Graph based Collaborative Filtering and Neural ODEs

Jeongwhan Choi (최정환)
Yonsei University, South Korea

Why?

Continuous-time NN

Neural ODE *(NIPS'18)*

ANODE *(NIPS'19)*

⋮

GCN for Recommender Systems

GCMC *(KDD'18)*

NGCF *(SIGIR'19)*

LightGCN *(SIGIR'20)*

and next?

Why?

Continuous-time NN

Neural ODE *(NIPS'18)*

ANODE *(NIPS'19)*

⋮

GCN for Recommender Systems

GCMC *(KDD'18)*

NGCF *(SIGIR'19)*

LightGCN *(SIGIR'20)*

and next?

LT-OCF: Learnable-Time ODE-based Collaborative Filtering

Jeongwhan Choi, Jinsung Jeon, Noseong Park
{jeongwhan.choi,jjsjjs0902,noseong}@yonsei.ac.kr
Yonsei University
Seoul, South Korea

Why?

Continuous-time NN

Neural ODE *(NIPS’18)*

ANODE *(NIPS’19)*

⋮

GCN for Recommender Systems

GCMC *(KDD’18)*

NGCF *(SIGIR’19)*

LightGCN *(SIGIR’20)*

LT-OCF *(CIKM’21)*

- We show *linear GCNs* can be seen as discretisation of a ODE.
- We redesign *linear GCNs* on top of the *NODE* regime.

Content

- GCN(Graph Convolution Network) based Collaborative Filtering
- Connecting ODEs to GCN based Collaborative Filtering
- LT-OCF
- Demo

GCN based Collaborative Filtering

Recommender Systems

- Building a good recommender remains an active research area.

An example of recommender systems

Collaborative Filtering

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

- GCN based Collaborative Filtering
  - Graph Convolutional Matrix Completion *(KDD'18)*
  - Neural Graph Collaborative Filtering *(SIGIR'19)*
  - LightGCN: Simplyfing annd Powering Graph Convolution Network for Recommendation *(SIGIR'20)*

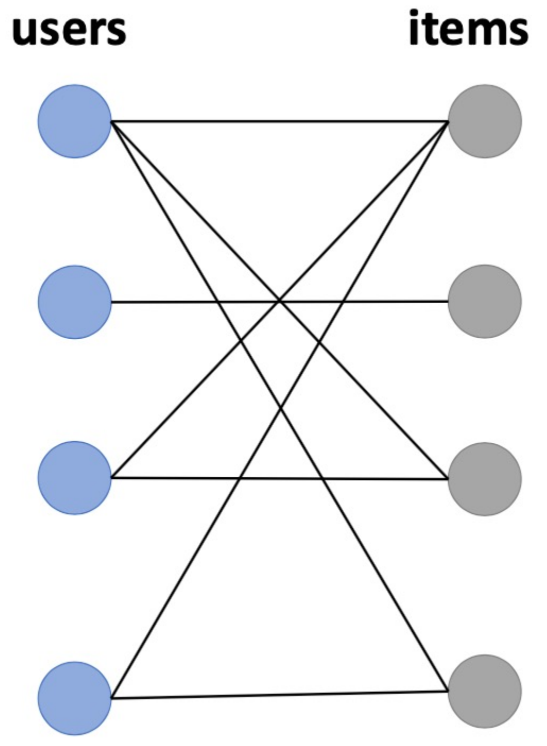
Interactions as Bipartite Graph (1/2)

items

users

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

0/1 Interaction matrix

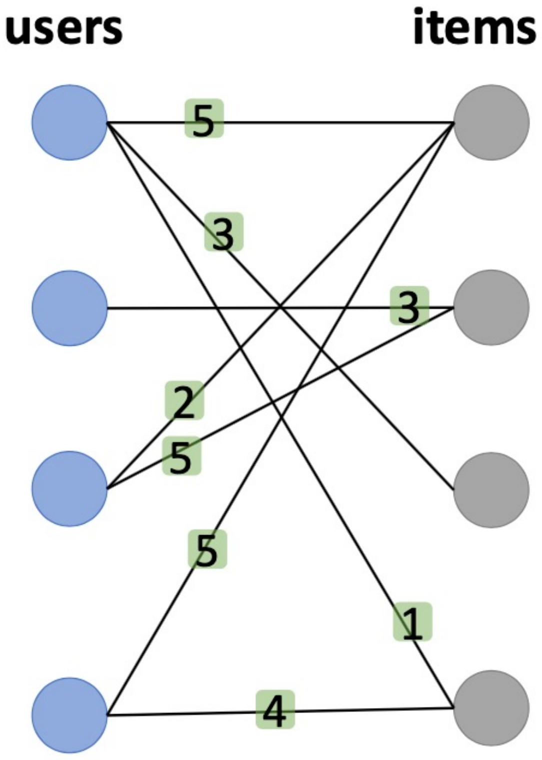


Bipartite Graph

Interactions as Bipartite Graph (2/2)

