

Graph Neural Networks Inspired by Differential Equations

Jeongwhan Choi (최정환)

Yonsei University



BigDyL

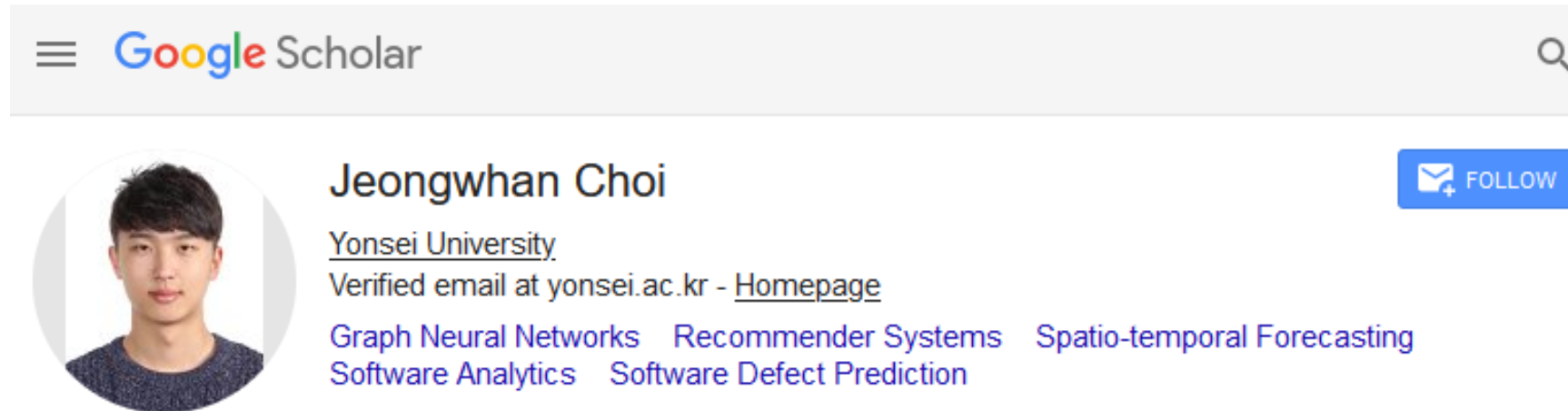
[big di:]

Big Data Analytics Laboratory
Yonsei University

Contents

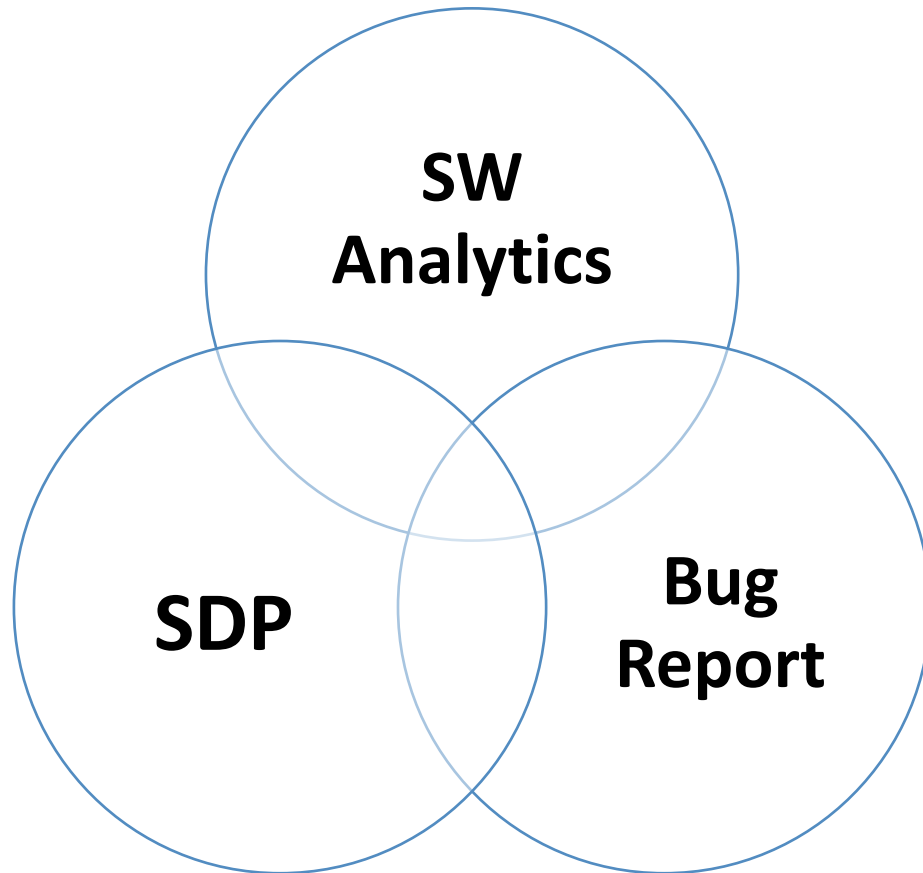
- **Introduction**
- Graph Neural Networks
- Diffusion Process on Graphs
- Graph Neural Networks Inspired by Diffusion Process
- Applications
- Conclusions

Who Am I?

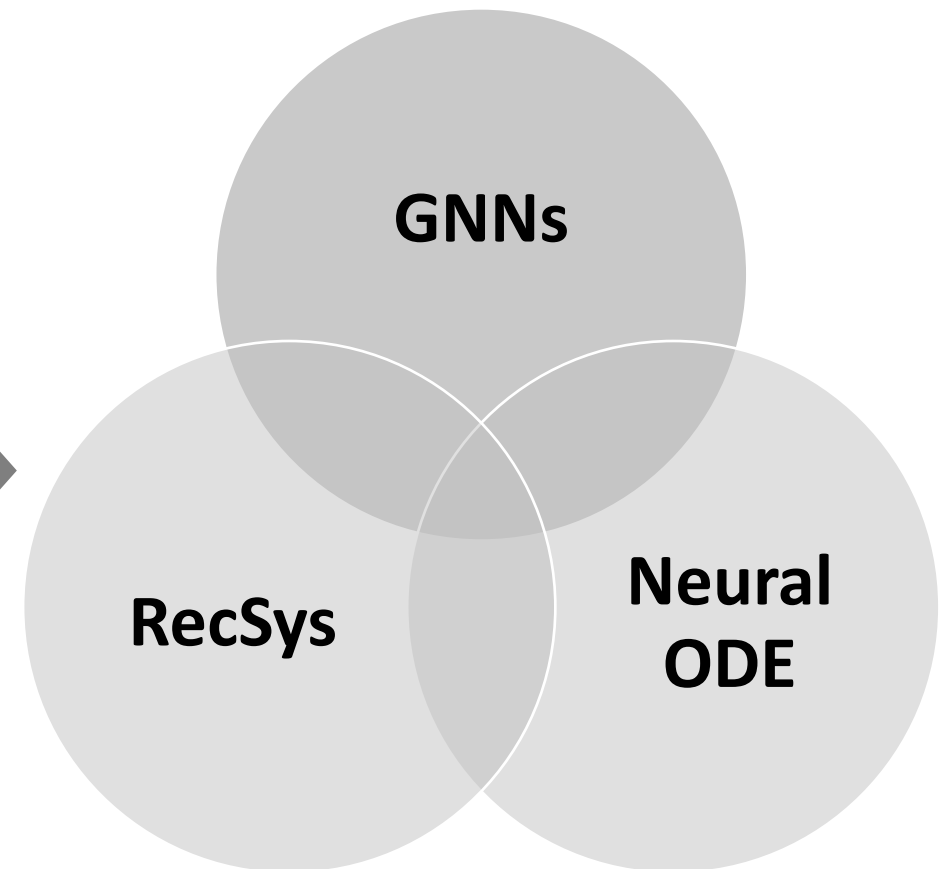
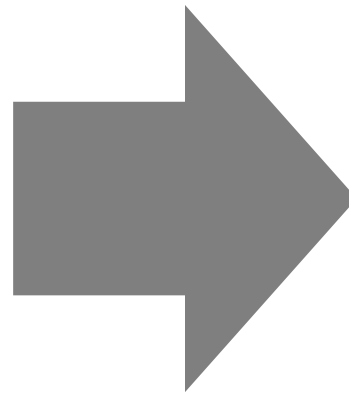


- 2020 – now, Integrated Ph.D., Artificial Intelligence, Yonsei University
- 2016 – 2020, B.S. Software Engineering, Jeonbuk National University

Research Interests



Software Engineering



Artificial Intelligence

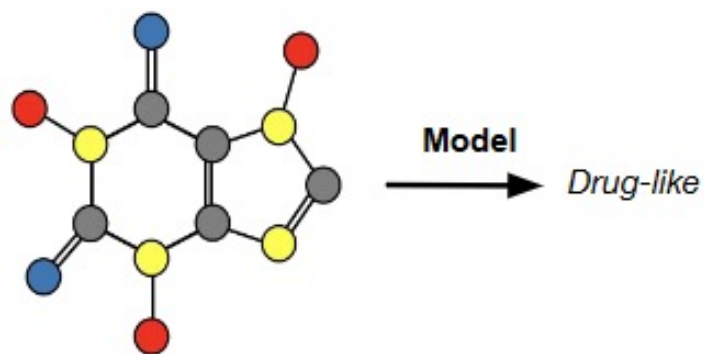
Research Covered in This Talk

- Recommender Systems:
 - LT-OCF (CIKM'21), BSPM (SIGIR'23)
- Graph Neural Networks:
 - GREAD (ICML'23)

Contents

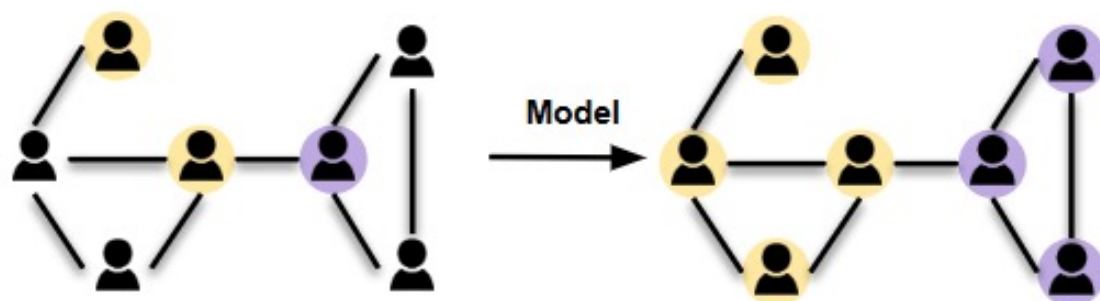
- Introduction
- **Graph Neural Networks**
- Diffusion Process on Graphs
- Graph Neural Networks Inspired by Diffusion Process
- Applications
- Conclusions

Tasks on Graphs



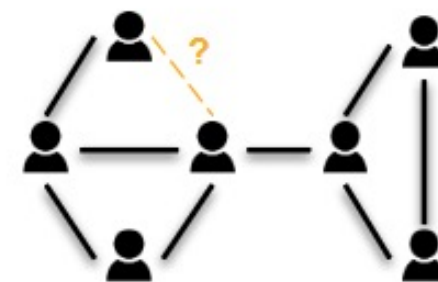
Model → *Drug-like*

Graph
Classification



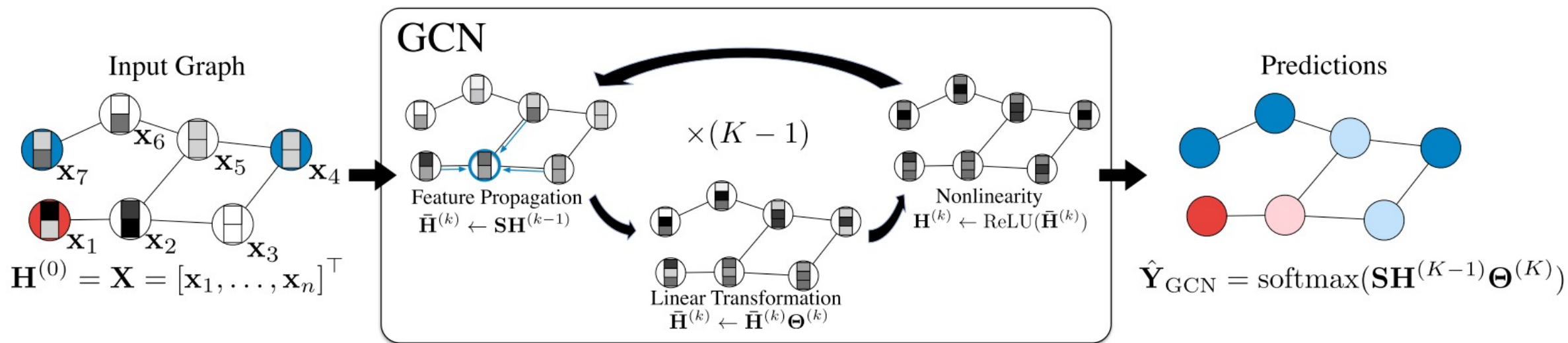
Node
Classification

● Omnivore
● Vegetarian

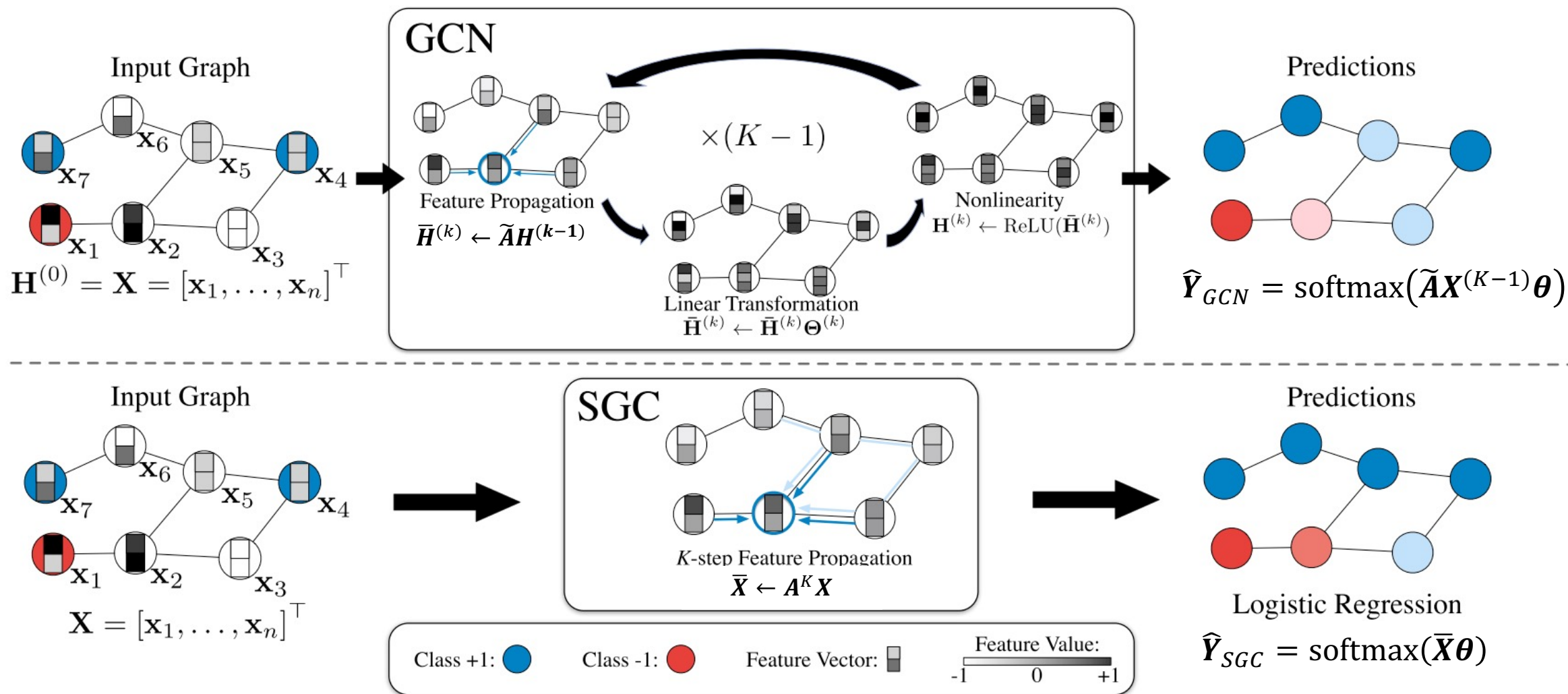


Link
Prediction

Graph Convolutional Networks (GCNs)



