Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation

SIGIR2020

PeterRec Data&Code: https://github.com/fajieyuan/sigir2020_peterrec

Fajie Yuan（Tencent）; Xiangnan He（University of Science and Technology of China）
Alexandros Karatzoglou (Google Research); Liguang Zhang (Tencent)

Outline

- Motivation
- Related Work
- PeterRec
- Experiments

Users engage with recommender systems and provided or left feedback.

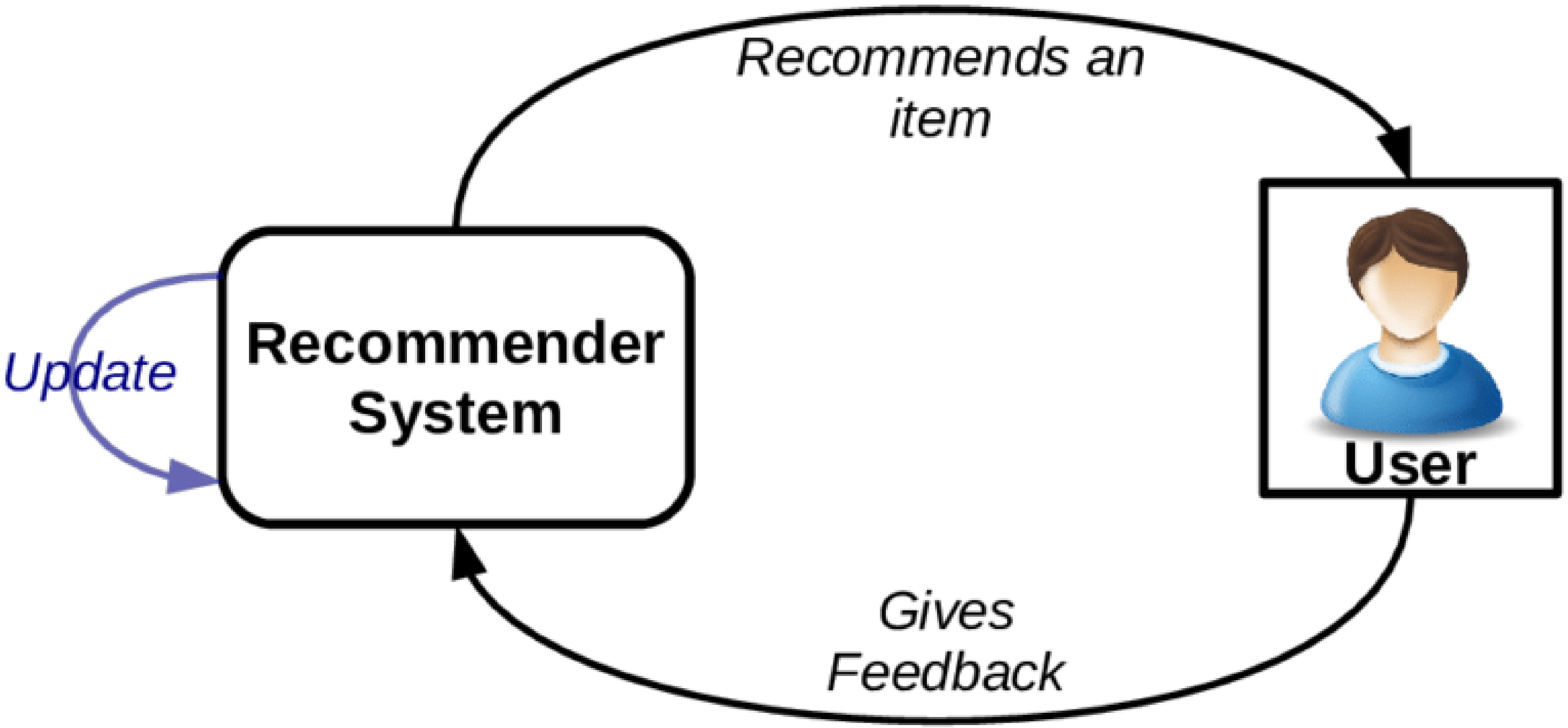
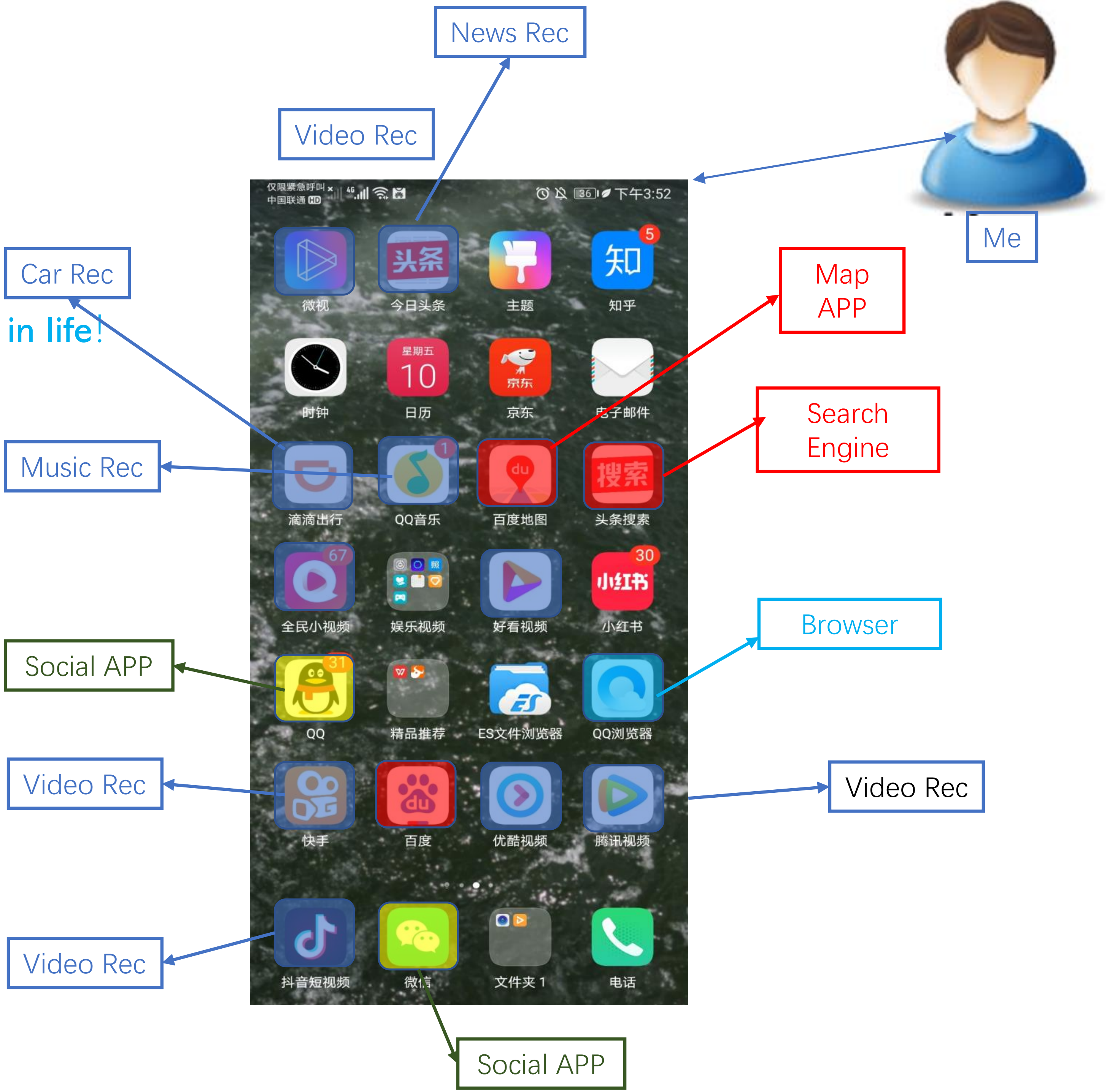


Figure: https://www.researchgate.net/figure/The-sequential-recommendation-process-After-the-RS-recommends-an-item-the-user-gives_fig4_311513879