	items			
users	5	0	3	1
	0	3	0	0
	2	5	0	0
	5	0	0	4

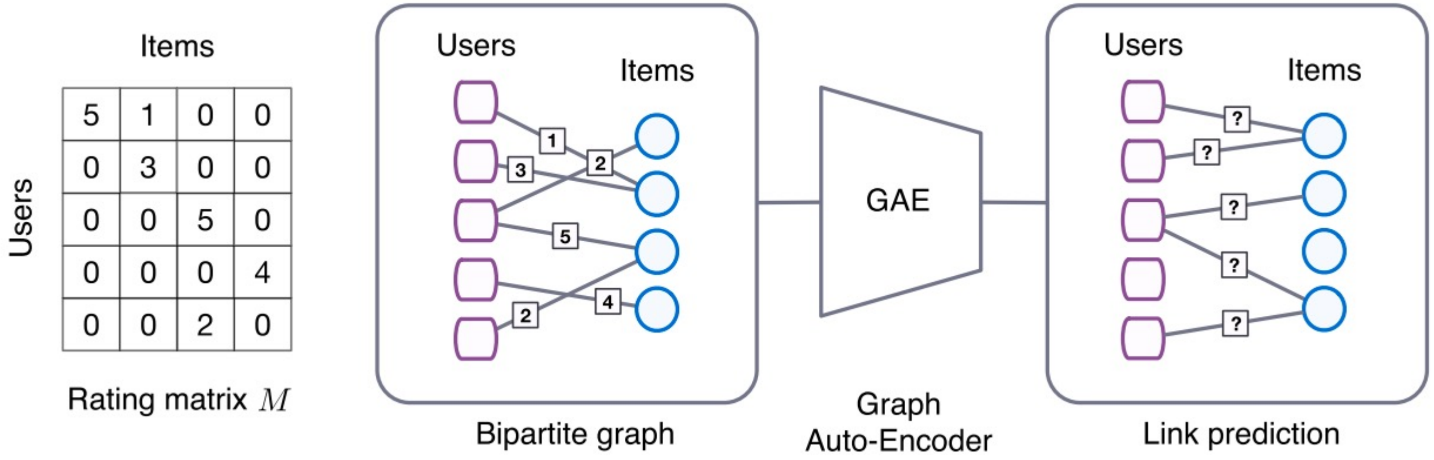
Weighted interaction matrix

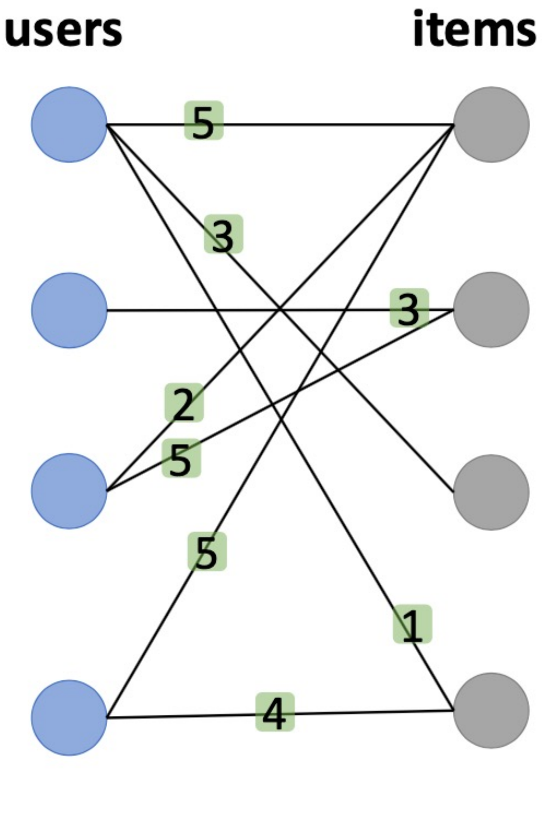


Bipartite Graph

GCMC

- User representation learning
  - Aggregate for each rating



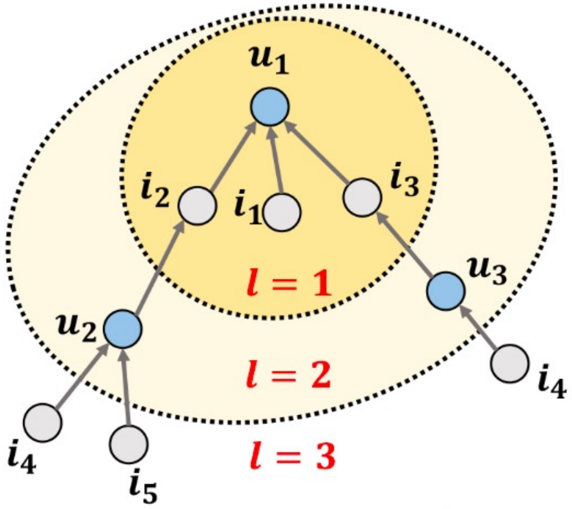


Bipartite Graph

R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” in KDD. 2018.

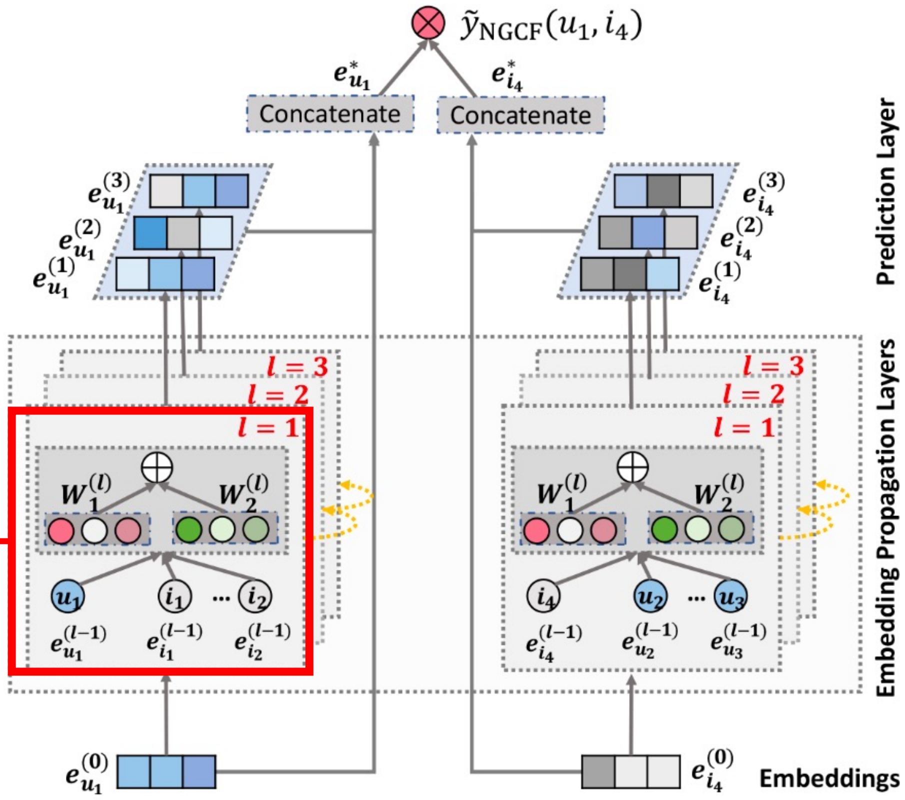
NGCF

- Embedding Propagation, inspired by GCNs
  - Propagate embeddings recursively on the user-item graph
  - Construct information flows in the embedding space



High-order Connectivity for u_1

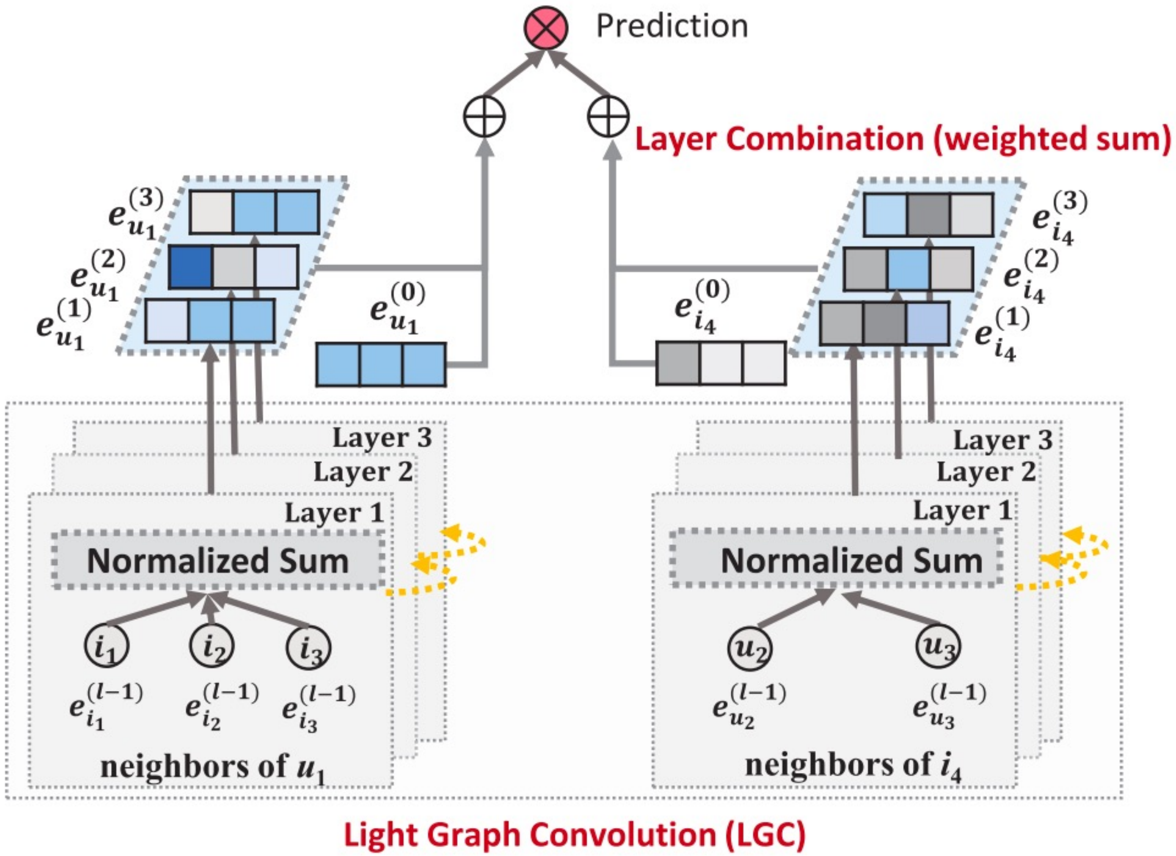
collaborative signal: message passed from interactive items to u



X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, “Neural Graph Collaborative Filtering,” in SIGIR 2019.