Simple Graph Convolution (SGC)



Linear GCN

- Linear GCNs achieve comparable performance to non-linear GCNs.
- Simple Graph Convolution (SGC)
 - Given input $\mathbf{X}$, label $\mathbf{Y}$, adjacency matrix $\mathbf{A}$, SGC predicts with

$$\hat{\mathbf{Y}}_{SGC} = \text{softmax}(\tilde{\mathbf{A}}^K \mathbf{X} \boldsymbol{\theta}).$$

- K propagation steps (core):

$$\mathbf{X}^{(K)} \leftarrow \tilde{\mathbf{A}} \mathbf{X}^{(K-1)}, \text{ where } \tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \implies \mathbf{X}^{(K)} = \tilde{\mathbf{A}}^K \mathbf{X}$$

- Linear classification:

$$\hat{\mathbf{Y}}_{SGC} = \text{softmax}(\mathbf{X}^{(K)} \boldsymbol{\theta}).$$

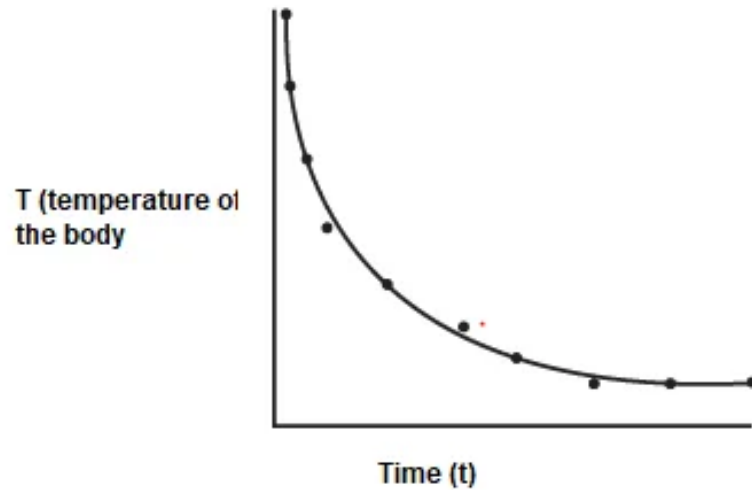
- Advantages: memory and parameter efficiency (preprocessed features).
 - Disadvantage: oversmoothing, inferior performance.

Contents

- Introduction
- Graph Neural Networks
- **Diffusion Process on Graphs**
- Graph Neural Networks Inspired by Diffusion Process
- Applications
- Conclusions

Newton Law of Cooling

- “The temperature a hot body loses in a given time is proportional to the temperature difference between the object and the environment.”



(824)

with a little pressing, I took a drop thereof, and in it discover'd a mighty number of living Creatures. I repeated my observation the same evening with the same success, but the next day I could find none of them alive; and whereas I had laid that drop upon a small Copper Plate, I fancied to my self that the exhalation of the moisture might be the cause of their death, and not the cold weather, which at that time was very moderate.

In the beginning of April I took the Male seed of a Jack or Pike, but could discover nothing more than in that of a Cod-fish, but having added about four times as much Water in quantity as the matter itself was, and then making my remarks, I could perceive that the *Animalcula* did not only wax stronger and swifter, but, to my great amazement, I saw them move with that celerity, that I could compare it to nothing more than what we have seen with our naked Eye, a River Fish chafed by its powerful Enemy, which is just ready to devour it: You must observe that this whole Course was not longer than the Diameter of a single Hair of ones Head.

VII. Scala graduum Caloris.

Calorum Descriptiones & signa.

0		C	Alor aeris hyberni ubi aqua incipit gelu
0,1,2.			rigescere. Innotescit hic calor accurate
2,3,4.			locando Thermometrum in nive compressa
4,5,6			quo tempore gelu solvitur.
6			Calores aeris hyberni.
12			Calores aeris verni & autumnalis.
			Calores aeris æstivi.
			Calor aeris meridiani circa mensum Ju-
			lium.
			Calor maximus quem Thermometer ad con-
			tactum

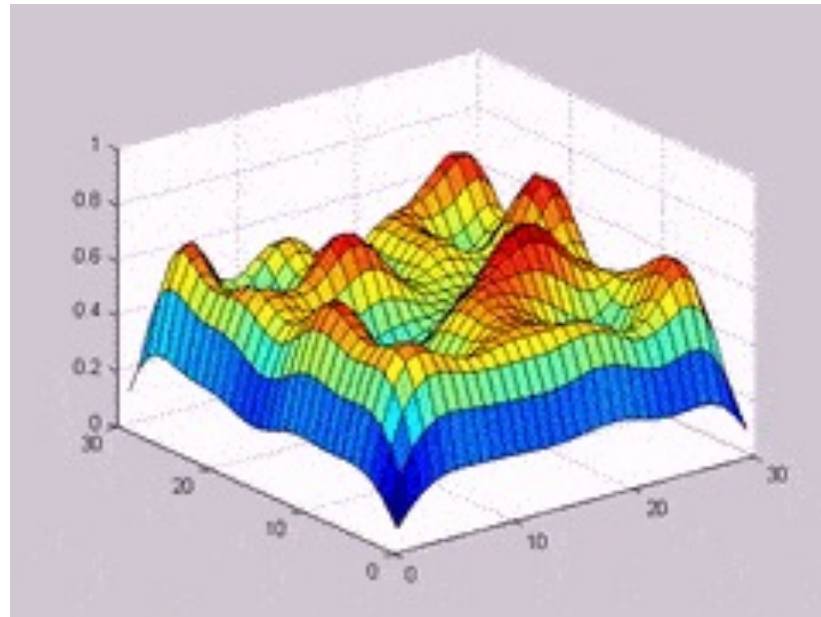


Issac Newton

Diffusion takes many forms

- Heat diffusion:

$$\frac{d\mathbf{H}(t)}{dt} = -\Delta\mathbf{H}(t)$$



Heat equation, wikipedia

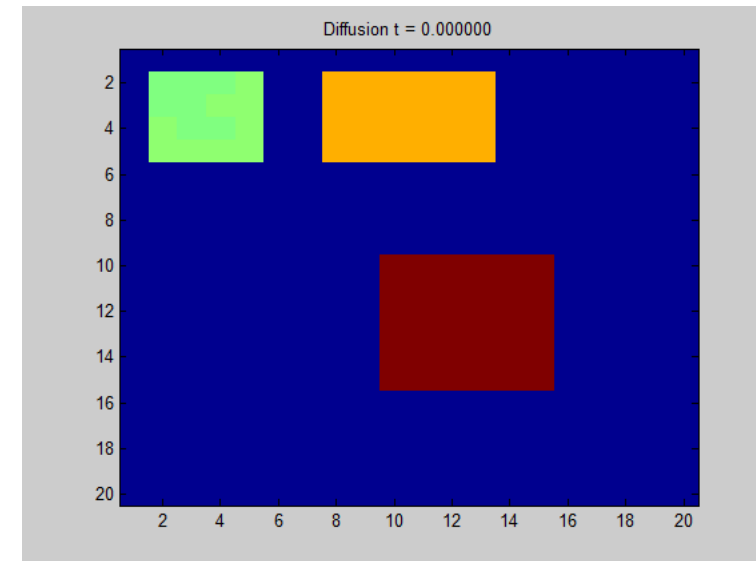


Joseph Fourier

Laplacian Diffusion on Graphs

- The following differential equation is a temperature distribution across a graph G .
- Let $L=D-A$ be the Laplace matrix of G .

$$\frac{d\mathbf{H}(t)}{dt} = -\mathbf{L}\mathbf{H}(t).$$



Laplacian matrix, wikipedia

Contents

- Introduction
- Graph Neural Networks
- Diffusion Process on Graphs
- **Graph Neural Networks Inspired by Diffusion Process**
- Applications
- Conclusions

Equivalence between SGC and Diffusion Process

- Key insight from a continuous perspective:
 - SGC's propagation = a discretization of the diffusion equation
- Graph Diffusion Equation

$$\frac{d\mathbf{X}(t)}{dt} = -\tilde{\mathbf{L}}\mathbf{X}(t),$$

where $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$ is the graph Laplacian matrix, and $\mathbf{X}(0)$ is an input $\mathbf{X}$.

Equivalence between SGC and Diffusion Process

- Key insight from a continuous perspective:
 - SGC's propagation = a discretization of the diffusion equation

- Graph Diffusion Equation

$$\frac{d\mathbf{X}(t)}{dt} = -\tilde{\mathbf{L}}\mathbf{X}(t),$$

where $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$ is the graph Laplacian matrix, and $\mathbf{X}(0)$ is an input $\mathbf{X}$.

- Discretization

- Applying the forward Euler method with time interval Δt :

$$\begin{aligned}\mathbf{X}(t + \Delta t) &= \mathbf{X}(t) - \Delta t \tilde{\mathbf{L}}\mathbf{X}(t) \\ &= \mathbf{X}(t) - \Delta t (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{X}(t) \\ &= [(1 - \Delta t)\mathbf{I} + \Delta t \tilde{\mathbf{A}}]\mathbf{X}(t)\end{aligned}$$

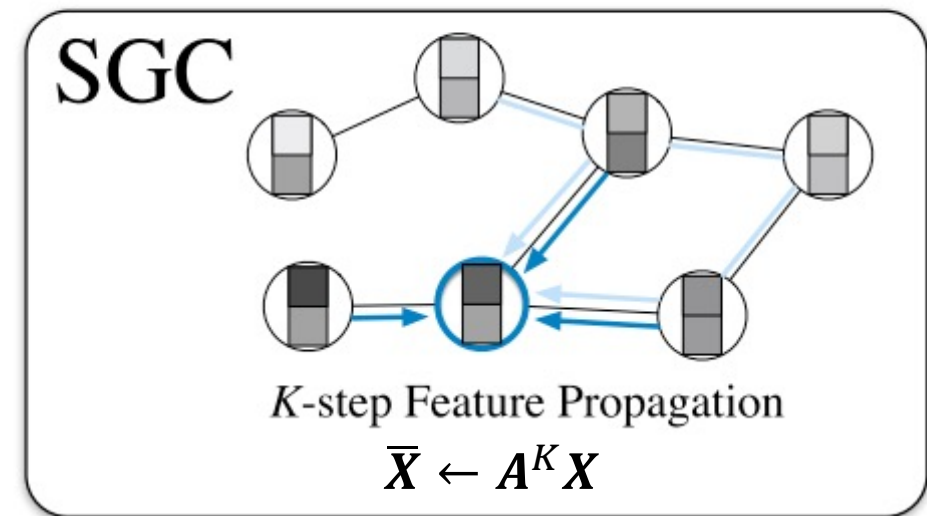
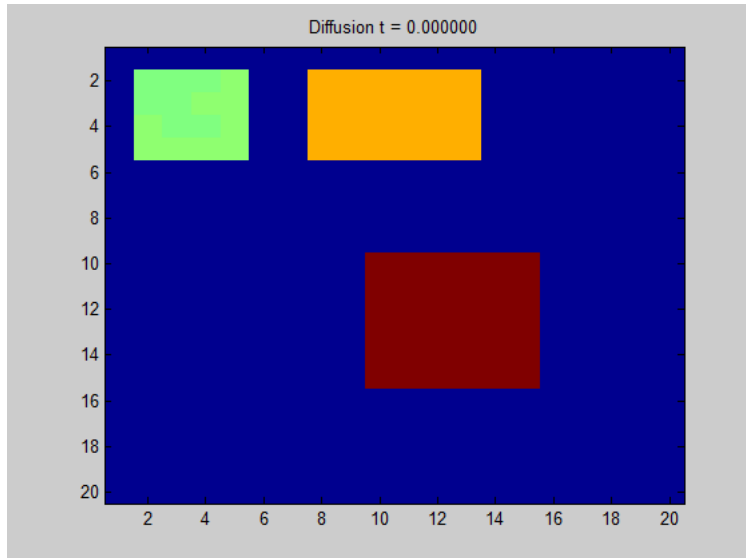
- Thus, SGC is the Euler discretization of Diffusion Eq. with step size $\Delta t = 1$.

GNNs Inspired by Diffusion Process

$$\frac{d\mathbf{X}(t)}{dt} = -\mathbf{L}\mathbf{X}(t)$$



$$\mathbf{X}^{(k)} = \mathbf{A}\mathbf{X}^{(k-1)}$$



Ordinary Differential Equations (ODEs) (1/4)

- Initial value problem:

$$\frac{dx(t)}{dt} = f(x(t), t, \theta); x(t_0) \text{ is given; } x(t_1) = ?$$

- Solution:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta) dt$$

Example:

$$\begin{aligned} \frac{dx}{dt} &= 2t; x(0) = 2; x(1) = ? \\ \Rightarrow x(1) &= x(0) + \int_0^1 2t dt \\ &= x(0) + (t^2|_{t=1} - t^2|_{t=0}) \\ &= 2 + 1^2 - 0^2 \\ &= 3 \end{aligned}$$

Ordinary Differential Equations (ODEs) (2/4)

- Initial value problem:

$$\frac{dx(t)}{dt} = f(x(t), t, \theta); x(t_0) \text{ is given}; x(t_1) = ?$$

- Solution:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta) dt$$

What if this cannot be
analytically integrated?