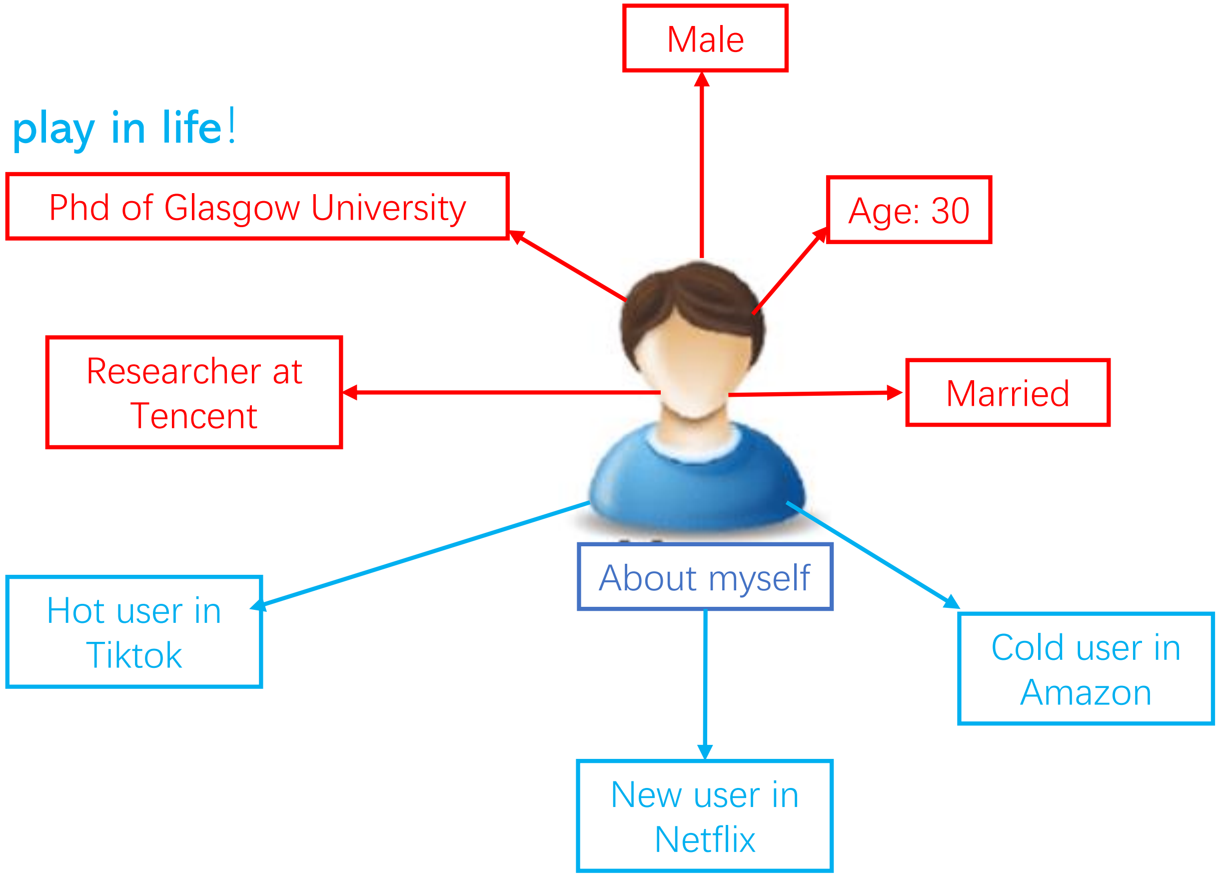
our Motivation

A user has different roles to play in life!