LightGCN

- Simplifying GCN for recommendation
  - discard feature transformation and nonlinear activation
  - "linear" GCNs

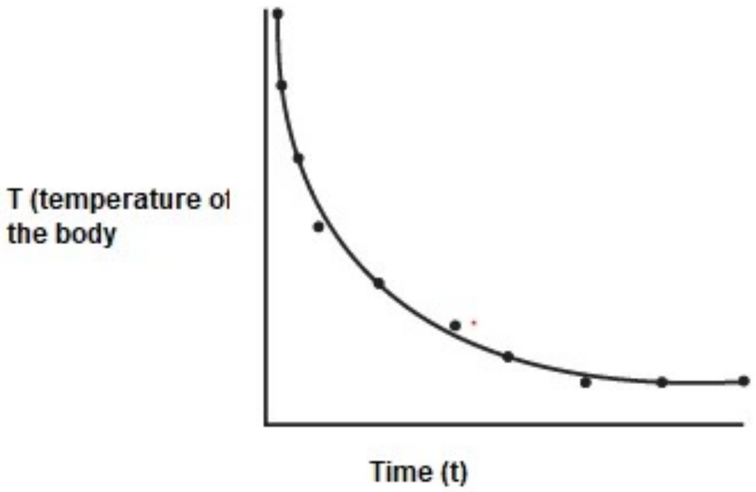


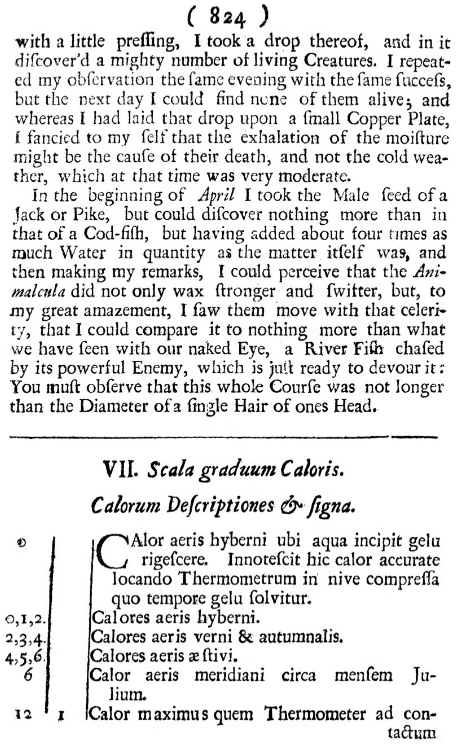
X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in SIGIR 2020.

Connecting ODEs to Graph-based Collaborative Filtering

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 chaſed 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 | CAlor aeris hyberni ubi aqua incipit gelu rigeſcere. Innoteſcit hic calor accurate locando Thermometrum in nive compreſſa quo tempore gelu ſolvitur.
0,1,2. | Calores aeris hyberni.
2,3,4. | Calores aeris verni & autumnalis.
4,5,6. | Calores aeris æſtivi.
6 | Calor aeris meridiani circa menſem Julium.
12 1 | Calor maximus quem Thermometer ad contactum

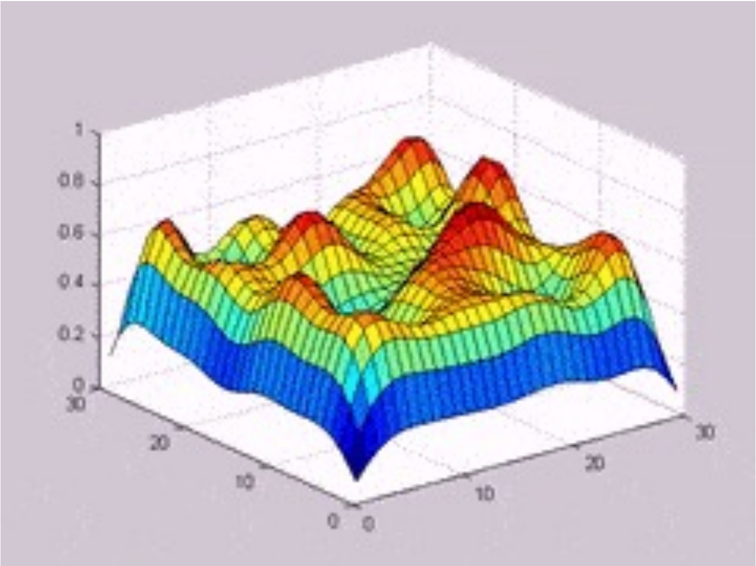
Issac Newton

Anonymous (March–April 1701), "Scala graduum Caloris. Calorum Descriptiones & signa.", *Philosophical Transactions*, **22** (270): 824–829

Diffusion takes many forms

Heat diffusion:

$$\frac{d\boldsymbol{H}_t}{dt} = -\Delta\boldsymbol{H}_t$$

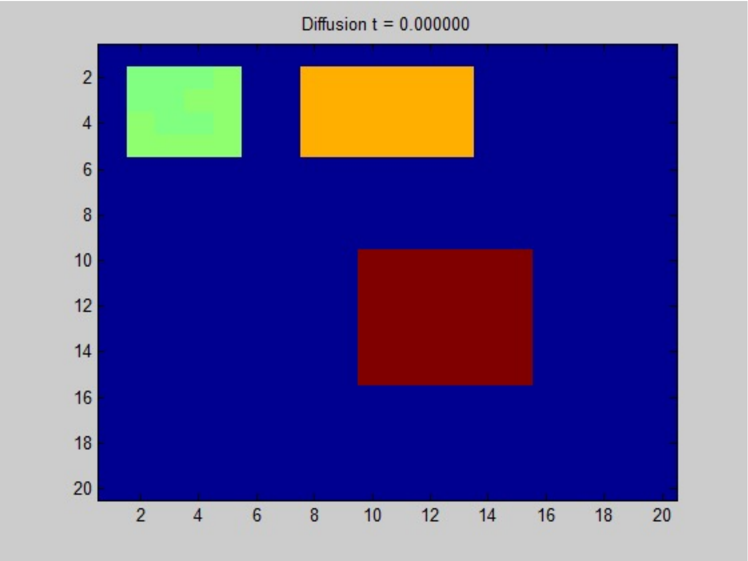


Joseph Fourier

Diffusion takes many forms

Laplacian diffusion on graph:

$$\boldsymbol{H}_t = -\boldsymbol{L}\boldsymbol{X}_{t-1}$$



Pierre-Simon Laplace

Linear GCNs and Newton's Law of Cooling (1/2)

- Linear GCNs of LightGCN are similar to the heat equation
  - ,which describes the law of thermal diffusive processes,
  - i.e, Laplacian diffusion equation.

$$\frac{d\boldsymbol{H}_t}{dt} = -\Delta \boldsymbol{H}_t = -\boldsymbol{L}\boldsymbol{H}_t$$

$$\Updownarrow$$

$$\boldsymbol{E}_k = \boldsymbol{A}\boldsymbol{E}_{k-1}$$

Appendix in LT-OCF

$$\boldsymbol{L} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \tilde{\boldsymbol{A}})\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \tilde{\boldsymbol{A}} \tag{17}$$

is the symmetrically normalized graph Laplacian operator.

When applying the Euler discretization to Eq. (16) with an interval step size $dt = \frac{K}{T+1}$, we have

$$\begin{aligned} \boldsymbol{H}_{t+dt} &= \boldsymbol{H}_t - dt\boldsymbol{L}\boldsymbol{H}_t \\ &= \boldsymbol{H}_t - dt(\boldsymbol{I} - \tilde{\boldsymbol{A}})\boldsymbol{H}_t \\ &= [(1-dt)\boldsymbol{I} + dt\tilde{\boldsymbol{A}}]\boldsymbol{H}_t. \end{aligned} \tag{18}$$

We will get the following final $\boldsymbol{H}_K$, if we keep evolving the ODE until the terminal time $K = dt(T+1)$.

$$\boldsymbol{H}_K = [\tilde{\boldsymbol{A}}^{dt}]^{T+1}\boldsymbol{H}. \tag{19}$$

We regard that LightGCN corresponds to the Euler discretization with a large step size $dt = 1$. This step size reduces the diffusion matrix to the Linear GCN diffusion matrix $\tilde{\boldsymbol{A}}$.

$$\tilde{\boldsymbol{A}}^{(dt)}|_{dt=1} = (1-1)\boldsymbol{I} + \tilde{\boldsymbol{A}} = \tilde{\boldsymbol{A}} \tag{20}$$

and the final $\boldsymbol{H}$ becomes equivalent:

$$\boldsymbol{H}_K|_{dt=1} = \boldsymbol{H}_{T+1} = \boldsymbol{H}^{(T+1)} = \tilde{\boldsymbol{A}}^{T+1}\boldsymbol{H}. \tag{21}$$

Then we can consider $\boldsymbol{H}$ as a product or user embedding, because we interpret each element of $\boldsymbol{E}_i^u$ and $\boldsymbol{E}_i^p$ as a temperature value. Therefore, we take the LightGCN model from the continuous thermal diffusive processes.

Linear GCNs and Newton’s Law of Cooling (2/2)

- Linear GCNs of LightGCN are similar to the heat equation
  - ,which describes the law of thermal diffusive processes,
  - i.e, Laplacian diffusion equation.

$$\frac{d\boldsymbol{H}_t}{dt} = -\Delta \boldsymbol{H}_t = -\boldsymbol{L}\boldsymbol{H}_t \quad \Longleftrightarrow \quad \boldsymbol{E}_k = \boldsymbol{A}\boldsymbol{E}_{k-1}$$

Appendix in LT-OCF

- LightGCN:
  - discrete thermal diffusive processes.
- LT-OCF:
  - ***continuous*** thermal diffusive processes.