Ordinary Differential Equations (ODEs) (3/4)

- Initial value problem:

$$\frac{dx(t)}{dt} = f(x(t), t, \theta); x(t_0) \text{ is given; } x(t_1) = ?$$

- Solution:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta) dt$$

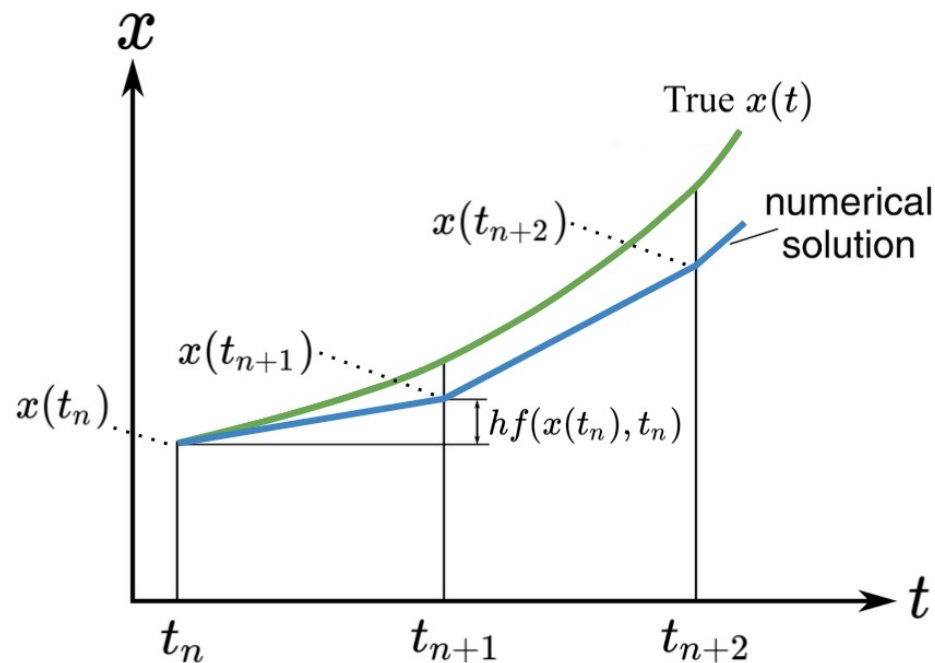
Approximation to $\int_{t_0}^{t_1} f(x(t), t, \theta) dt$ i.e.

Numerical Integration:

- Euler method
- Runge-Kutta methods
- ...

Ordinary Differential Equations (ODEs) (4/4)

- Euler's method / RK-1 method:
 - $t_{n+1} = t_n + h \rightarrow$ Step size h
 - $x(t_{n+1}) = x(t_n) + hf(x(t_n), t_n) \rightarrow$ Update using derivative f



Neural ODEs (Chen et al., NeurIPS '18)

- Initial value problem:

$$\frac{dx(t)}{dt} = f(x(t), t, \theta); x(t_0) \text{ is given}; x(t_1) = ?$$

- Solution:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta) dt$$

f is neural network!

What's the Big Deal?

- Initial value problem:

$$\frac{dx(t)}{dt} = f(x(t), t, \theta); x(t_0) \text{ is given}; x(t_1) = ?$$

- Solution:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta) dt$$

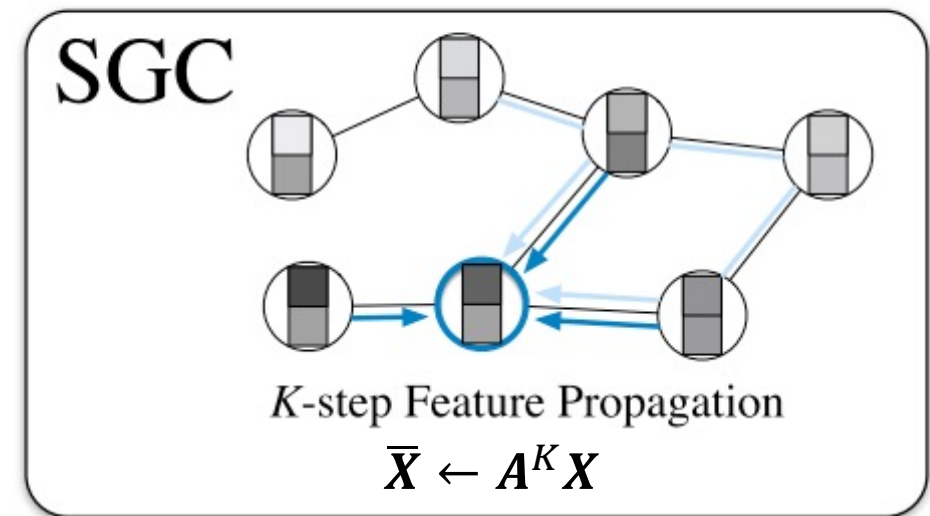
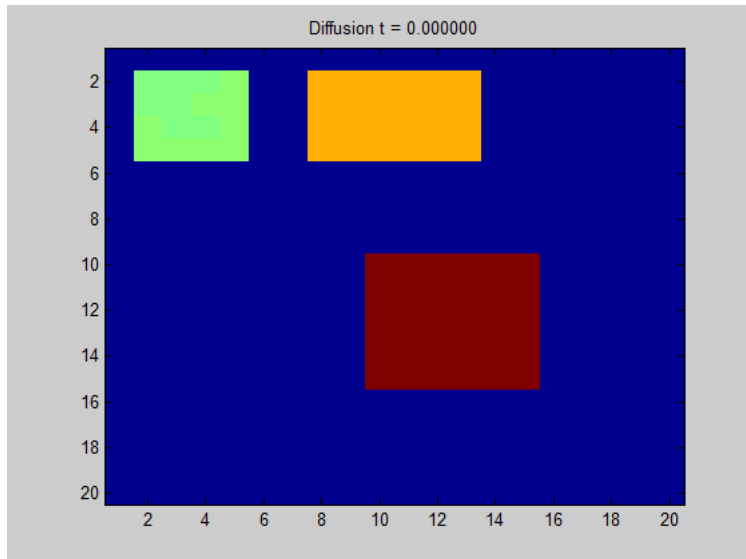
f is diffusion equation! $\frac{dX(t)}{dt} = -LX(t)$
 f can be linear GCNs!

GNNs Inspired by Diffusion Process

$$\frac{d\mathbf{X}(t)}{dt} = -\mathbf{L}\mathbf{X}(t)$$



$$\mathbf{X}^{(k)} = \mathbf{A}\mathbf{X}^{(k-1)}$$



Contents

- Introduction
- Graph Neural Networks
- Diffusion Process on Graphs
- Graph Neural Networks Inspired by Diffusion Process
- **Applications**
 - **Recommender Systems**
 - Node Classification
- Conclusions

Recommender Systems

- Building a good recommender remains an active research area.

Customers who bought this item also bought

Page 1 of 11

←

Book Title	Author	Rating	Format	Price	Prime
The Elements of Statistical Learning: Data Mining, Inference, and...	Trevor Hastie	★★★★☆ 17	Hardcover	CDN\$ 49.90	
Applied Predictive Modeling	Max Kuhn	★★★★☆ 9	Hardcover	CDN\$ 85.09	✓prime
Deep Learning	Ian Goodfellow	★★★★★ 8	Hardcover	CDN\$ 92.40	✓prime
R for Data Science: Import, Tidy, Transform, Visualize, and Model Data	Hadley Wickham	★★★★★ 7	Paperback	CDN\$ 41.48	✓prime
ggplot2: Elegant Graphics for Data Analysis	Hadley Wickham	★★★★☆ 1	Paperback	CDN\$ 55.65	✓prime
Python Machine Learning	Sebastian Raschka	★★★★☆ 7	Paperback	CDN\$ 47.97	✓prime
R for Everyone: Advanced Analytics and Graphics	Jared P. Lander	★★★★☆ 8	Paperback	CDN\$ 38.43	✓prime

→

An example of recommender systems

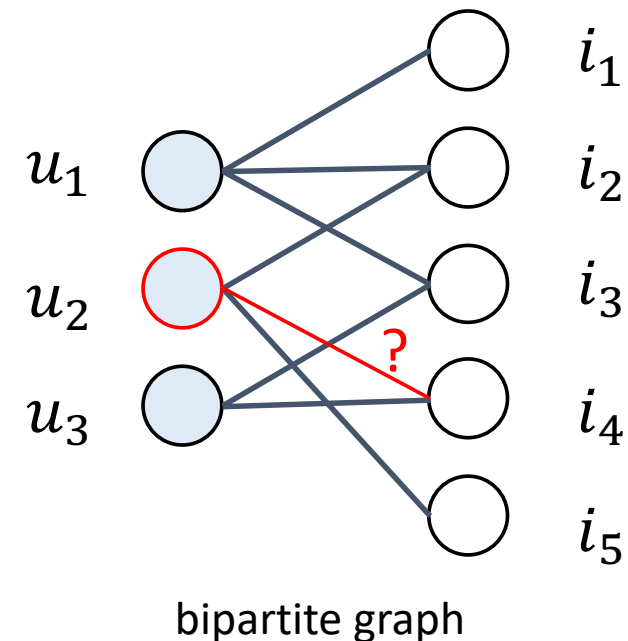
Collaborative Filtering (1/2)

- **Collaborative Filtering (CF)** is a long-standing research problem of recommender systems.
 - CF is to predict users' preferences from patterns.

items

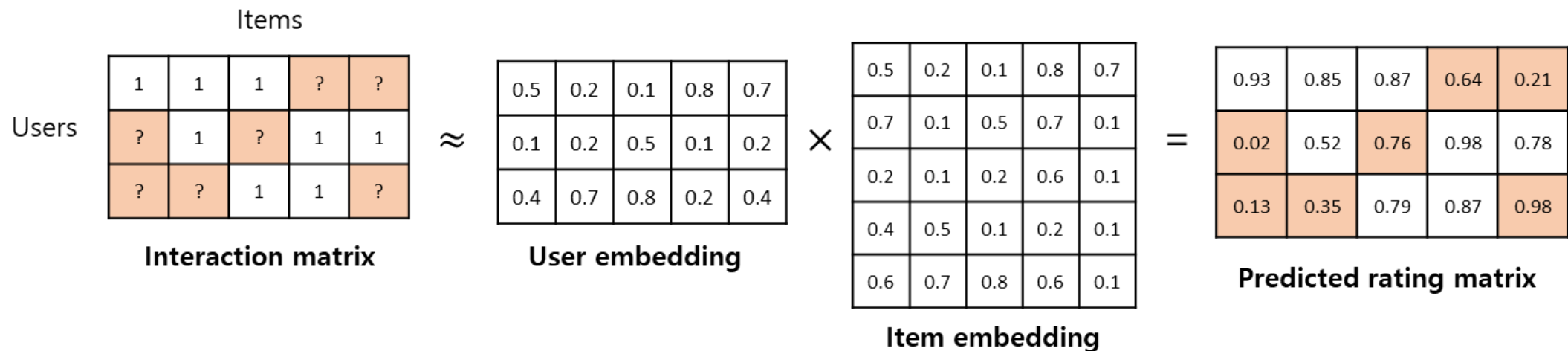
	1	1	1	0	0
users	0	1	0	?	1
	0	0	1	1	0

Interaction matrix



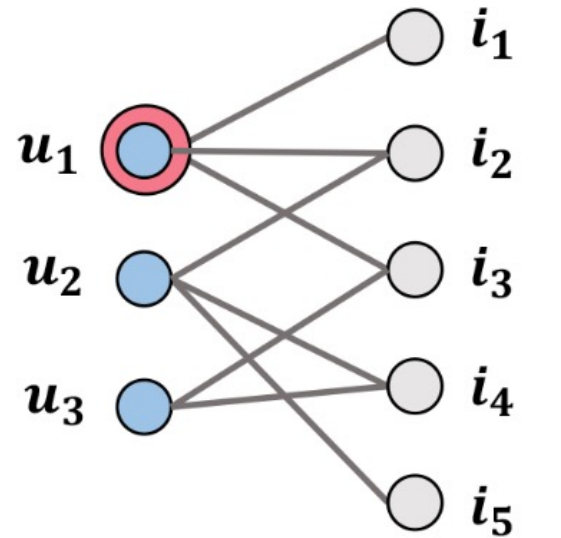
Collaborative Filtering (2/2)

- Let $\mathbf{R} \in \{0,1\}^{|U| \times |V|}$ be an interaction matrix.
- Matrix factorization had been widely used.
 - $\hat{\mathbf{R}} = \mathbf{E}_U \cdot \mathbf{E}_V$
- These days, however, GNN-based methods are popular for CF.

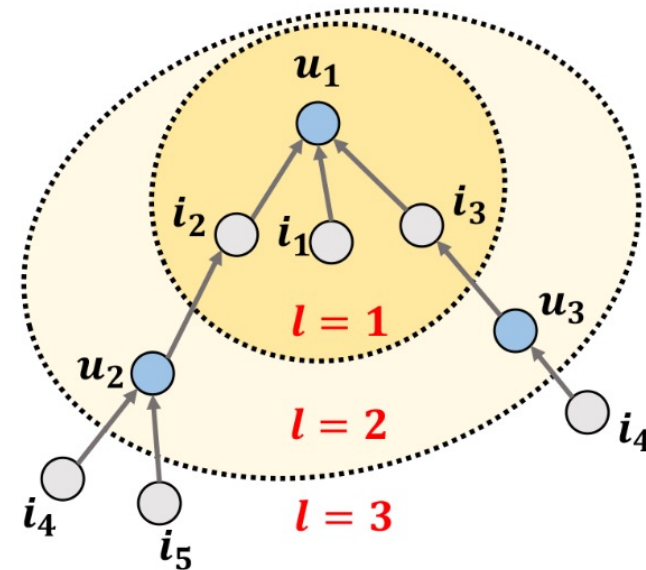


High-Order Connectivity

- High-order connectivity from user-item interactions
 - Definition: the paths that reach u_1 from any node with the path length l larger than 1.
 - A natural way to encode collaborative signal in the interaction graph structure.



