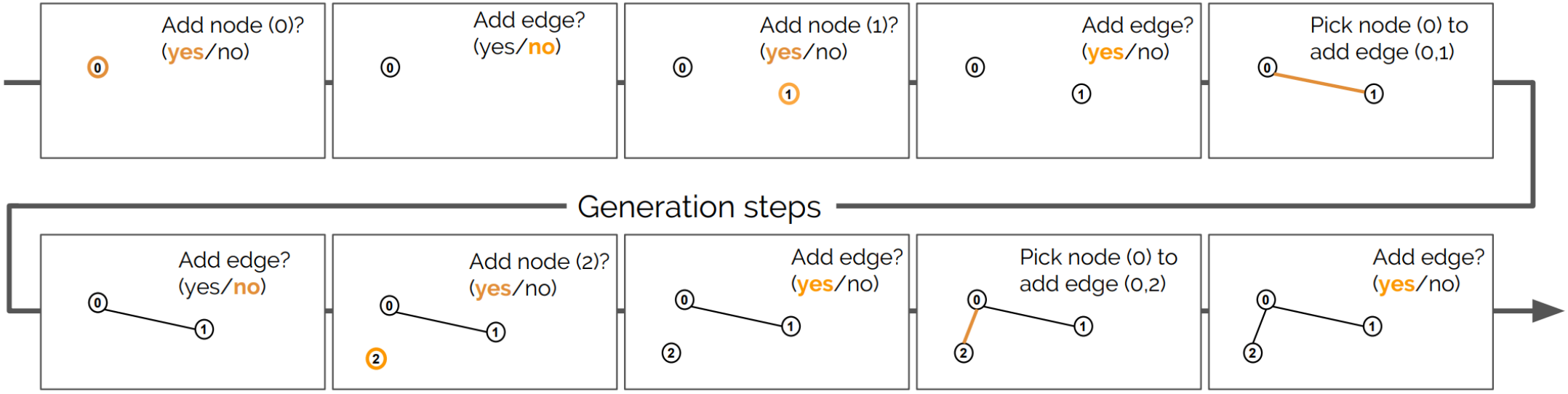


Figure 1. Depiction of the steps taken during the generation process.

$$\mathbf{a}_v = \sum_{u:(u,v)\in E} f_e(\mathbf{h}_u, \mathbf{h}_v, \mathbf{x}_{u,v}) \quad \forall v \in V$$