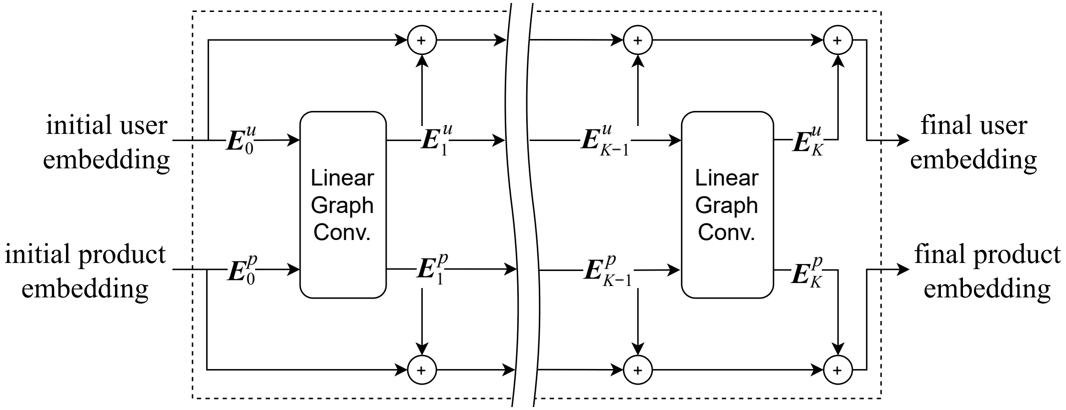
Motivation

- We redesign the *linear GCN* on top of the concept of Neural Ordinary Differential Equations (NODEs).
  - e.g., diffusion equation

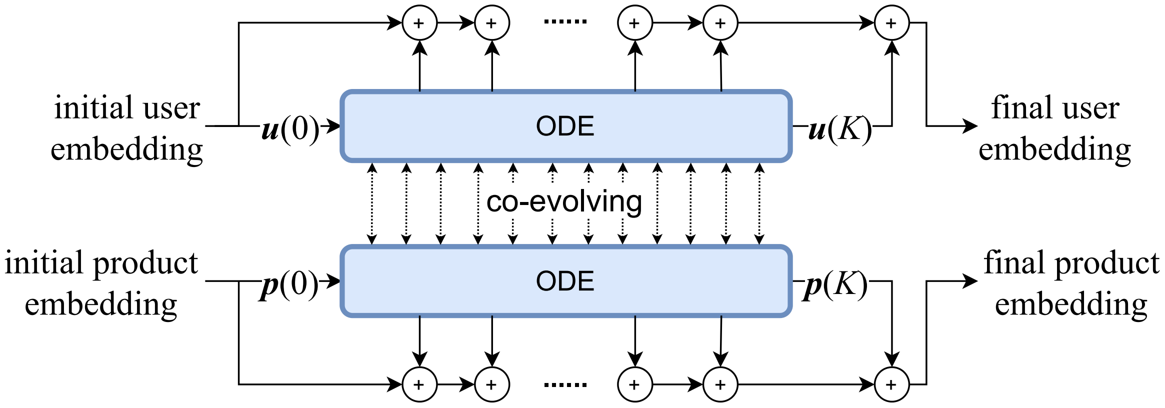
LT-OCF: Learnable-Time ODE-based Collaborative Filtering

Redesign the *linear GCN*

- We redesign the *linear GCN* on top of the concept of Neural Ordinary Differential Equations (NODEs).
- Why we redesign?
  - Linear GCNs can be interpreted as the diffusion equation concept.
  - Time variable t is not only ***continuous*** but also ***trainable***.

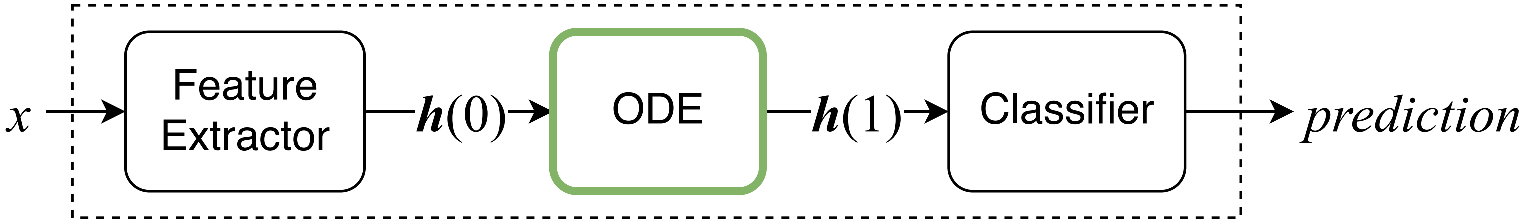


The architecture of LightGCN

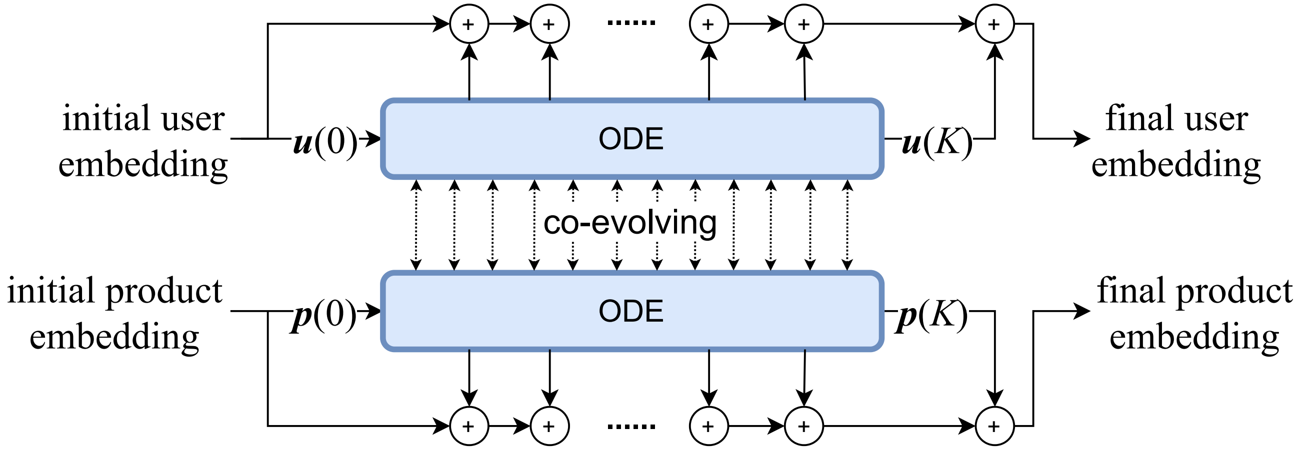


Our proposed LT-OCF

Neural Ordinary Differential Equations



The typical architecture of NODEs

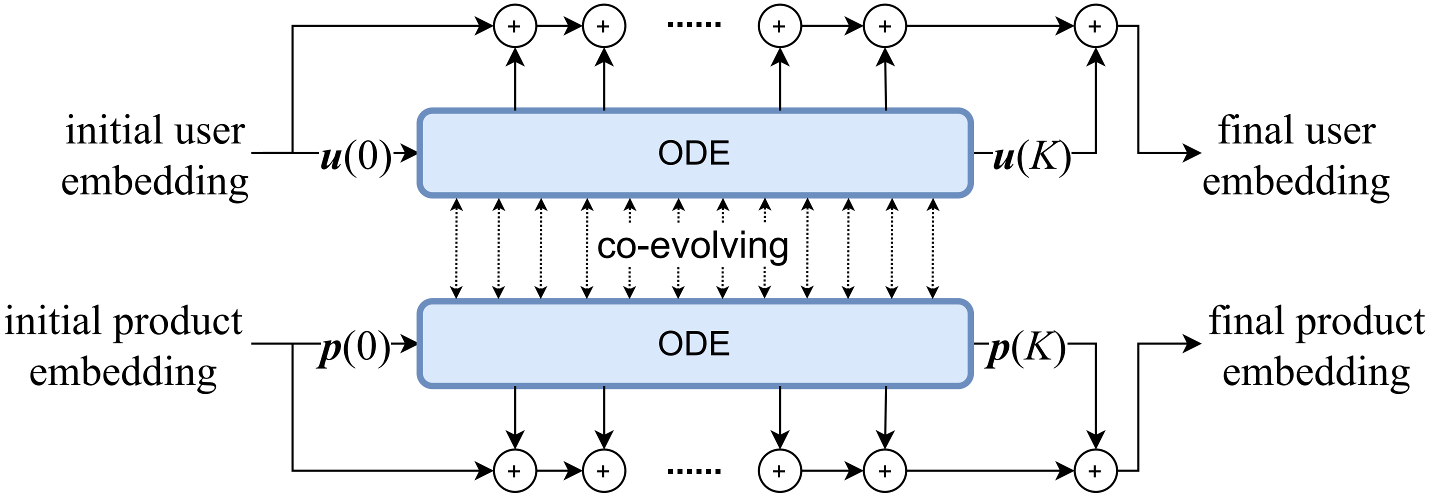


Our proposed LT-OCF

ODE-based User and Product Embeddings

- The user and product embedding **co-evolutionary** process.