User-Item Interaction Graph



High-order Connectivity for u_1

A Timeline of Collaborative Filtering (CFs)

MF and its variants

MF-BPR [UAI'09]
Neu-MF [WWW'17]

NGCF [SIGIR'19]

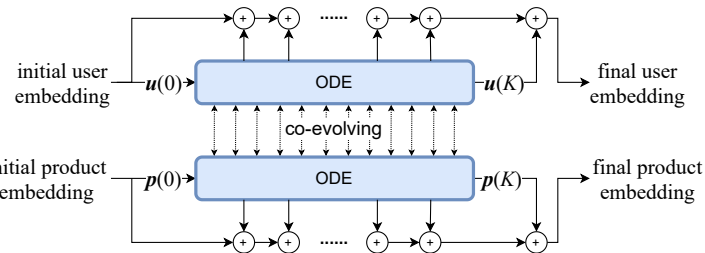
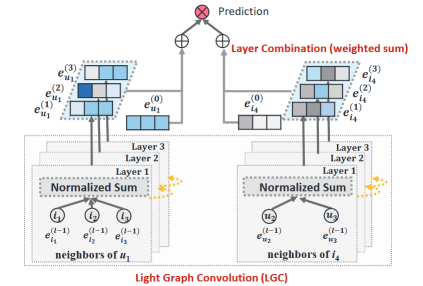
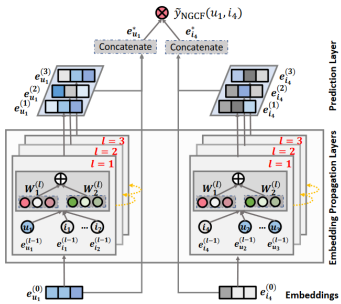
GCN inspired CF

Lighter GCN-based CF

LightGCN [SIGIR'20]
LR-GCCF [AAAI'20]

LT-OCF [CIKM'21]

Heat Diffusion inspired CF



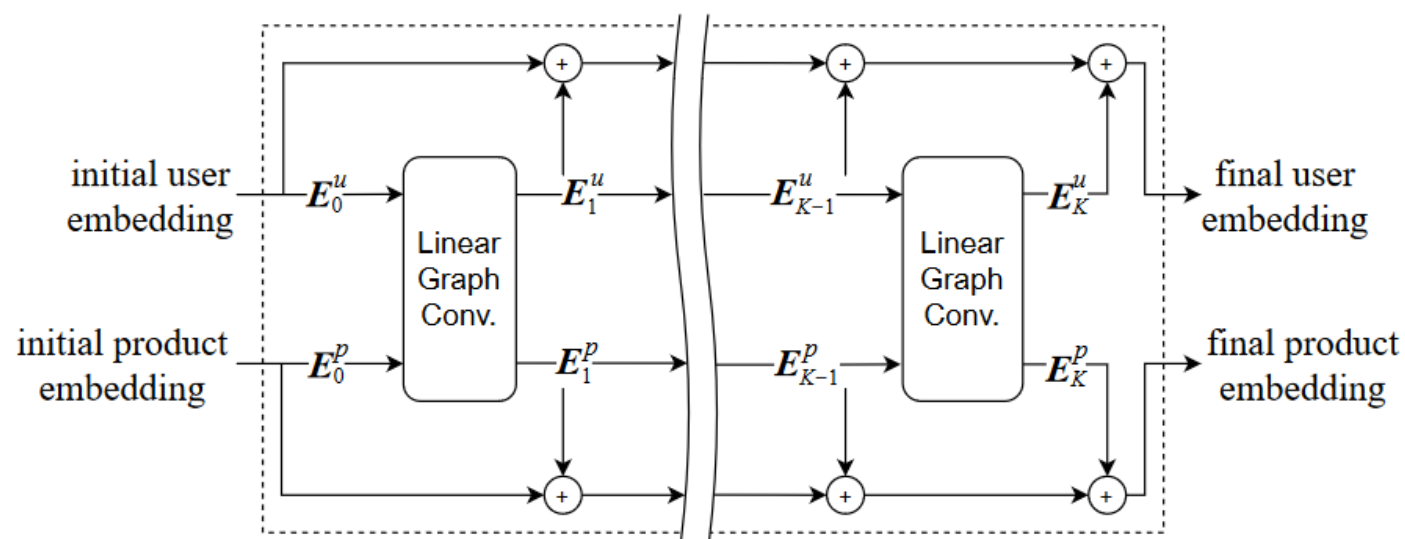
LightGCN

- A matrix form of linear GCNs:

$$\mathbf{E}^{(k)} = \tilde{\mathbf{A}} \mathbf{E}^{(k-1)}$$

- Layer combination (weighted sum):

$$\mathbf{E} = \alpha_0 \mathbf{E}^{(0)} + \alpha_1 \tilde{\mathbf{A}} \mathbf{E}^{(0)} + \dots + \alpha_K \tilde{\mathbf{A}}^K \mathbf{E}^{(0)}$$



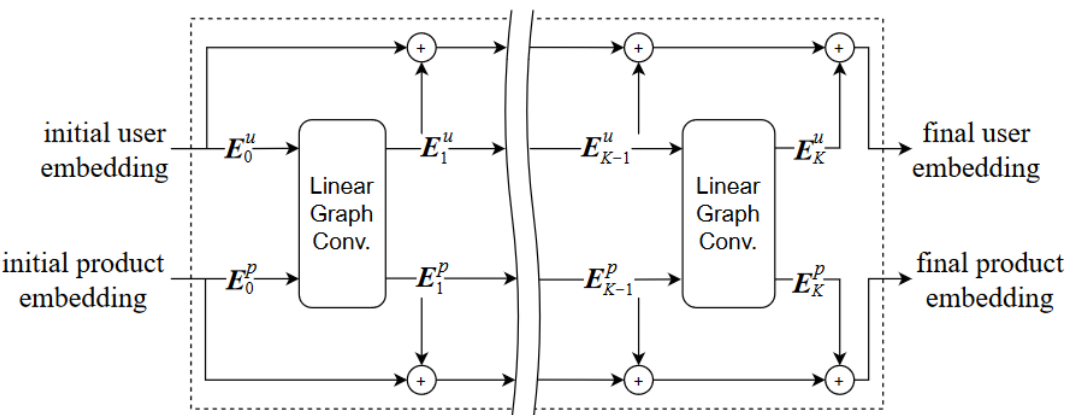
LT-OCF: Learnable-Time ODE-based CF (1/2)

$$\frac{d\mathbf{X}(t)}{dt} = -\mathbf{L}\mathbf{X}(t) \quad \longleftrightarrow \quad \mathbf{E}^{(k)} = \tilde{\mathbf{A}}\mathbf{E}^{(k-1)}$$

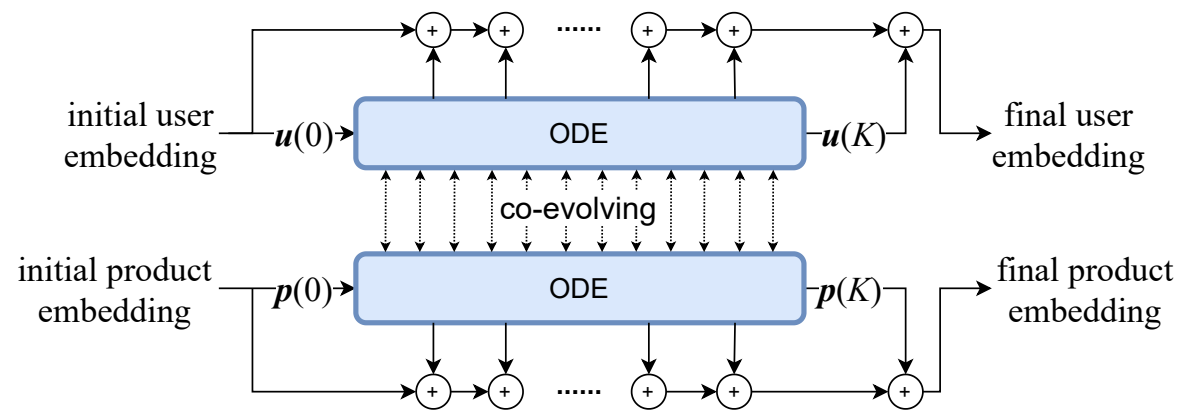
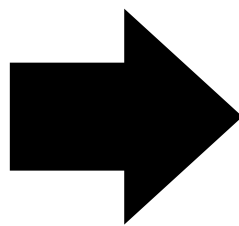
- LT-OCF **redesigns** the *linear GCN* on top of the concept of ODEs.
- LightGCN corresponds to discrete thermal diffusive processes.
- LT-OCF corresponds to continuous thermal diffusive processes.

LT-OCF: Learnable-Time ODE-based CF (2/2)

- Why we redesign?
 - Linear GCNs can be interpreted as the diffusion equation concept.
 - Time variable t is not only **continuous** but also **trainable**.



The architecture of LightGCN



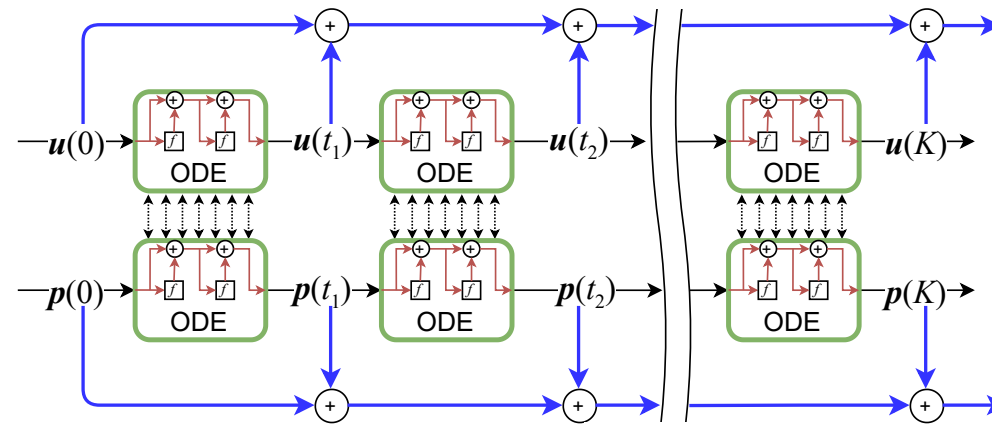
Our proposed LT-OCF

Learnable Time Architecture

- We extract $\mathbf{u}(t)$ and $\mathbf{p}(t)$ with different **learnable** time-points.

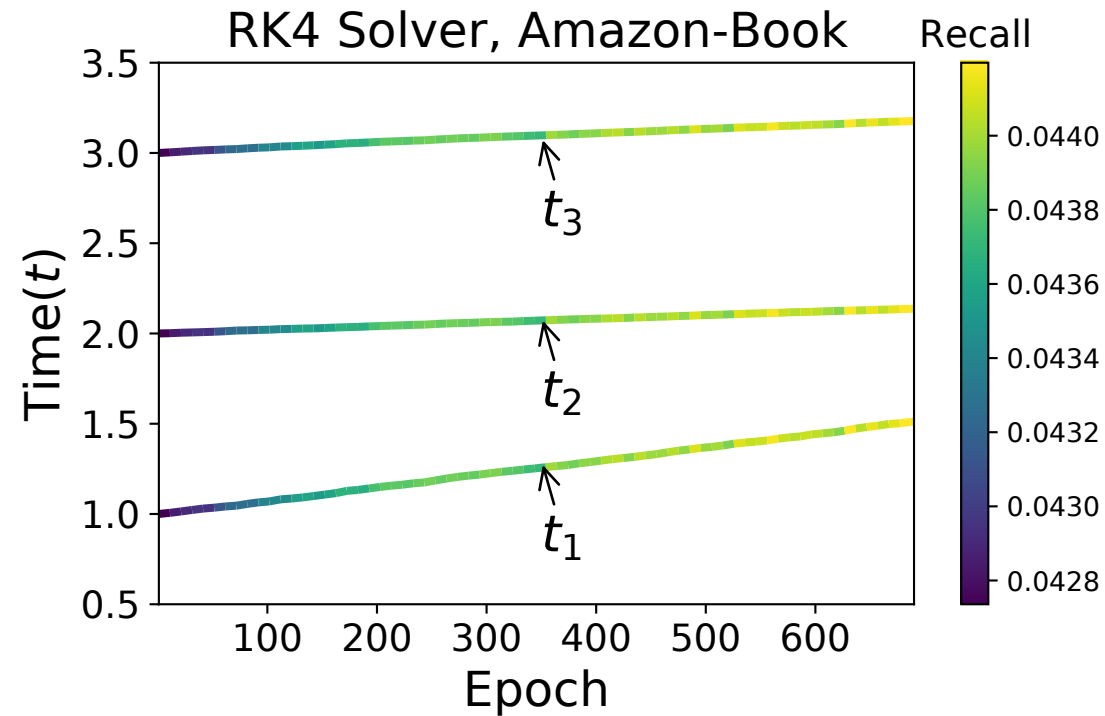
$$\mathbf{u}(t_1) = \mathbf{u}(0) + \int_0^{t_1} f(\mathbf{p}(t))dt, \quad \mathbf{p}(t_1) = \mathbf{p}(0) + \int_0^{t_1} g(\mathbf{u}(t))dt,$$

$$\mathbf{u}(K) = \mathbf{u}(t_T) + \int_{t_T}^K f(\mathbf{p}(t))dt, \quad \mathbf{p}(K) = \mathbf{p}(t_T) + \int_{t_T}^K g(\mathbf{u}(t))dt,$$



Our proposed dual co-evolving ODEs

Trianing Curves of Time Points



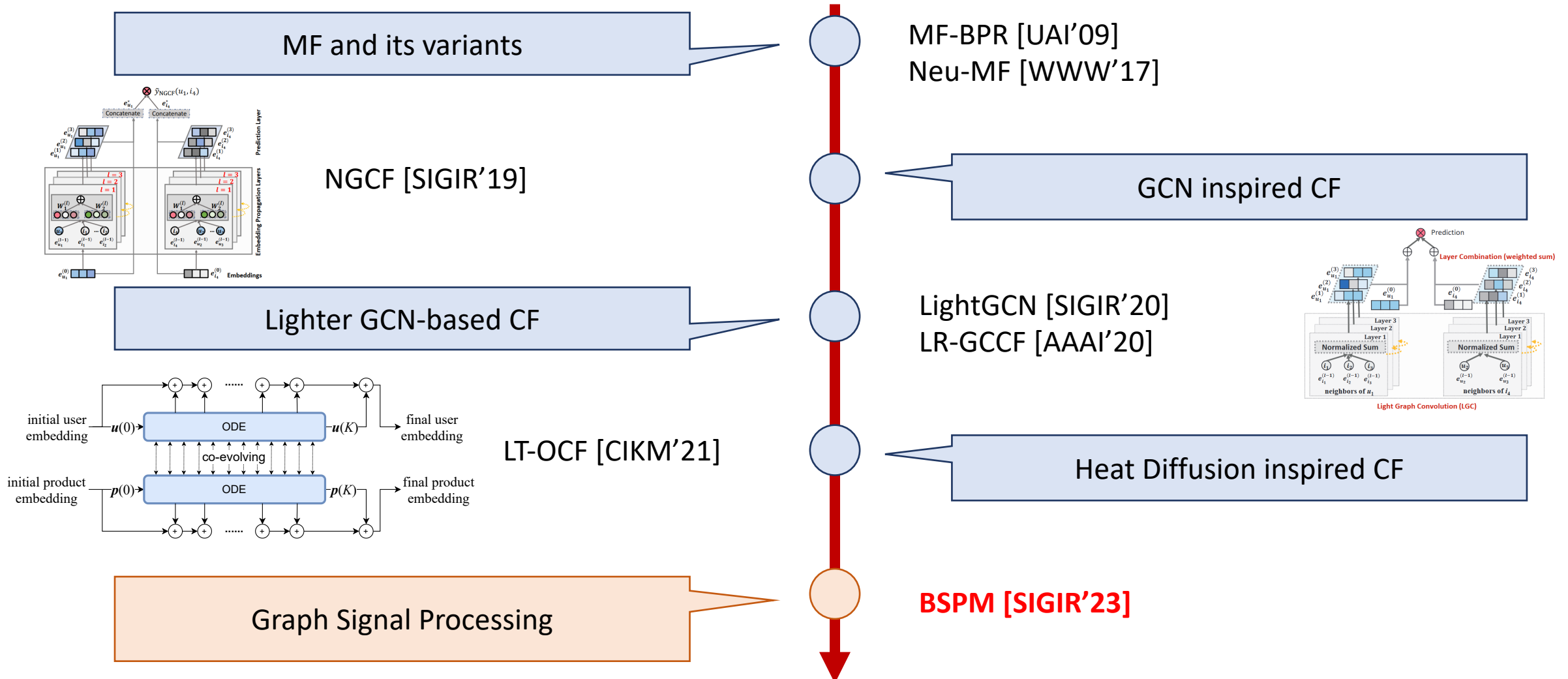
Training curve of t_i in Amazon-book

Experimental Results

- LightGCN achieves significant improvements over the baselines.