$$\mathbf{h}_v' = f_n(\mathbf{a}_v, \mathbf{h}_v) \quad \forall v \in V,$$

$$\mathbf{h}_V^{(T)} = \text{prop}^{(T)}(\mathbf{h}_V, G)$$

$$\mathbf{h}_G = R(\mathbf{h}_V^{(T)}, G)$$

Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

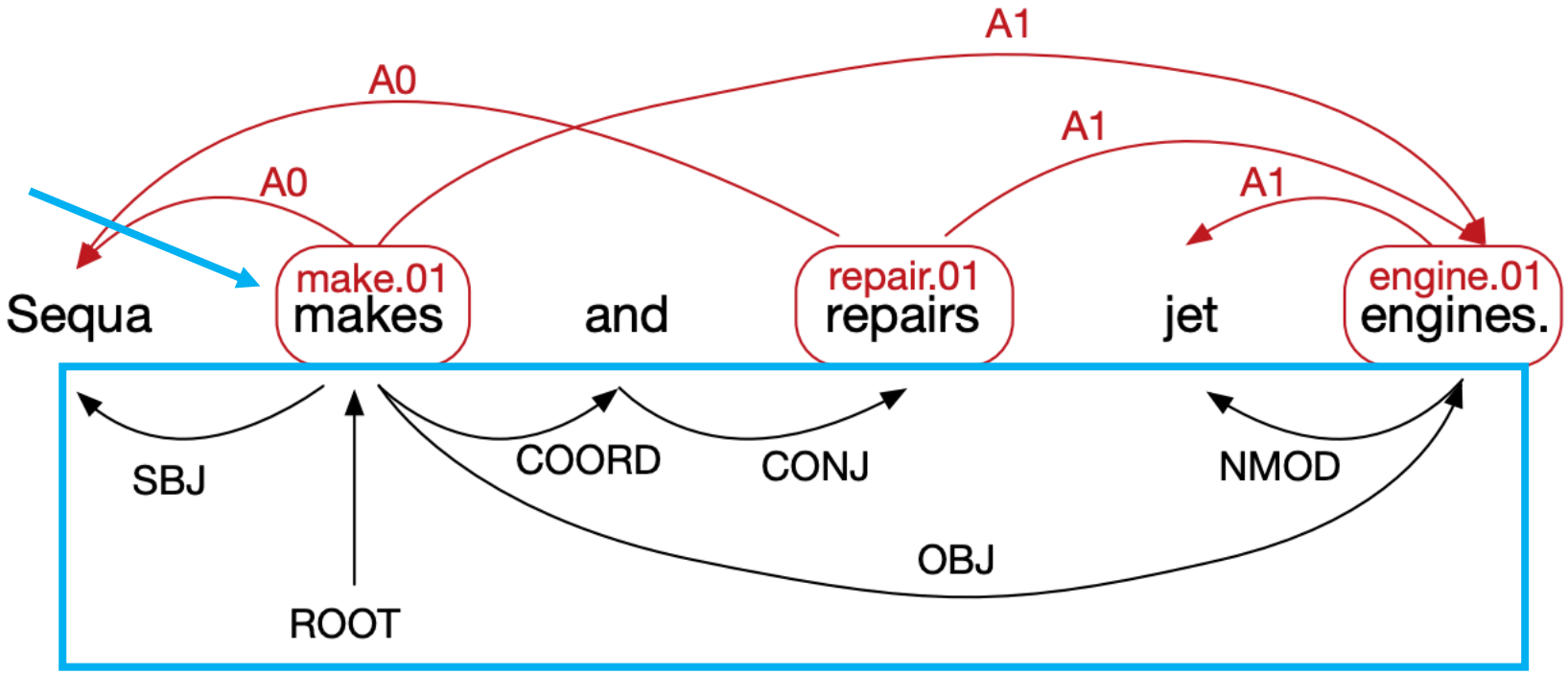
GNN for NLP

- ✓ Semantic Roles Labeling
- ✓ Event Detection
- ✓ Document Time Stamping
- ✓ Name Entity Recognition
- ✓ Relation Extraction
- ✓ Knowledge Graph

GCN for Semantic Roles Labeling

- ✓ Assume we have syntactic parse tree, we know where the predicates in the sentences are, and our model don't need to deal with predicate disambiguation problems.