$$\boldsymbol{u}(K) = \boldsymbol{u}(0) + \int_0^K f\big(\boldsymbol{p}(t)\big)dt, \quad \boldsymbol{p}(K) = \boldsymbol{p}(0) + \int_0^K g\big(\boldsymbol{u}(t)\big)dt,$$



Our proposed LT-OCF

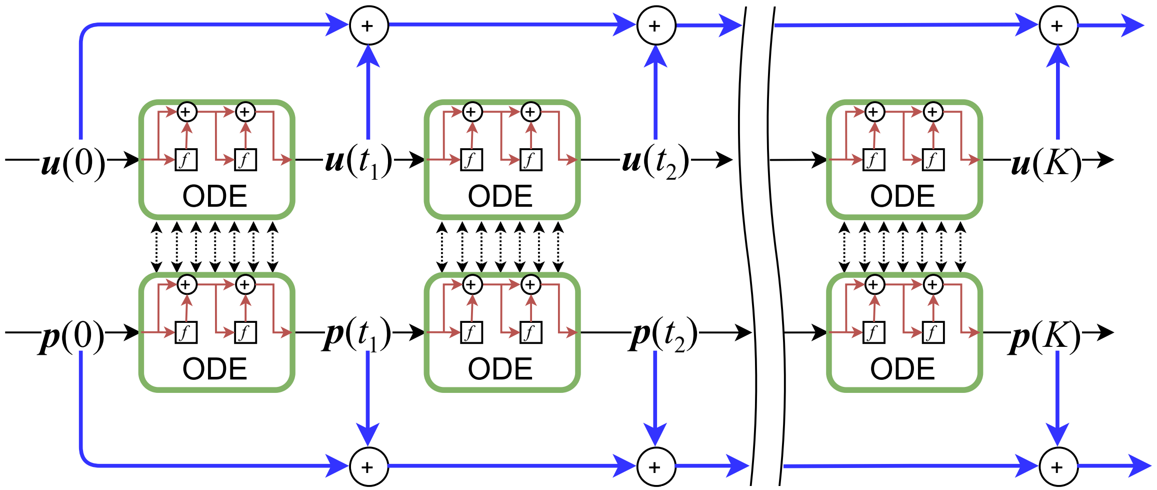
Learnable-time Architecture (1/2)

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

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

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

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

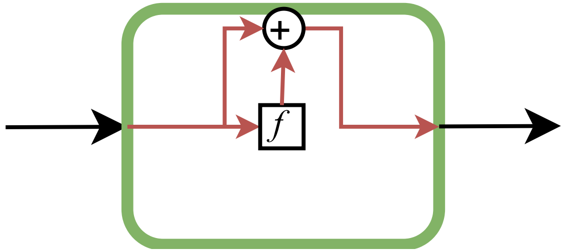


Our proposed dual co-evolving ODEs

Residual/Dense Connections and ODE Solvers

- Explicit Euler
  - is identical to a residual connection when $s = 1$.

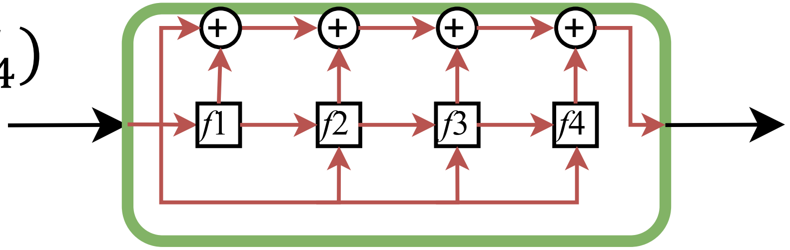
$$\boldsymbol{h}(t+s) = \boldsymbol{h}(t) + s \cdot f(\boldsymbol{h}(t), t; \boldsymbol{\theta}_f)$$



The explicit Euler method in a step

- The fourth-order Runge-Kutta (RK4)
  - is similar to DenseNets and FractalNet.

$$\boldsymbol{h}(t+s) = \boldsymbol{h}(t) + \frac{s}{6}(f_1 + 2f_2 + 2f_3 + f_4)$$



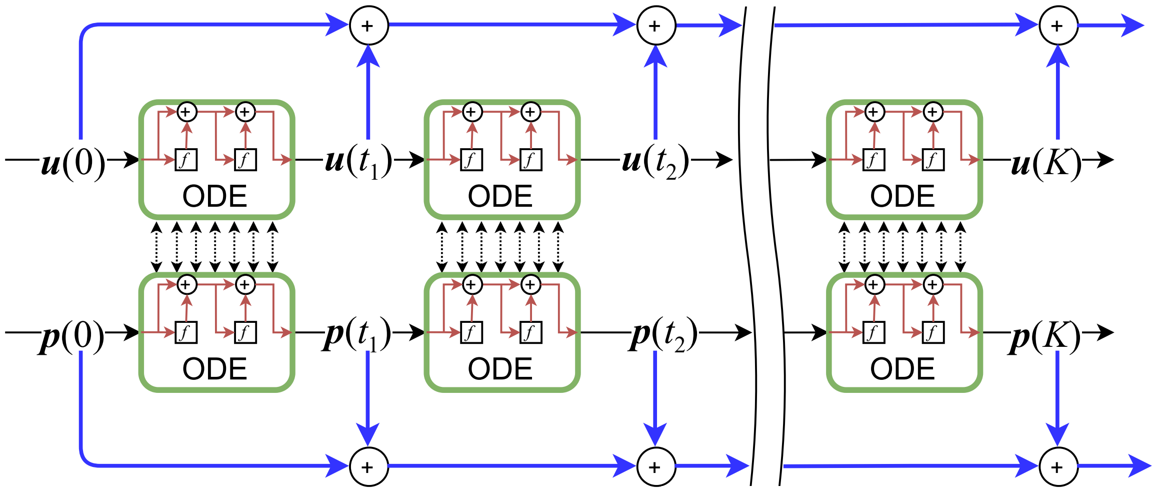
The RK4 method in a step

Learnable-time Arhictecture (2/2)

- The final embeddings are calculated as follows:

$$E^{u}_{final} = w_0 \boldsymbol{u}(0) + \sum_{i=1}^{T} w_i \boldsymbol{u}(t_i) + w_K \boldsymbol{u}(K),$$

$$E^{p}_{final} = w_0 \boldsymbol{p}(0) + \sum_{i=1}^{T} w_i \boldsymbol{p}(t_i) + w_K \boldsymbol{p}(K).$$



Our proposed dual co-evolving ODEs

Relation with Linear GCN-based CF Methods

- Suppose the following setting in our method:
  - **fixed** t,
  - **explicit Euler** method with its step size parameter $s = 1$,
  - and do not use the residual connection but the **linear connection**.

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

$$\dots$$

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

$$\boldsymbol{u}(1) = f\big(\boldsymbol{p}(0)\big), \qquad \boldsymbol{p}(1) = g\big(\boldsymbol{p}(0)\big),$$

$$\dots$$

$$\boldsymbol{u}(K) = f\big(\boldsymbol{p}(K-1)\big), \qquad \boldsymbol{p}(K) = g\big(\boldsymbol{p}(K-1)\big).$$

LT-OCF is a ***continuous*** generalization of linear GCNs, including LightGCN.