Dataset Method	Gowalla		Yelp2018		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	0.1405	0.1221	0.0475	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
Mult-VAE	0.1641	0.1335	0.0584	0.0450	0.0407	0.0315
GRMF	0.1477	0.1205	0.0571	0.0462	0.0354	0.0270
GRMF-Norm	0.1557	0.1261	0.0561	0.0454	0.0352	0.0269
NGCF	0.1570	0.1327	0.0579	0.0477	0.0344	0.0263
LR-GCCF	0.1518	0.1259	0.0574	0.0349	0.0341	0.0258
LightGCN	0.1830	0.1554	0.0649	0.0530	0.0411	0.0315
LT-OCF	0.1875	0.1574	0.0671	0.0549	0.0442	0.0341

A Timeline of Collaborative Filtering (CFs)



Blurring? Low-pass Filters!

- Low-pass Filter



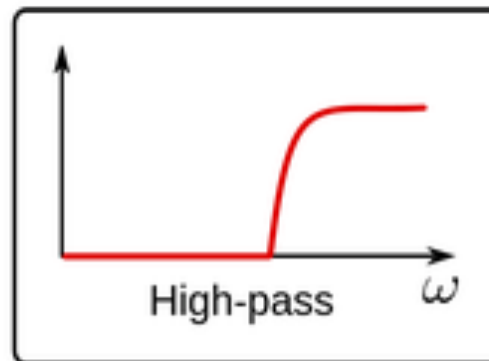
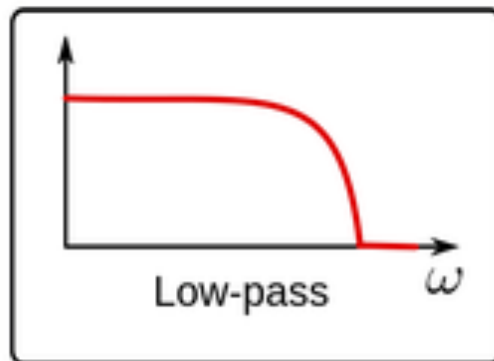
Original Image



Filtered Image

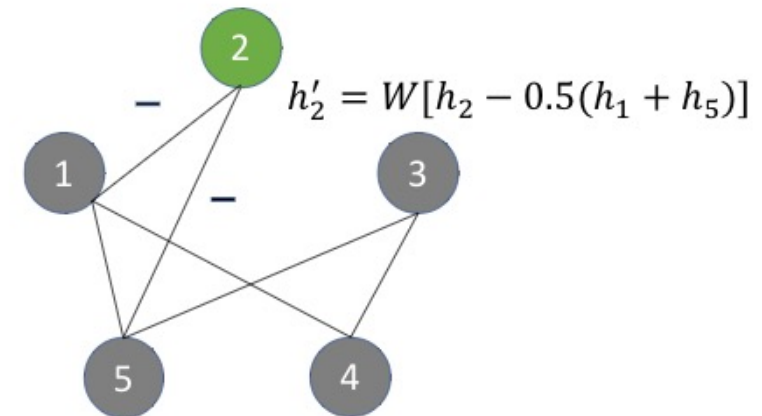
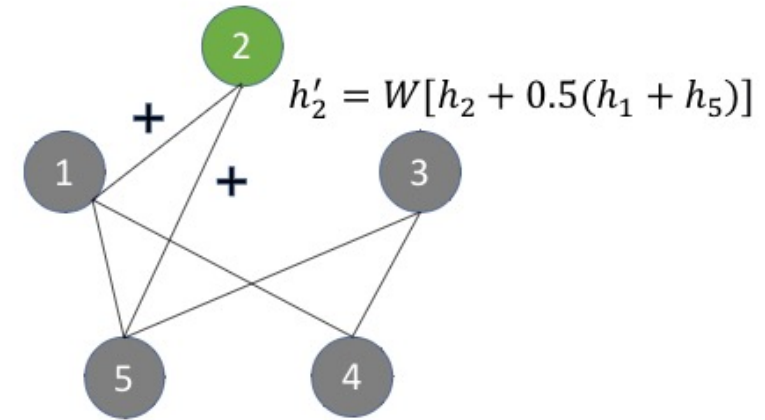
Various Filters

- Low-pass Filter (LPF)
 - passes signals with a lower frequency and attenuates ones with a higher frequency.
- High-pass Filter (HPF)
 - passes signals with a higher frequency and attenuates ones with a lower frequency.



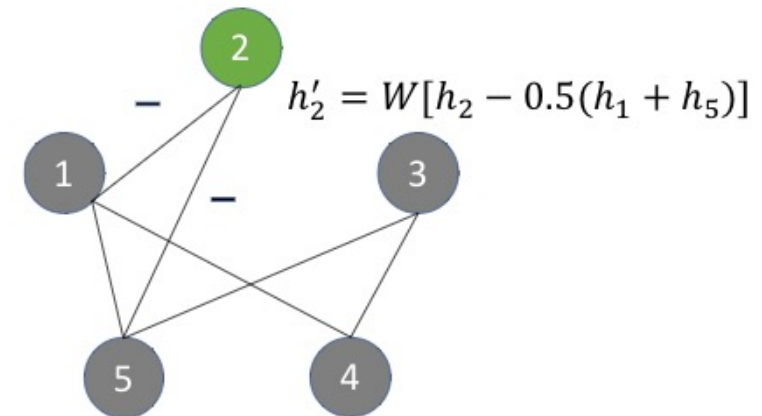
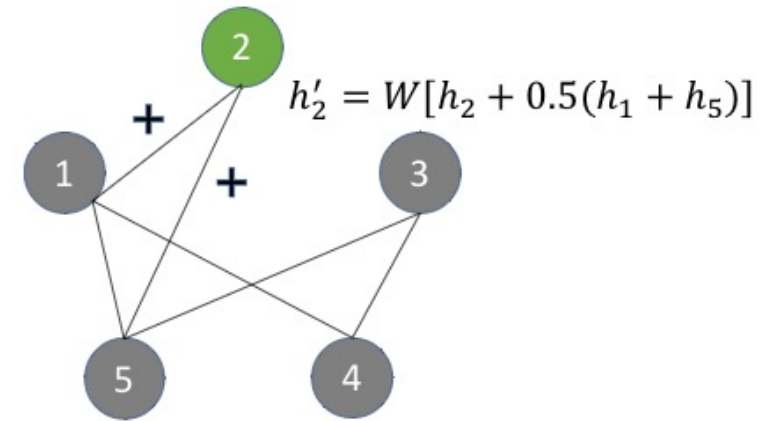
Graph Filter (1/2)

- A low-pass filter smooths the node self-information by taking the average of its neighboring information into account.
- A high-pass filter computes the difference between the node self-information with its neighboring information.



Graph Filter (1/2)

- A low-pass filter smooths the node self-information by taking the average of its neighboring information into account.
- A high-pass filter computes the difference between the node self-information with its neighboring information.



Graph Filter (2/2)

- $L = U\Lambda U^T$
- Given a graph Laplacian matrix L , as well as its eigenvectors U and eigenvalues Λ , then the graph filter $H(L)$ is defined as follows:

$$H(L) = U \text{Diag}(h(\lambda_1), \dots, h(\lambda_n)) U^T$$

where $h(\cdot)$ is the filter defined on the eigenvalues.

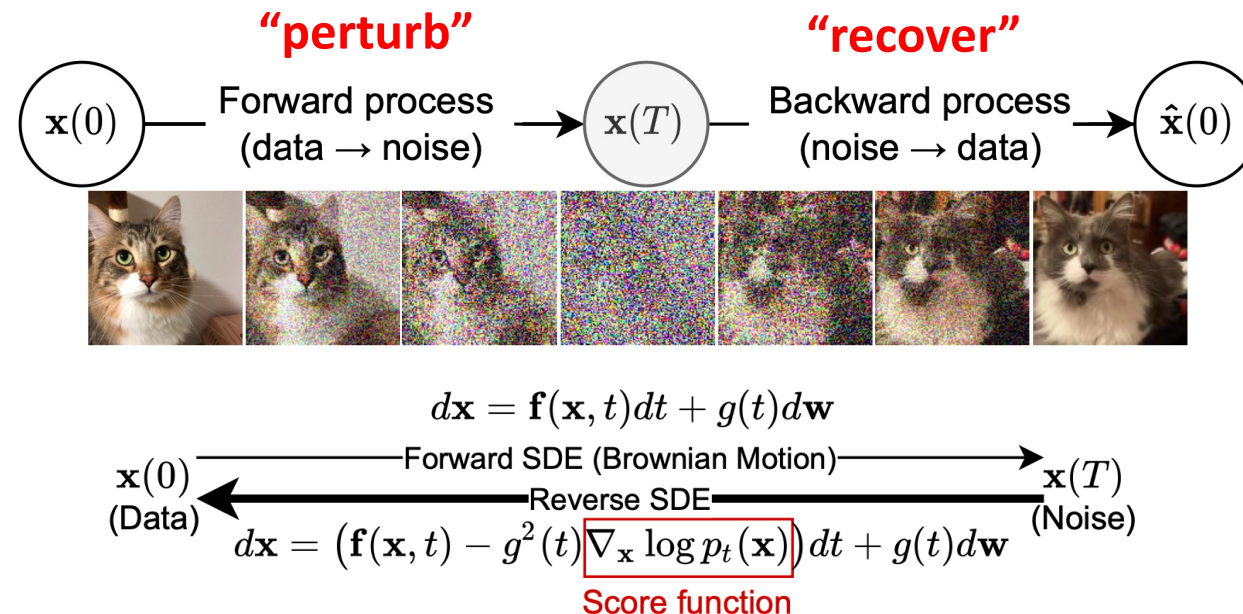
Existing methods

- Comparison of existing methods.
 - Our BSPMs not only combine the blurring and the sharpening processes but also interpret them in a continuous time domain.

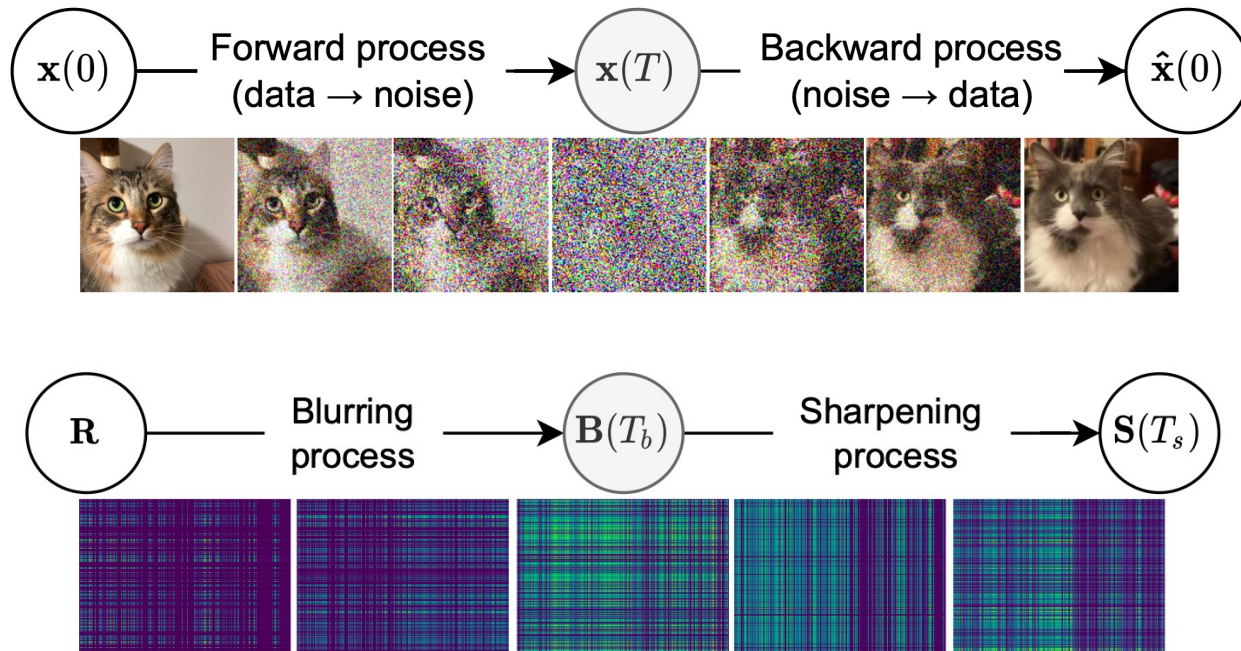
Model	Blurring	Sharpening
LightGCN	Discrete heat equation	X
LT-OCF	Continuous heat equation	X
BSPM	Continuous filters	A continuous filter

Score-based Generative Models (SGMs)

- Score-based generative models (SGMs) [Song *et al.*, 2021]
 - Blurring and Sharpening Process Models (BSPM) is greatly inspired by the recent successes of score-based models for deep generative tasks.



SGMs and BSPMs

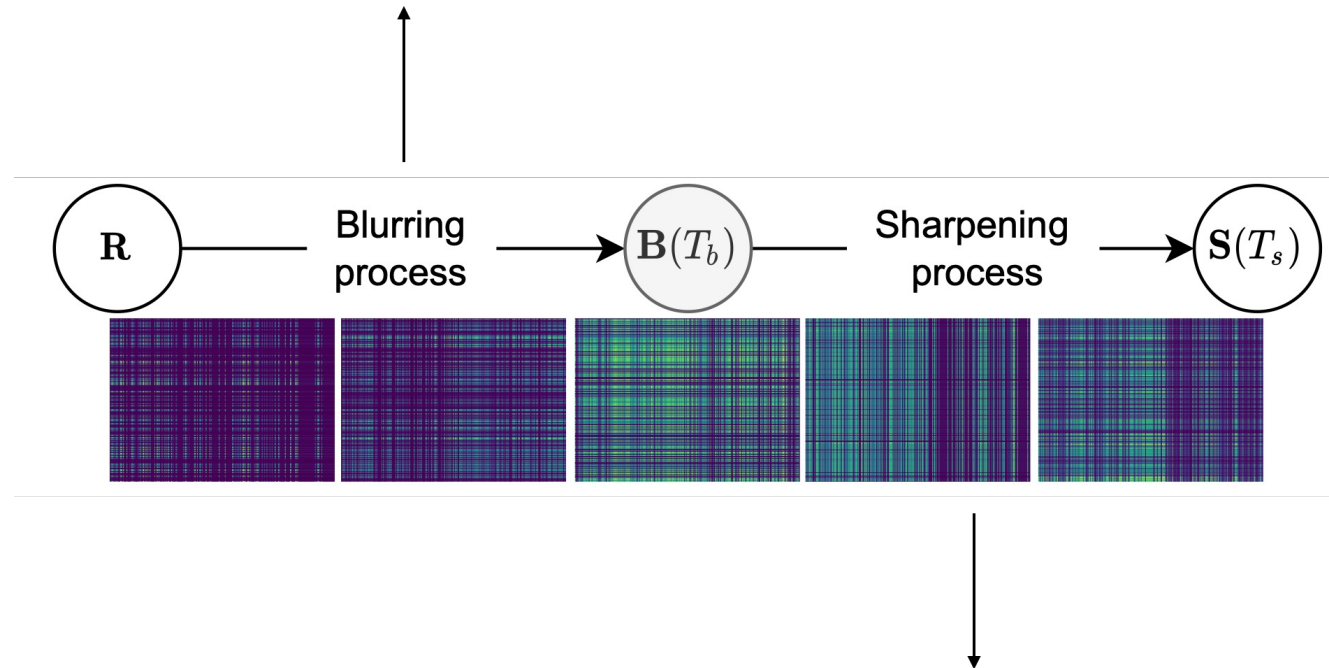


The comparison between SGMs and our BSPMs

	SGM	Our proposed BSPM
Type	SDEs	ODEs
Perturbation	Adding noises to images	Blurring interaction matrix
Recovery	Denoising images	Sharpening blurred matrix
Data	Many images	One interaction matrix
What to Learn	Score function	N/A

Blurring and Sharpening Process Models (BSPM)

The **blurring** process is a core of collaborative filtering.
In general, popular items are recommended to users after this process.