We have the disambiguated predicates



We have the syntactical tree

https://www.aclweb.org/anthology/D17-1159.pdf

GCN for Semantic Roles Labeling

✓ Model architecture

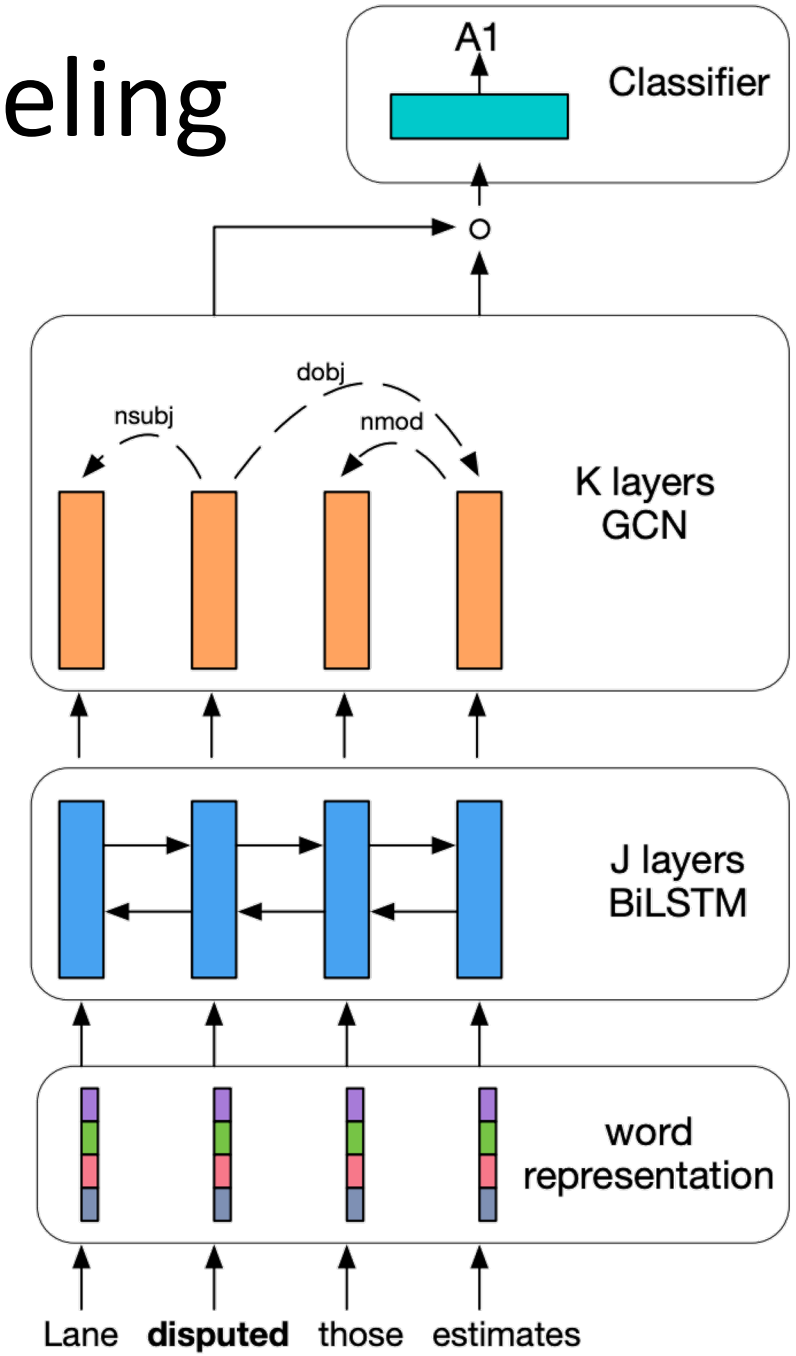
Bias depends on dependency relation

$$h_v^{(k+1)} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} \big(V_{dir(u,v)}^{(k)} h_u^{(k)} + b_{L(u,v)}^{(k)}\big)\Big)$$

Weight, just like GAT

Different edge directions use different weight

1) Head > dependent
2) Dependent > head
3) Self-loop



Outline

- ✓ Introduction
- ✓ Roadmap
- ✓ Tasks, Dataset, and Benchmark
- ✓ Spatial-based GNN
- ✓ Graph Signal Processing and Spectral-based GNN
- ✓ Graph Generation
- ✓ GNN for NLP
- ✓ Online Resources

PyTorch

- ✓ PyTorch Geometric

- ✓ Deep Graph Library

GNN Roadmap

Theoretical analysis: GIN, GCN

Convolution

Spatial-based

Aggregation	Method
Sum	NN4G
Mean	DCNN, DGC, GraphSAGE
Weighted sum	MoNET, GAT, GIN
LSTM	GraphSAGE
Max Pooling	GraphSAGE

Spectral-based

ChebNet → GCN → HyperGCN

Tasks

- Supervised classification
- Semi-Supervised Learning
- Representation learning: Graph InfoMax
- Generation: GraphVAE, MolGAN, etc.

~~Application: Natural Language Processing~~

Summary and Take Home Notes

- ✓ GAT and GCN are the most popular GNNs
- ✓ Although GCN is mathematically driven, we tend to ignore its math
- ✓ GNN (or GCN) suffers from information lose while getting deeper
- ✓ Many deep learning models can be slightly modified and designed to fit graph data, such as Deep Graph InfoMax, Graph Transformer, GraphBert.
- ✓ Theoretical analysis must be dealt with in the future
- ✓ GNN can be applied to a variety of tasks

Q&A

Reference

- ✓ Functions of Matrices: Theory and Computation
- ✓ Signal and System by Prof. Lee: http://speech.ee.ntu.edu.tw/courses.html
- ✓ Machine Learning by Prof. Hung-yi Lee

  http://speech.ee.ntu.edu.tw/~tlkagk/courses.html
- ✓ Vertex-Frequency Analysis of Graph Signals
- ✓ All the papers mentioned have link appended on the corresponding page