Experimental Environments

- Datasets

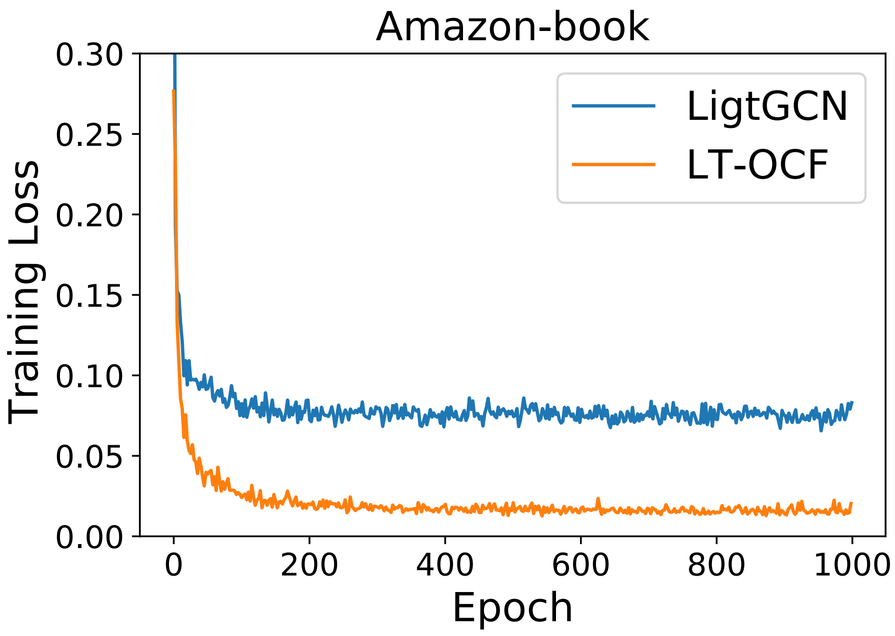
Name	#Users	#Items	#Interactions
Gowalla	29,858	40,981	1,027,370
Yelp2018	31,668	38,048	1,561,406
Amazon-Book	52,643	91,599	2,984,108

Statistics of datasets

- Baselines
  - We consider the 9 baselines to compare with.
- Metrics
  - Recall@20 and NDCG@20, with the all-ranking protocol,
    - i.e., all items that do not have any interactions with a user are recommendation candidates.

Experimental Results (1/2)

- All best results are achieved by RK4.



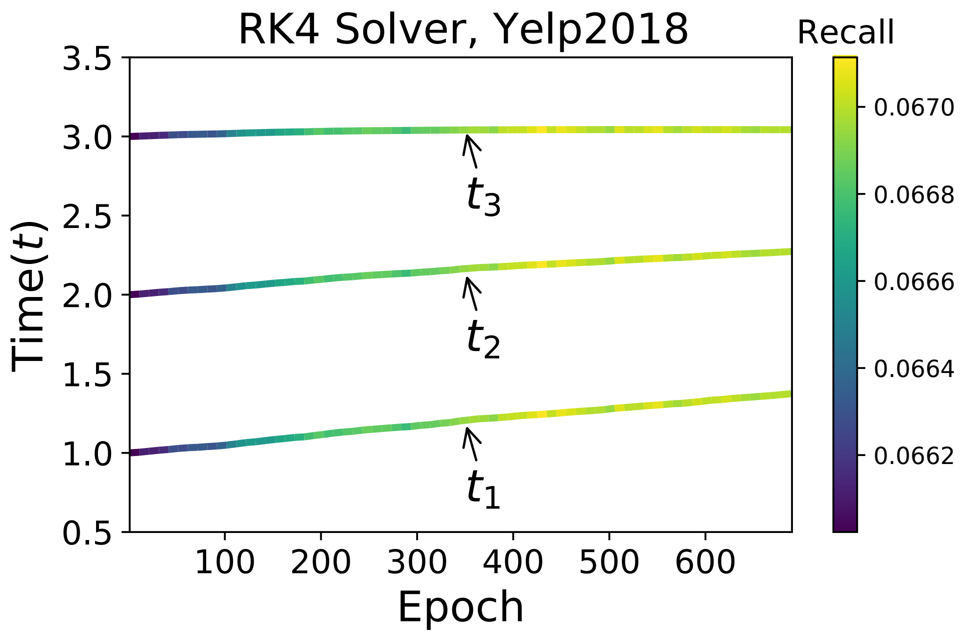
Training curve of loss in Amazon-book

Dataset	Gowalla		Yelp2018		Amazon-Book	
Method	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**

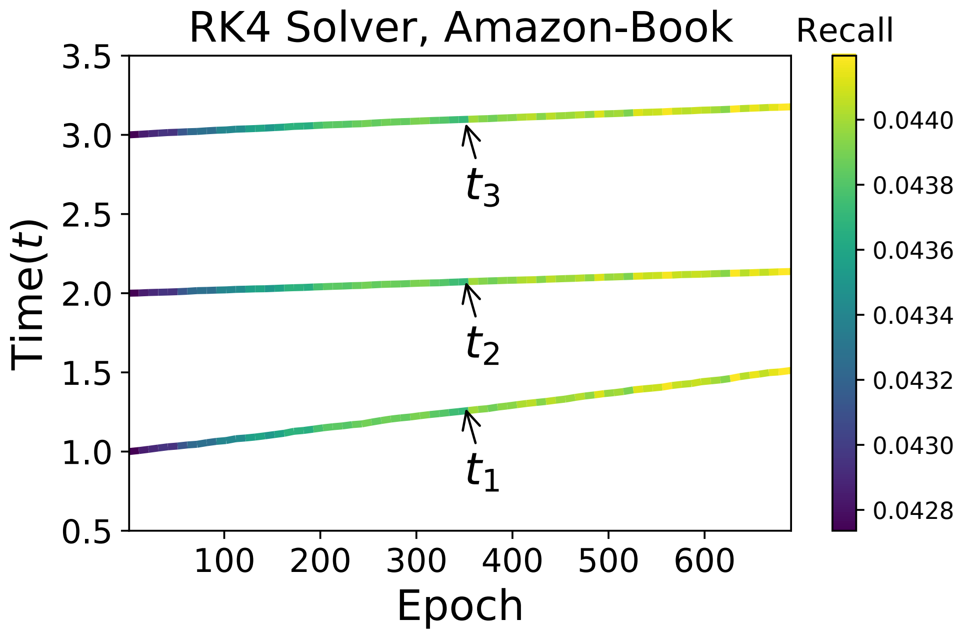
Model performance comparison

Experimental Results (2/2)

- Trianing curves of t_1, t_2, t_3 when $T = 3$.
- Reliable early layers for the layer combination.