The **sharpening** process increases the accuracy while user-specific, less popular items are also recommended. (* it needs to be sophisticatedly designed)

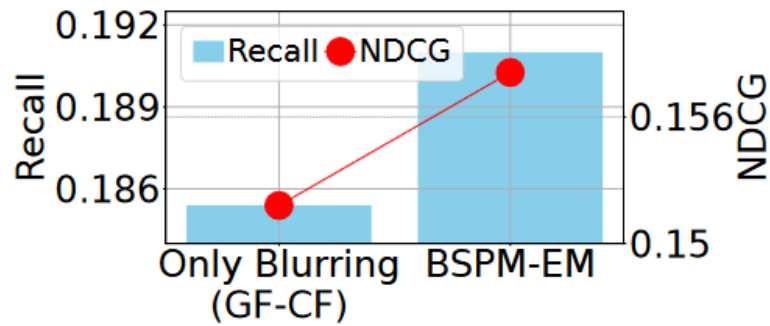
Experimental Results

Model	Gowalla		Yelp2018		Amazon-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
LightGCN	0.1830	0.1554	0.0649	0.0530	0.0411	0.0315
SGL-ED	0.1835	0.1539	0.0675	0.0555	0.0478	0.0379
DeosGCF	0.1784	0.1477	0.0626	0.0504	0.0410	0.0316
IMP-GCN	0.1845	0.1567	0.0653	0.0531	0.0460	0.0357
BUIR _{NB}	0.1575	0.1301	0.0647	0.0526	0.0439	0.0346
DGCF	0.1842	0.1561	0.0654	0.0534	0.0422	0.0324
IA-GCN	0.1839	0.1562	0.0659	0.0537	0.0472	0.0373
UltraGCN	0.1862	0.1580	0.0683	0.0561	0.0681	0.0556
SimpleX	0.1872	0.1557	0.0701	0.0575	0.0583	0.0468
LT-OCF	0.1875	0.1574	0.0671	0.0549	0.0442	0.0341
GF-CF	0.1849	0.1518	0.0697	0.0571	0.0710	0.0584
HMLET	0.1874	<u>0.1589</u>	0.0675	0.0557	0.0482	0.0371
LinkProp	0.1814	0.1477	0.0676	0.0559	0.0684	0.0559
LinkProp-Multi	<u>0.1908</u>	0.1573	0.0690	0.0571	<u>0.0721</u>	0.0588
MGDCF	0.1864	<u>0.1589</u>	0.0696	0.0572	0.0490	0.0378
GTN	0.1870	0.1588	0.0679	0.0554	0.0450	0.0346
BSPM-LM	0.1901	0.1570	<u>0.0713</u>	<u>0.0584</u>	0.0733	0.0610
BSPM-EM	0.1920	0.1597	0.0720	0.0593	0.0733	<u>0.0609</u>

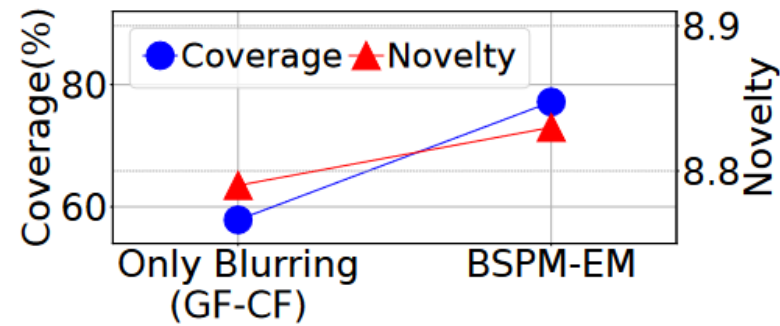
Model	Gowalla		Yelp2018		Amazon-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF-BPR	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
GRMF	0.1477	0.1205	0.0571	0.0462	0.0354	0.0270
GRMF-Norm	0.1557	0.1261	0.0561	0.0454	0.0352	0.0269
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
ENMF	0.1523	0.1315	0.0624	0.0515	0.0359	0.0281
MF-CCL	0.1837	0.1493	0.0698	0.0572	0.0559	0.0447
Multi-VAE	0.1641	0.1335	0.0584	0.0450	0.0407	0.0315
Macrid-VAE	0.1618	0.1202	0.0612	0.0495	0.0383	0.0295
EASE ^R	0.1765	0.1467	0.0657	0.0552	0.0710	0.0567
YouTubenet	0.1754	0.1473	0.0686	0.0567	0.0502	0.0388
CMN	0.1405	0.1221	0.0475	0.0369	0.0267	0.0218
CML	0.1670	0.1292	0.0622	0.0536	0.0522	0.0428
DeepWalk	0.1034	0.0740	0.0476	0.0378	0.0346	0.0264
LINE	0.1335	0.1056	0.0549	0.0446	0.0410	0.0318
Node2Vec	0.1019	0.0709	0.0452	0.0350	0.0402	0.0309
Item2Vec	0.1325	0.1057	0.0503	0.0411	0.0326	0.0251
GAT	0.1401	0.1236	0.0543	0.0431	0.0326	0.0235
JKNet	0.1622	0.1391	0.0608	0.0502	0.0268	0.0343
DropEdge	0.1627	0.1394	0.0614	0.0506	0.0342	0.0270
APPNP	0.1708	0.1462	0.0635	0.0521	0.0384	0.0299
DisenGCN	0.1356	0.1174	0.0558	0.0454	0.0329	0.0254
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF	0.1570	0.1327	0.0579	0.0477	0.0344	0.0263
NIA-GCN	0.1359	0.1106	0.0599	0.0491	0.0369	0.0287
LR-GCCF	0.1701	0.1452	0.0604	0.0498	0.0375	0.0296
LightGCN	0.1830	0.1554	0.0649	0.0530	0.0411	0.0315
SGL-ED	0.1835	0.1539	0.0675	0.0555	0.0478	0.0379
DeosGCF	0.1784	0.1477	0.0626	0.0504	0.0410	0.0316
IMP-GCN	0.1845	0.1567	0.0653	0.0531	0.0460	0.0357
BUIR _{NB}	0.1575	0.1301	0.0647	0.0526	0.0439	0.0346
DGCF	0.1842	0.1561	0.0654	0.0534	0.0422	0.0324
IA-GCN	0.1839	0.1562	0.0659	0.0537	0.0472	0.0373
UltraGCN	0.1862	0.1580	0.0683	0.0561	0.0681	0.0556
SimpleX	0.1872	0.1557	0.0701	0.0575	0.0583	0.0468
LT-OCF	0.1875	0.1574	0.0671	0.0549	0.0442	0.0341
GF-CF	0.1849	0.1518	0.0697	0.0571	0.0710	0.0584
HMLET	0.1874	<u>0.1589</u>	0.0675	0.0557	0.0482	0.0371
LinkProp	0.1814	0.1477	0.0676	0.0559	0.0684	0.0559
LinkProp-Multi	<u>0.1908</u>	0.1573	0.0690	0.0571	<u>0.0721</u>	0.0588
MGDCF	0.1864	<u>0.1589</u>	0.0696	0.0572	0.0490	0.0378
GTN	0.1870	0.1588	0.0679	0.0554	0.0450	0.0346
Only Blurring (HE)	0.1682	0.1331	0.0684	0.0565	0.0710	0.0584
Only Blurring (IDL)	0.1776	0.1489	0.0668	0.0549	0.0395	0.0316
Only Blurring (GF-CF)	0.1854	0.1518	0.0701	0.0575	0.0710	0.0584
BSPM-LM	0.1901	0.1570	<u>0.0713</u>	<u>0.0584</u>	0.0733	0.0610
BSPM-EM	0.1920	0.1597	0.0720	0.0593	0.0733	<u>0.0609</u>
Relative Improvement	0.63%	0.50%	2.71%	3.13%	1.66%	3.74%

Efficacy of the Sharpening Process

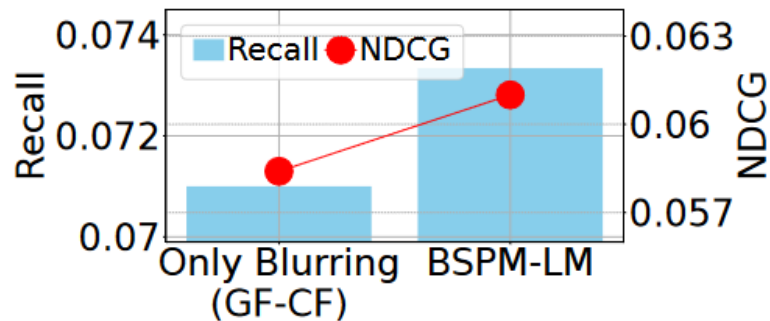
- Additional use of coverage and novelty evaluation indicators



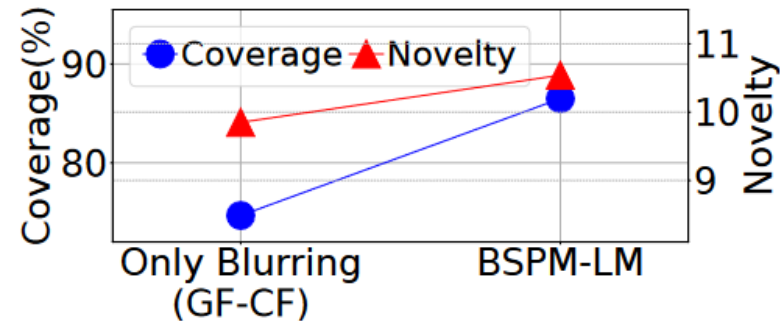
(a) Gowalla



(b) Gowalla



(e) Amazon-book



(f) Amazon-book

Novelty – “New to you”



Meaghan - TeamYouTube



Community Manager • Original Poster

Oct 26, 2021



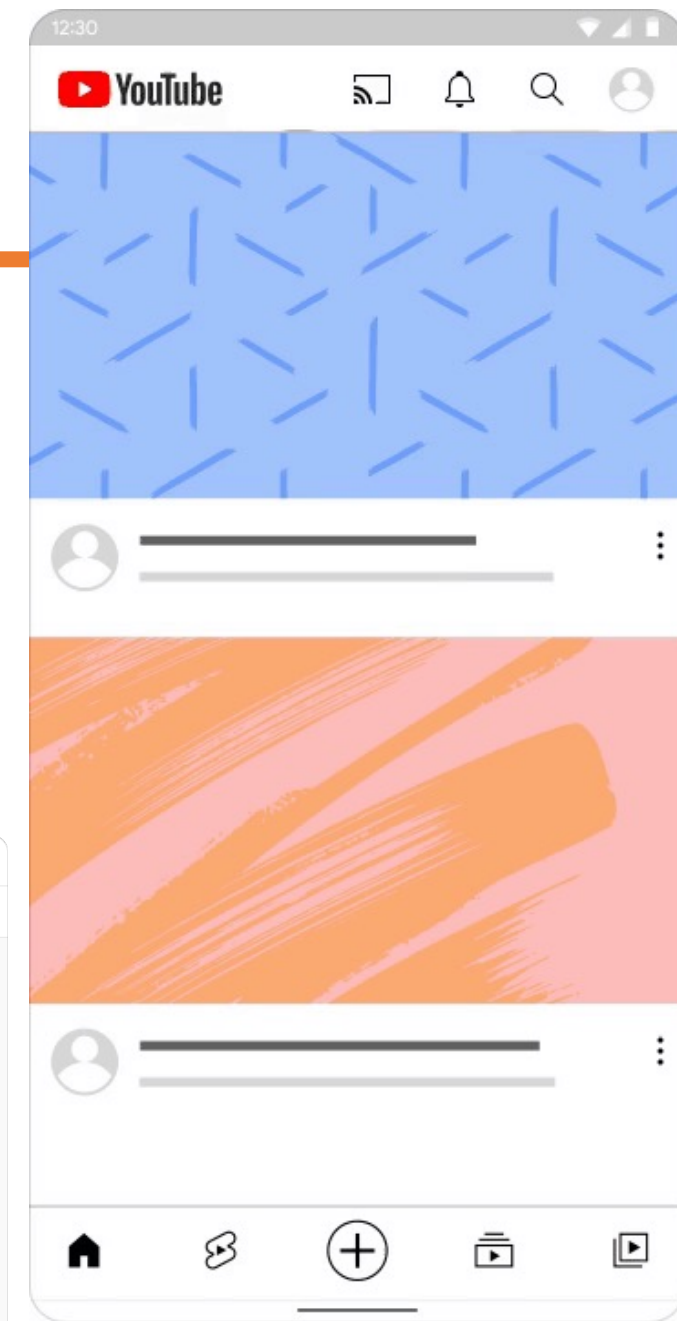
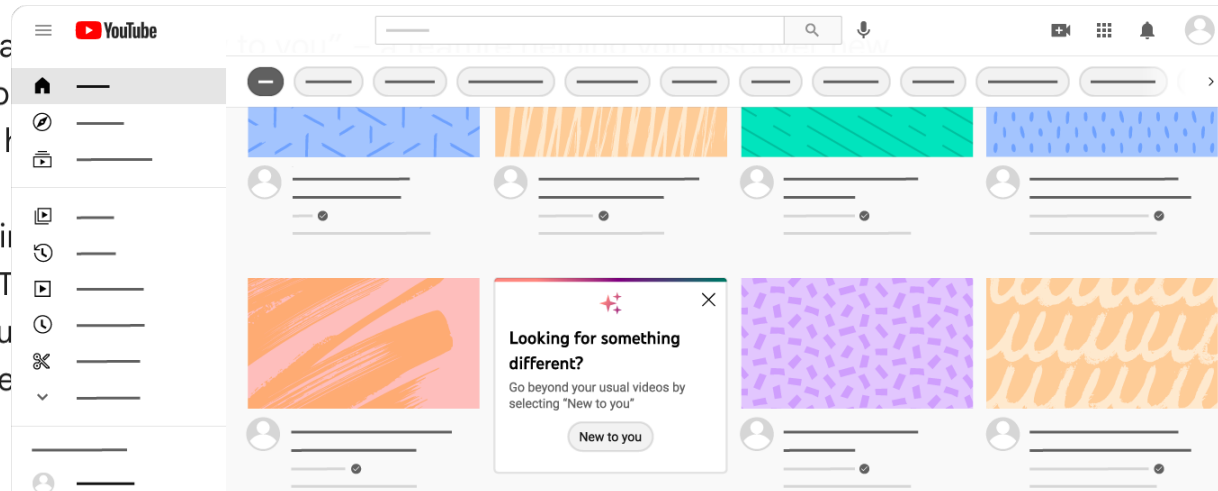
"New to you" - Discover more content on the YouTube homepage

Announcement

Hi everyone!

We're excited to share more details about the "New to you" feature, which helps creators and fresh content, beyond what you usually see, is now available on the YouTube homepage.

Try out "New to you" today: simply scroll through your homepage and you'll find it. The "New to you" prompt as you scroll through your homepage will always be available, and you'll never miss it.