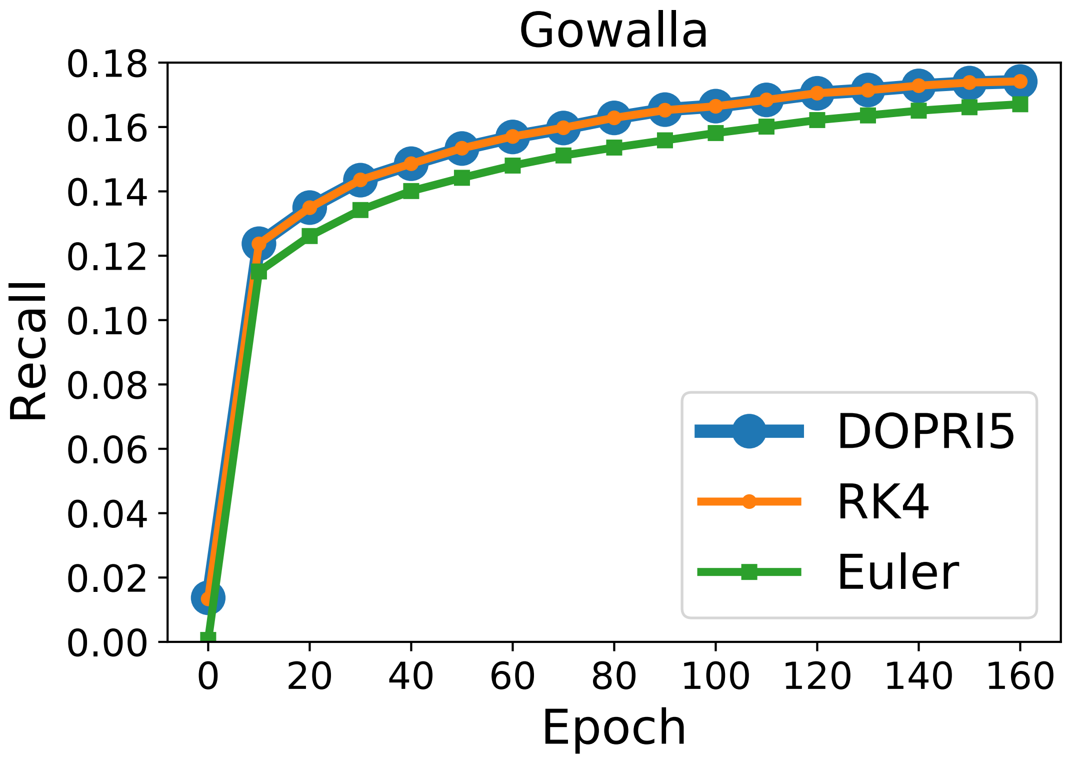
Training curve of t_i in Yelp2018



Training curve of t_i in Amazon-book

Ablation Studies

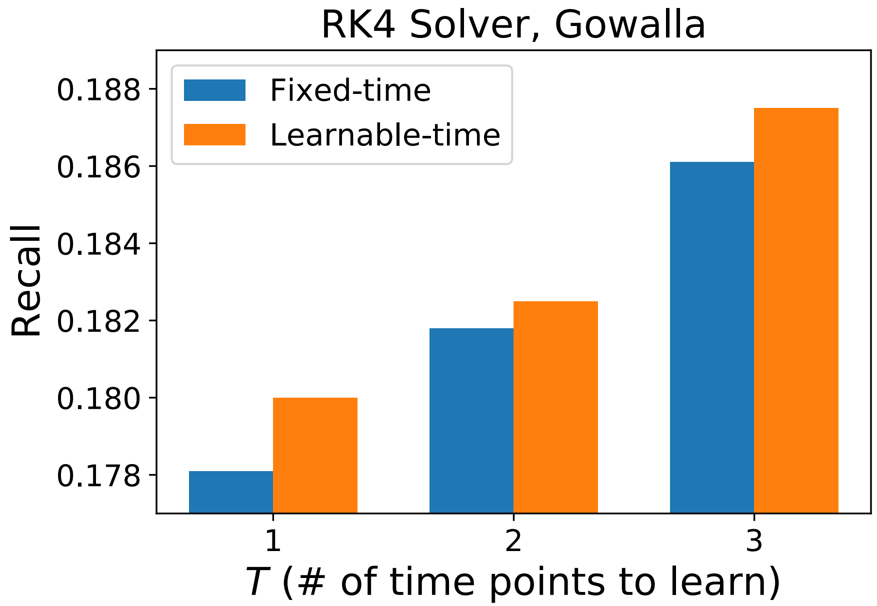
- Euler vs. RK4 vs. DOPRI



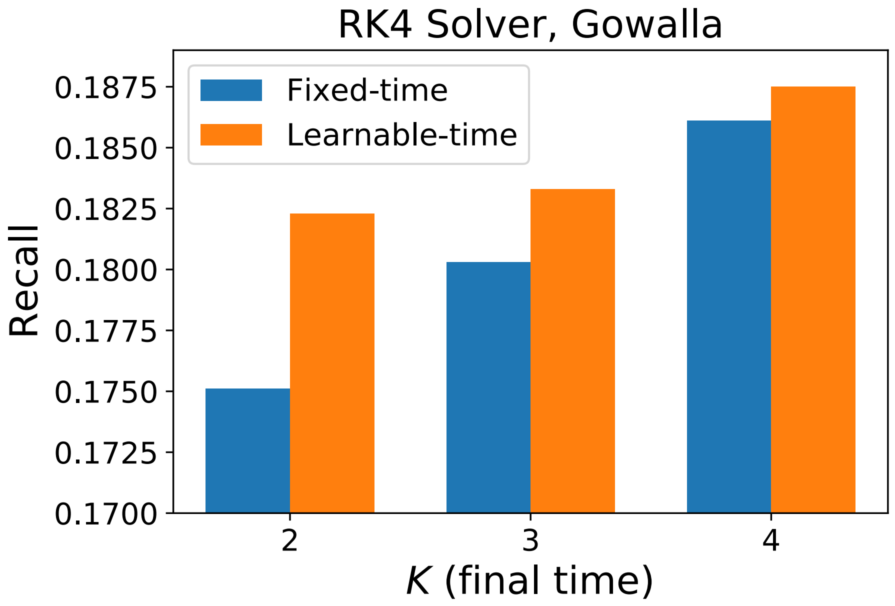
Various ODE solvers

Sensitivity Studies

- Sensitivity on T
  - # of time points to learn
- Sensitivity on Final time K



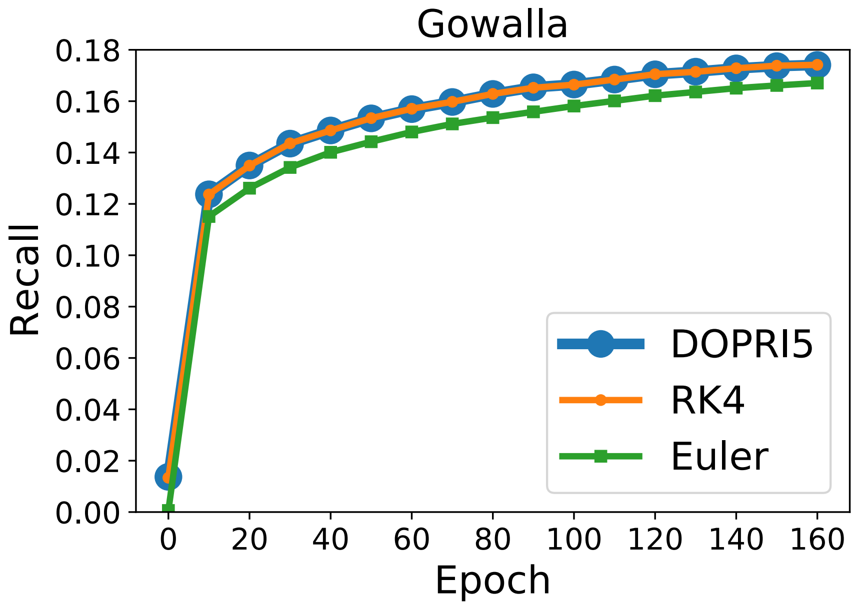
Performance comparison by varying T



Performance comparison by varying K

Discussion on Linear vs. Dense

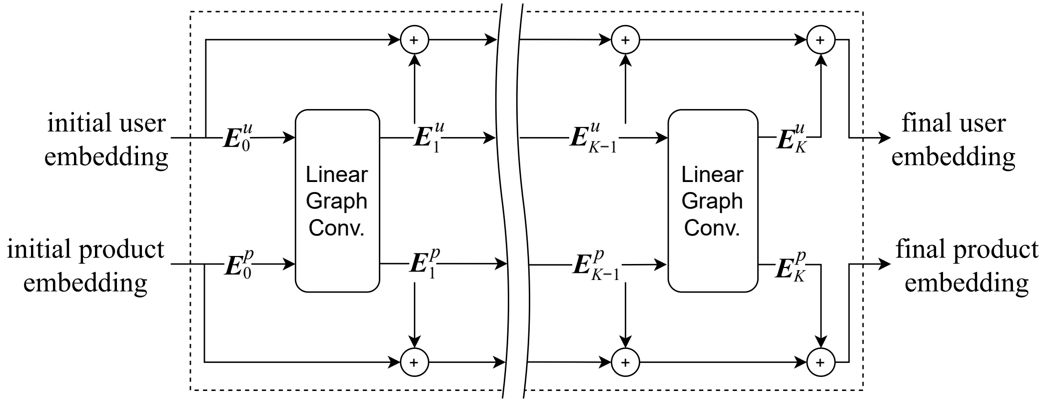
- LightGCN:
  - linear layers with a layer combination.
- **Dense layers** with a *layer combination* are better than linear layers.
- An open question:
  - *Dense connections are also optimal for non-ODE-based CF methods?*