Empirical Model Complexity

- Since our method does not include any training step, it is much faster than other methods.

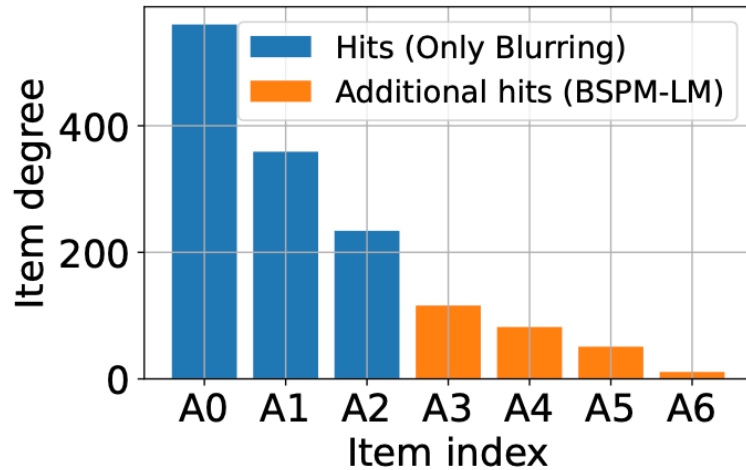
The training time of LightGCN, and the pre-processing time of our method

Model	Gowalla	Yelp2018	Amazon-book
Training of LightGCN	$1.0 \times 10^4\text{s}$	$1.5 \times 10^4\text{s}$	$9.7 \times 10^4\text{s}$
Training of LT-OCF	$3.6 \times 10^4\text{s}$	$4.8 \times 10^4\text{s}$	$2.3 \times 10^5\text{s}$
Pre-processing of BSPM	35.4s	42.3s	101.6s

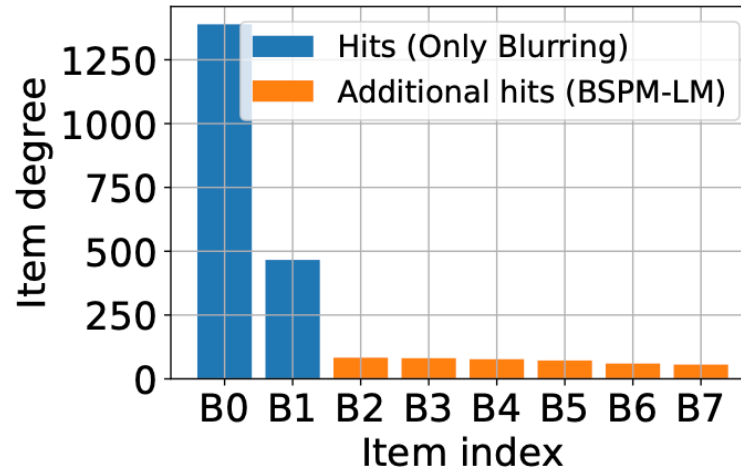
The inference time of BSPM and GF-CF

Model	Gowalla	Yelp2018	Amazon-book
GF-CF	9.8s	10.6s	40.1s
BSPM-EM (Euler)	10.5s	11.0s	53.0s
BSPM-EM (RK4)	17.9s	19.8s	125.5s
BSPM-LM (Euler)	10.6s	11.2s	59.3s
BSPM-LM (RK4)	17.6s	19.6s	127.2s

Case Studies

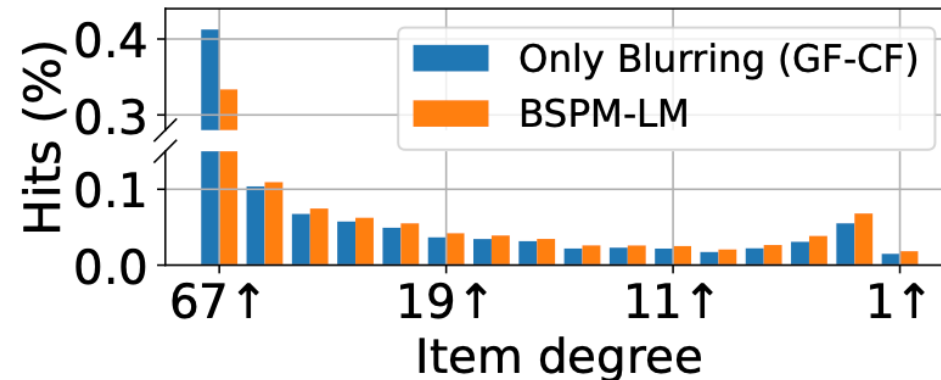


(a) User A



(b) User B

Node degrees of hits (correctly recommended items) on Amazon-book

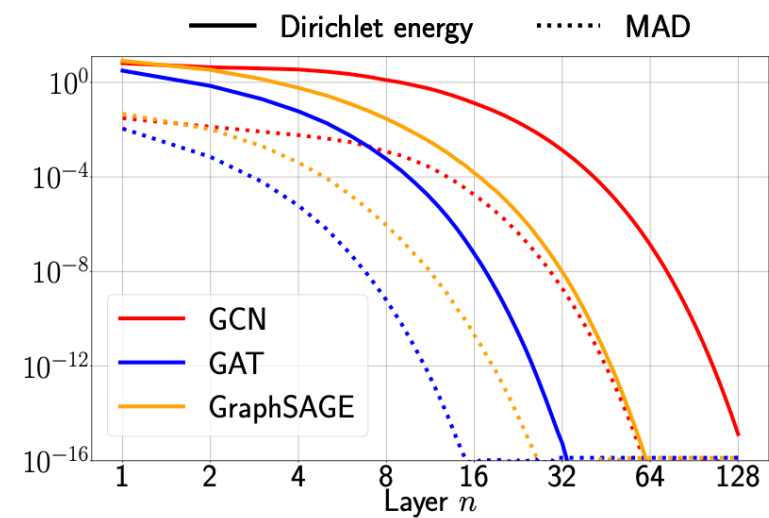
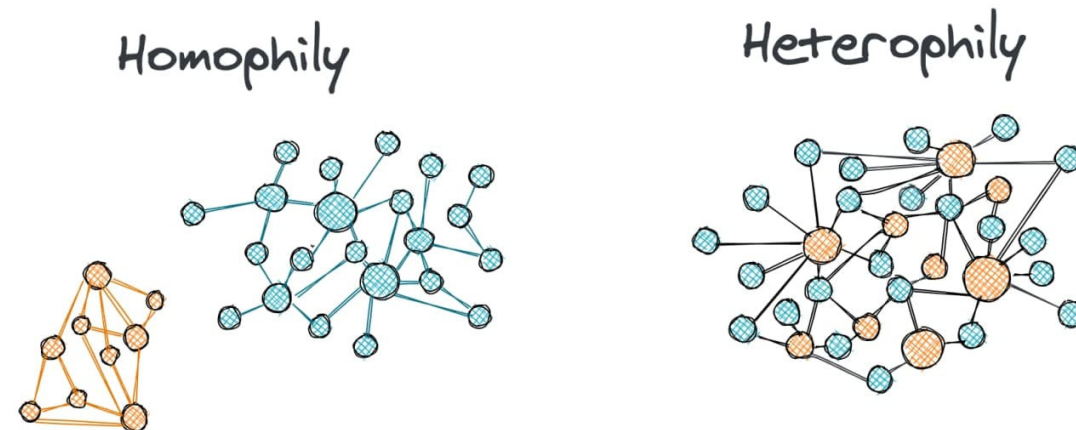
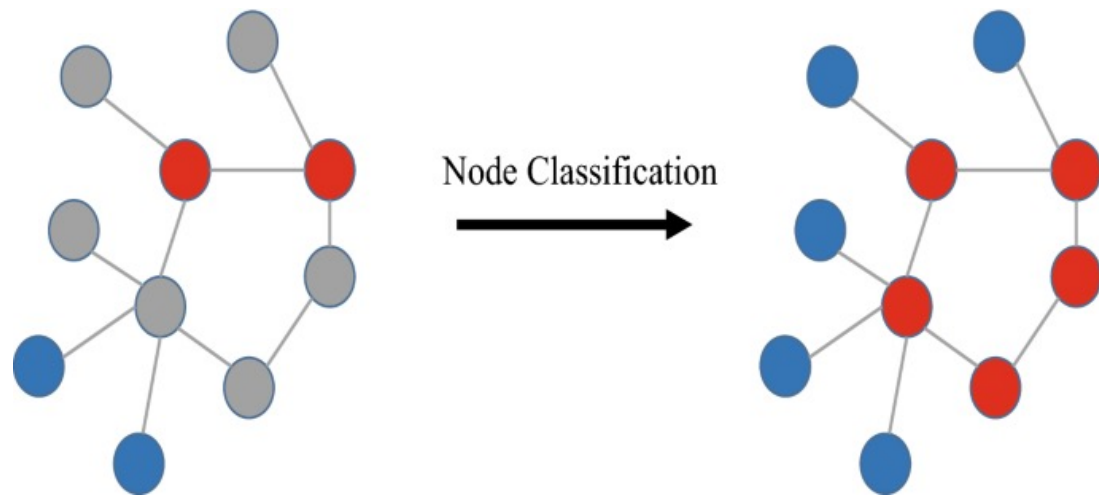


Histogram of Amazon-book's node degrees of correctly recommended items (i.e., hits).

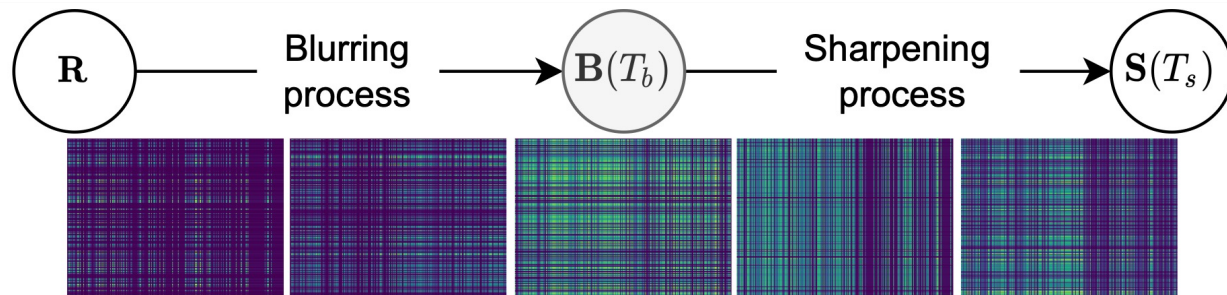
Contents

- Introduction
- Graph Neural Networks
- Diffusion Process on Graphs
- Graph Neural Networks Inspired by Diffusion Process
- **Applications**
 - Recommender Systems
 - **Node Classification**
- Conclusions

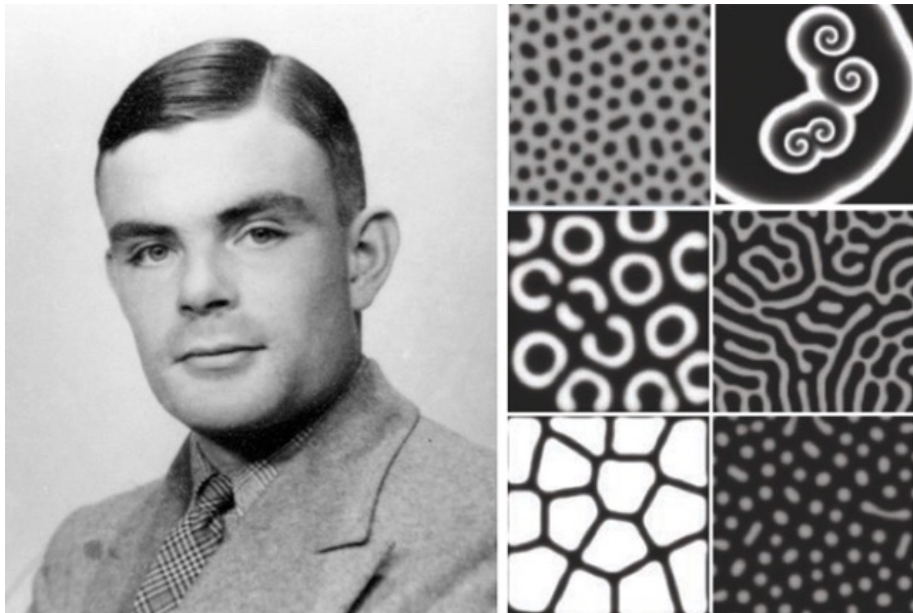
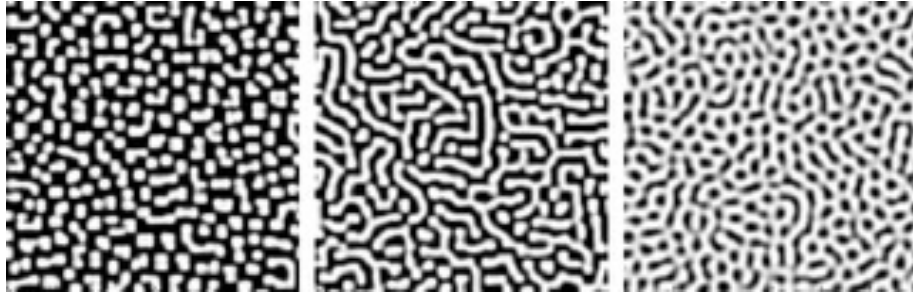
Node Classification



What If We Repeatedly Blurring & Sharpening?



Turing Pattern



THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. *University of Manchester*

(Received 9 November 1951—Revised 15 March 1952)

It is suggested that a system of chemical substances, called morphogens, reacting together and diffusing through a tissue, is adequate to account for the main phenomena of morphogenesis. Such a system, although it may originally be quite homogeneous, may later develop a pattern or structure due to an instability of the homogeneous equilibrium, which is triggered off by random disturbances. Such reaction-diffusion systems are considered in some detail in the case of an isolated ring of cells, a mathematically convenient, though biologically unusual system. The investigation is chiefly concerned with the onset of instability. It is found that there are six essentially different forms which this may take. In the most interesting form stationary waves appear on the ring. It is suggested that this might account, for instance, for the tentacle patterns on *Hydra* and for whorled leaves. A system of reactions and diffusion on a sphere is also considered. Such a system appears to account for gastrulation. Another reaction system in two dimensions gives rise to patterns reminiscent of dappling. It is also suggested that stationary waves in two dimensions could account for the phenomena of phyllotaxis.