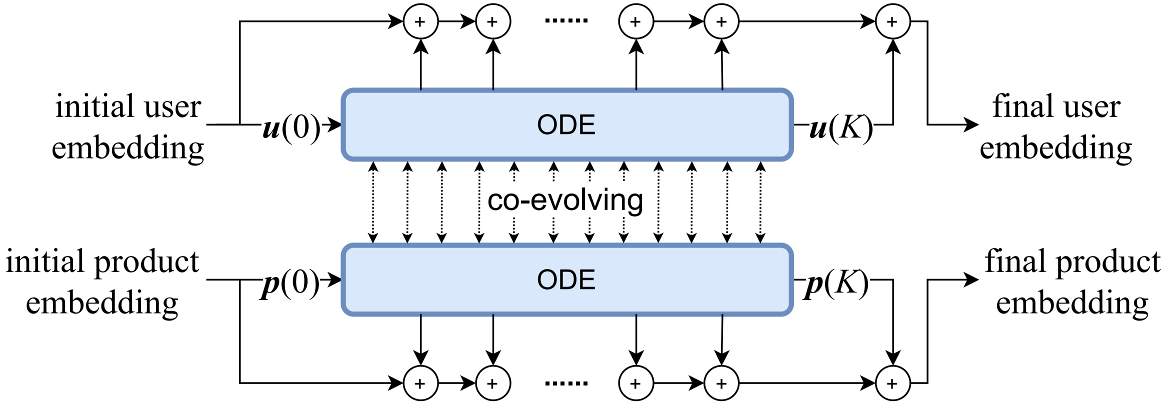
Various ODE solvers

Conclusion

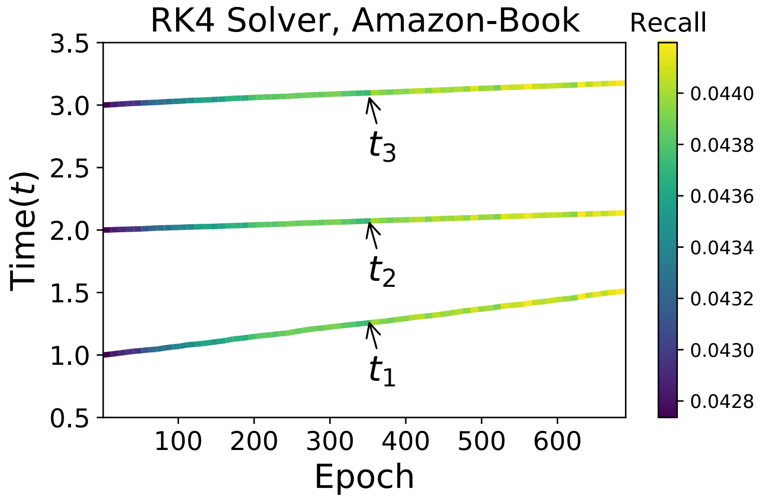
Redesign linear GCNs



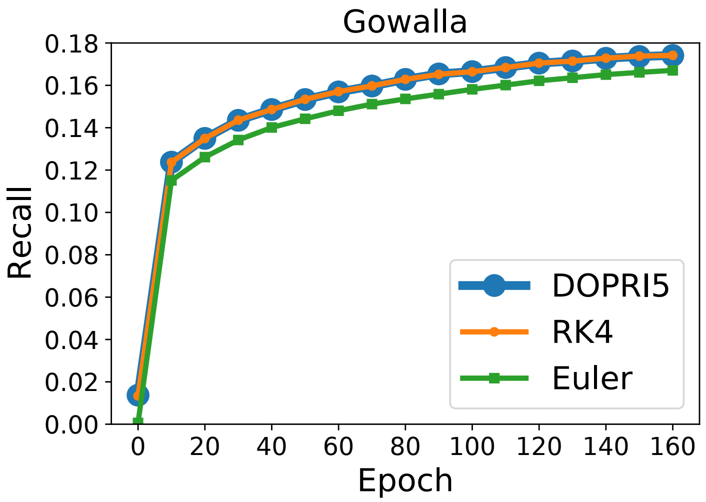
LT-OCF: Learnable-time ODE-based CF



Learnable-Time Architecture



Linear vs. Dense Connection



Further Work

Continuous-time NN

Neural ODE *(NIPS'18)*

ANODE *(NIPS'19)*

⋮

GCN for Recommender Systems

GCMC *(KDD'18)*

NGCF *(SIGIR'19)*

LightGCN *(SIGIR'20)*

and next?

LT-OCF: Learnable-Time ODE-based Collaborative Filtering

Jeongwhan Choi, Jinsung Jeon, Noseong Park
{jeongwhan.choi,jjsjjs0902,noseong}@yonsei.ac.kr
Yonsei University
Seoul, South Korea

Further Work

Continuous-time NN

Neural ODE *(NIPS'18)*

ANODE *(NIPS'19)*

⋮

GCN for Recommender Systems

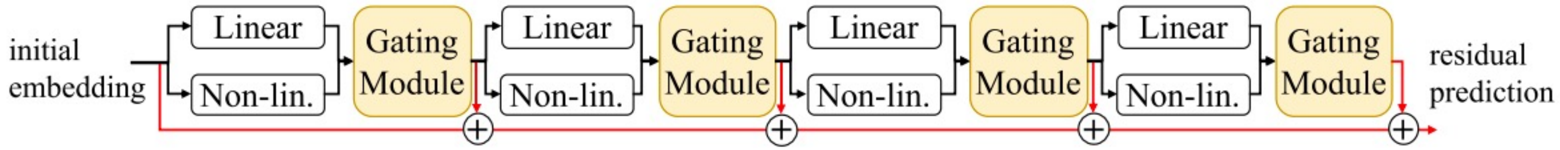
GCMC *(KDD'18)*

NGCF *(SIGIR'19)*

LightGCN *(SIGIR'20)*

LT-OCF *(CIKM'21)*

Linear, or Non-Linear, That is the Question! *(WSDM'22)*



(a) HMLET(All)

Further Work

Continuous-time NN

Neural ODE *(NIPS'18)*

ANODE *(NIPS'19)*

⋮

GCN for Rec. Systems

GCMC *(KDD'18)*

NGCF *(SIGIR'19)*

LightGCN *(SIGIR'20)*

LT-OCF *(CIKM'21)*

Neural ODE + GNN

GNODE *(AAAI workshop DLGMA'20)*

Continuous GNN *(ICML'20)*

GRAND *(ICML'21)*

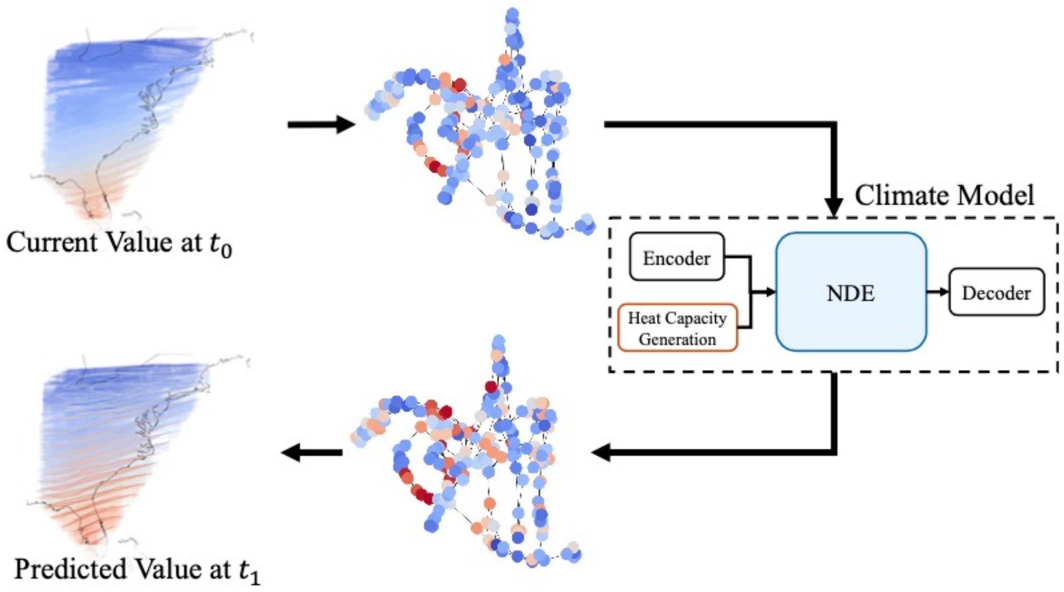
➡ *NODE + Diffusion Eq.+* ***Climate Modeling***

Further Work

Climate Modeling with Neural Diffusion Equations

Jeehyun Hwang[1], Jeongwhan Choi[1], Hwangyong Choi[1], Kookjin Lee[2], Dongeun Lee[3], Noseong Park[1]
Yonsei University[1], Seoul, South Korea
Arizona State University[2], Tempe, AZ, USA
Texas A&M University–Commerce[3], Commerce, TX, USA

Accepted in *ICDM'21*



The overall workflow of our proposed neural diffusion equation(NDE)

Demo Link

github.com/jeongwhanchoi/LT-OCF-Tutorial

THANK YOU

Github Repository: github.com/jeongwhanchoi/LT-OCF

Email: jeongwhan.choi@yonsei.ac.kr