Reaction-diffusion Equation

- The reaction-diffusion equations can be written as:

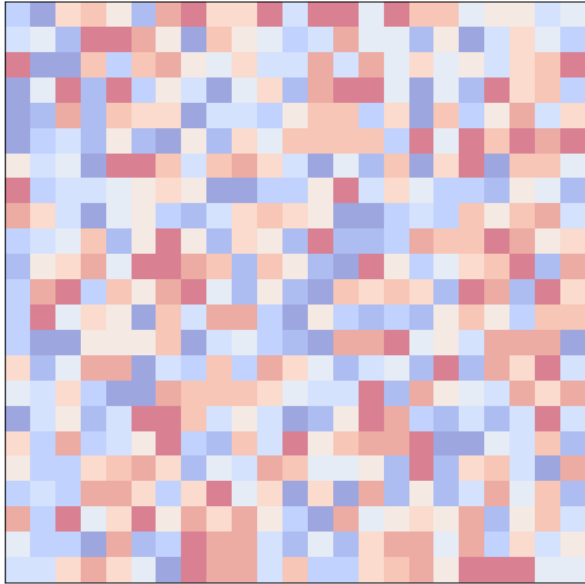
$$\frac{\partial \mathbf{u}}{\partial t} = D \nabla^2 \mathbf{u} + R(\mathbf{u}).$$

- Using the Euler scheme, we can rewrite reaction-diffusion equation:

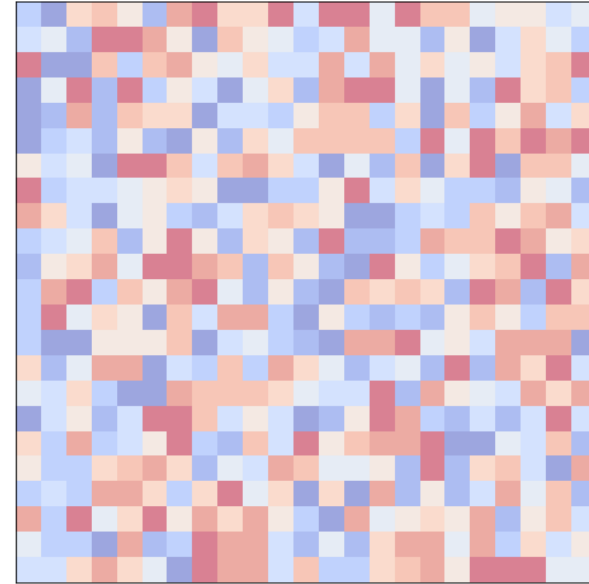
$$\mathbf{u}(t + \Delta t) = \mathbf{u}(t) + \Delta t \left(-D \mathbf{L} \mathbf{u}(t) + R(\mathbf{u}(t)) \right).$$

An Illustrative Comparison

- Diffusion equation



- Reaction-diffusion equation



Graph Reaction-Diffusion Networks

- Graph Reaction-Diffusion Networks (GREAD):

$$\begin{aligned} \mathbf{X}(T) &= \mathbf{X}(0) + \int_0^T f(\mathbf{X}(t))dt, \\ f(\mathbf{X}(t)) &:= -\mathbf{L}\mathbf{X}(t) + \mathbf{R}(\mathbf{X}(t)). \end{aligned}$$

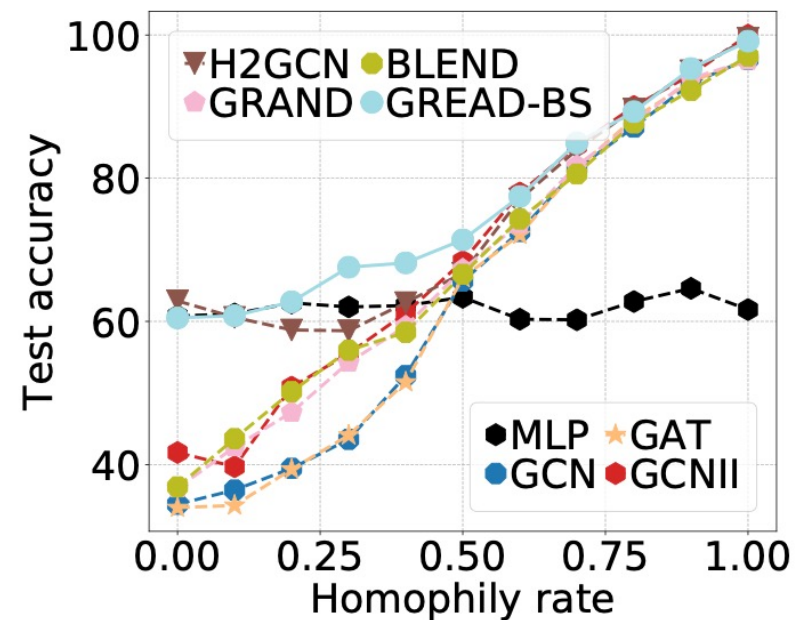
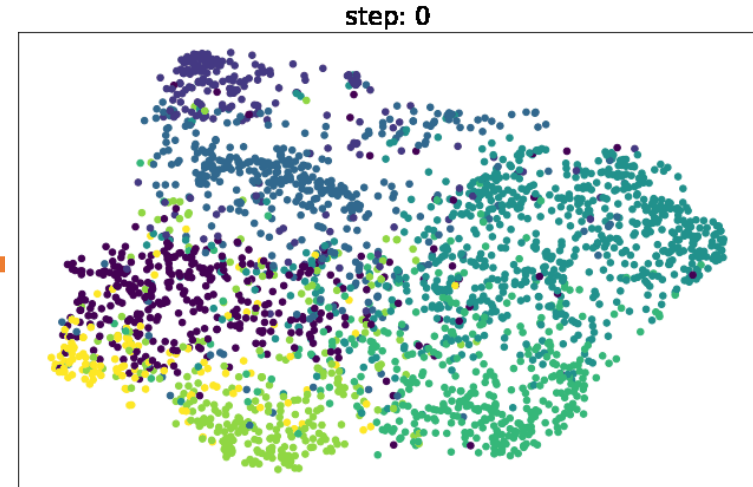
Table 1. A comparison table of existing methods. ‘ Δ ’ means that it corresponds to a specific type of reaction only.

Model	Diffusion	Reaction	Continuous-time
FA-GCN	O	Δ	X
GPR-GNN	O	Δ	X
ACM-GCN	O	Δ	X
CGNN	O	X	O
GRAND	O	X	O
BLEND	O	X	O
GREAD	O	O	O

Node Classification

Table 4. Results on real-world datasets: mean $\pm$ std. dev. accuracy for 10 different data splits. We show the best three methods in **red** (first), **blue** (second), and **purple** (third). Other missing 16 baselines are in Appendix B.

Dataset	Texas	Wisconsin	Cornell	Film	Squirrel	Chameleon	Cora	Citeseer	PubMed	Avg.
Geom-GCN	66.76 $\pm$ 2.72	64.51 $\pm$ 3.66	60.54 $\pm$ 3.67	31.59 $\pm$ 1.15	38.15 $\pm$ 0.92	60.00 $\pm$ 2.81	85.35 $\pm$ 1.57	78.02$\pm$1.15	89.95 $\pm$ 0.47	63.87
H2GCN	84.86 $\pm$ 7.23	87.65 $\pm$ 4.98	82.70 $\pm$ 5.28	35.70 $\pm$ 1.00	36.48 $\pm$ 1.86	60.11 $\pm$ 2.15	87.87 $\pm$ 1.20	77.11 $\pm$ 1.57	89.49 $\pm$ 0.38	71.33
GGCN	84.86 $\pm$ 4.55	86.86 $\pm$ 3.29	85.68 $\pm$ 6.63	37.54 $\pm$ 1.56	55.17 $\pm$ 1.58	71.14$\pm$1.84	87.95 $\pm$ 1.05	77.14 $\pm$ 1.45	89.15 $\pm$ 0.37	75.05
LINKX	74.60 $\pm$ 8.37	75.49 $\pm$ 5.72	77.84 $\pm$ 5.81	36.10 $\pm$ 1.55	61.81$\pm$1.80	68.42 $\pm$ 1.38	84.64 $\pm$ 1.13	73.19 $\pm$ 0.99	87.86 $\pm$ 0.77	71.11
GloGNN	84.32 $\pm$ 4.15	87.06 $\pm$ 3.53	83.51 $\pm$ 4.26	37.35 $\pm$ 1.30	57.54$\pm$1.39	69.78 $\pm$ 2.42	88.31 $\pm$ 1.13	77.41 $\pm$ 1.65	89.62 $\pm$ 0.35	74.99
ACM-GCN	87.84 $\pm$ 4.40	88.43$\pm$3.22	85.14 $\pm$ 6.07	36.28 $\pm$ 1.09	54.40 $\pm$ 1.88	66.93 $\pm$ 1.85	87.91 $\pm$ 0.95	77.32 $\pm$ 1.70	90.00 $\pm$ 0.52	74.92
GCNII	77.57 $\pm$ 3.83	80.39 $\pm$ 3.40	77.86 $\pm$ 3.79	37.44 $\pm$ 1.30	38.47 $\pm$ 1.58	63.86 $\pm$ 3.04	88.37$\pm$1.25	77.33 $\pm$ 1.48	90.15$\pm$0.43	70.16
CGNN	71.35 $\pm$ 4.05	74.31 $\pm$ 7.26	66.22 $\pm$ 7.69	35.95 $\pm$ 0.86	29.24 $\pm$ 1.09	46.89 $\pm$ 1.66	87.10 $\pm$ 1.35	76.91 $\pm$ 1.81	87.70 $\pm$ 0.49	63.96
GRAND	75.68 $\pm$ 7.25	79.41 $\pm$ 3.64	82.16 $\pm$ 7.09	35.62 $\pm$ 1.01	40.05 $\pm$ 1.50	54.67 $\pm$ 2.54	87.36 $\pm$ 0.96	76.46 $\pm$ 1.77	89.02 $\pm$ 0.51	68.94
BLEND	83.24 $\pm$ 4.65	84.12 $\pm$ 3.56	85.95 $\pm$ 6.82	35.63 $\pm$ 1.01	43.06 $\pm$ 1.39	60.11 $\pm$ 2.09	88.09 $\pm$ 1.22	76.63 $\pm$ 1.60	89.24 $\pm$ 0.42	71.79
Sheaf	85.05 $\pm$ 5.51	89.41$\pm$4.74	84.86 $\pm$ 4.71	37.81$\pm$1.15	56.34 $\pm$ 1.32	68.04 $\pm$ 1.58	86.90 $\pm$ 1.13	76.70 $\pm$ 1.57	89.49 $\pm$ 0.40	75.06
GRAFF	88.38$\pm$4.53	87.45 $\pm$ 2.94	83.24 $\pm$ 6.49	36.09 $\pm$ 0.81	54.52 $\pm$ 1.37	71.08$\pm$1.75	87.61 $\pm$ 0.97	76.92 $\pm$ 1.70	88.95 $\pm$ 0.52	74.92
GREAD-BS	88.92$\pm$3.72	89.41$\pm$3.30	86.49$\pm$7.15	37.90$\pm$1.17	59.22$\pm$1.44	71.38$\pm$1.31	88.57$\pm$0.66	77.60$\pm$1.81	90.23$\pm$0.55	76.64
GREAD-F	89.73$\pm$4.49	86.47 $\pm$ 4.84	86.49$\pm$5.13	36.72 $\pm$ 0.66	46.16 $\pm$ 1.44	65.20 $\pm$ 1.40	88.39 $\pm$ 0.91	77.40 $\pm$ 1.54	90.09 $\pm$ 0.31	74.13
GREAD-AC	85.95 $\pm$ 2.65	86.08 $\pm$ 3.56	87.03$\pm$4.95	37.21 $\pm$ 1.10	45.10 $\pm$ 2.11	65.09 $\pm$ 1.08	88.29 $\pm$ 0.67	77.38 $\pm$ 1.53	90.10 $\pm$ 0.36	73.71
GREAD-Z	87.30 $\pm$ 5.68	86.29 $\pm$ 4.32	85.68 $\pm$ 5.41	37.01 $\pm$ 1.11	46.25 $\pm$ 1.72	62.70 $\pm$ 2.30	88.31 $\pm$ 1.10	77.39 $\pm$ 1.90	90.11 $\pm$ 0.27	73.45
GREAD-ST	81.08 $\pm$ 5.67	86.67 $\pm$ 3.01	86.22$\pm$5.98	37.66 $\pm$ 0.90	45.83 $\pm$ 1.40	63.03 $\pm$ 1.32	88.47$\pm$1.19	77.25 $\pm$ 1.47	90.13$\pm$0.36	72.93
GREAD-FB	86.76 $\pm$ 5.05	87.65 $\pm$ 3.17	86.22$\pm$5.85	37.40 $\pm$ 0.55	50.83 $\pm$ 2.27	66.05 $\pm$ 1.21	88.03 $\pm$ 0.78	77.28 $\pm$ 1.73	90.07 $\pm$ 0.45	74.48
GREAD-FB*	87.03 $\pm$ 3.97	88.04$\pm$1.63	85.95 $\pm$ 5.64	37.70$\pm$0.51	50.57 $\pm$ 1.52	65.83 $\pm$ 1.10	88.01 $\pm$ 0.80	77.42$\pm$1.93	90.08 $\pm$ 0.46	74.51



(a) GREAD-BS vs. baselines

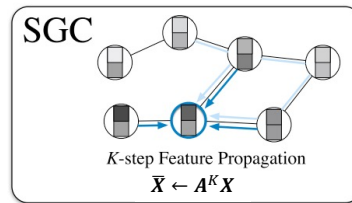
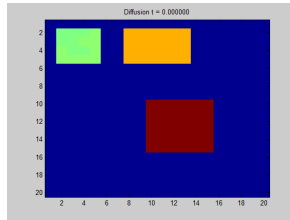
Contents

- Graph Neural Networks
- Diffusion Process on Graphs
- Graph Neural Networks Inspired by Diffusion Process
- Applications
- **Conclusions**

Summary

GNNs Inspired by Diffusion Process

$$\frac{dX(t)}{dt} = -LX(t) \quad \longleftrightarrow \quad X^{(k)} = AX^{(k-1)}$$

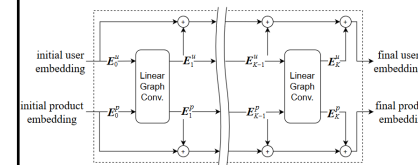


18

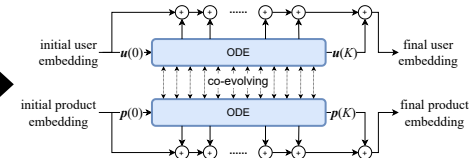
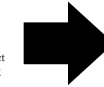
LT-OCF: Learnable-Time ODE-based CF (2/2)

• Why we redesign?

- Linear GCNs can be interpreted as the diffusion equation concept.
- Time variable t is not only *continuous* but also *trainable*.



The architecture of LightGCN

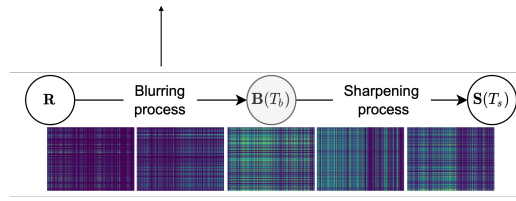


Our proposed LT-OCF

27

Blurring and Sharpening Process Models (BSPM)

The **blurring** process is a core of collaborative filtering.
In general, popular items are recommended to users after this process.

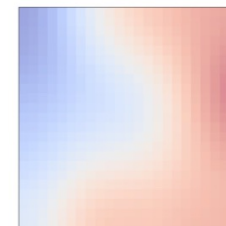


The **sharpening** process increases the accuracy while user-specific, less popular items are also recommended. (* it needs to be sophisticatedly designed)

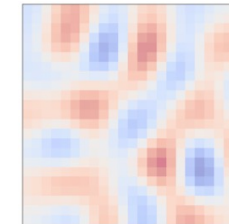
39

An Illustrative Comparison

• Diffusion equation



• Reaction-diffusion equation



50

Thank You

JEONGWHAN.CHOI@YONSEI.AC.KR

<https://jeongwhanchoi.me>