our Motivation

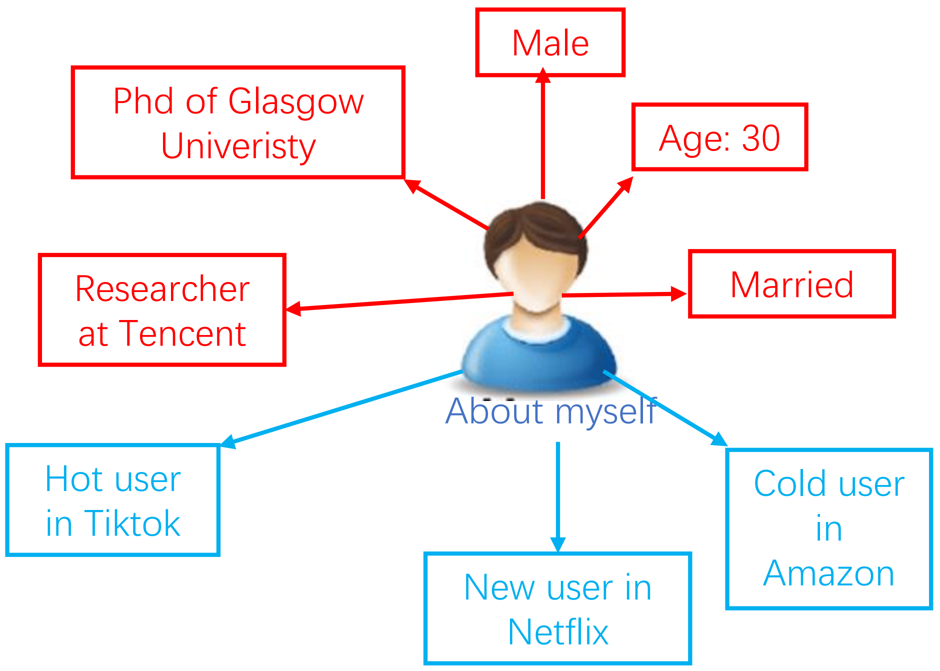
A user has different roles to play in life!



My user Model

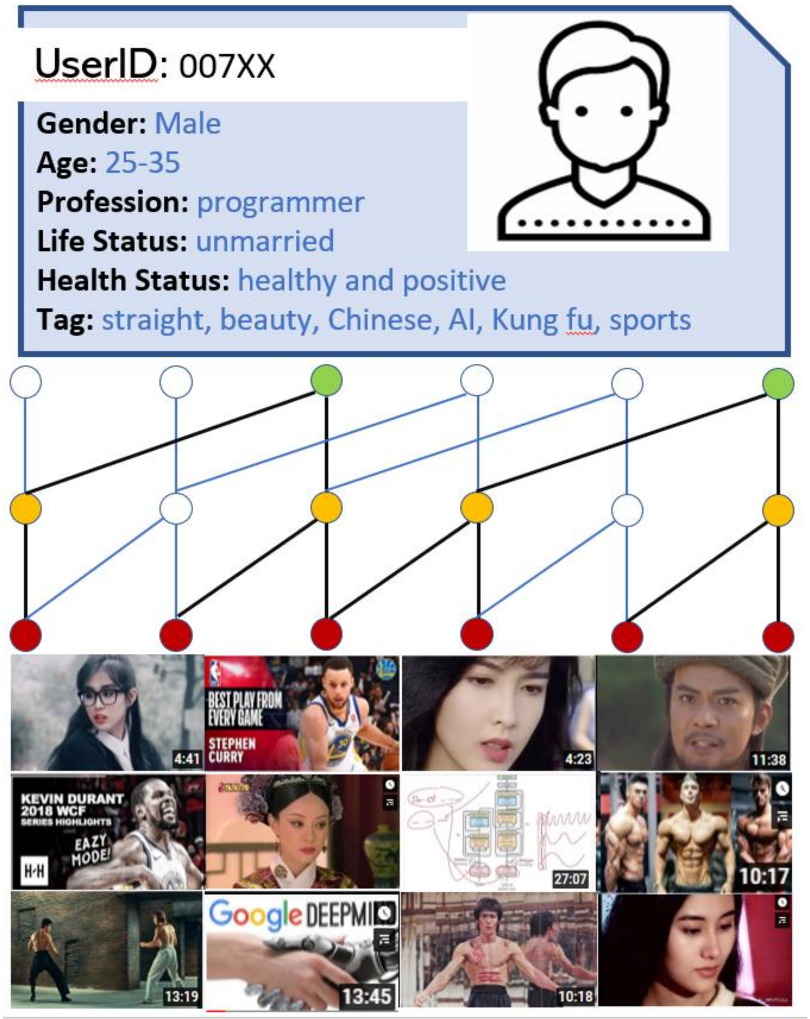
our Motivation

A user has different roles to play in life!



My user Model

Our PeterRec



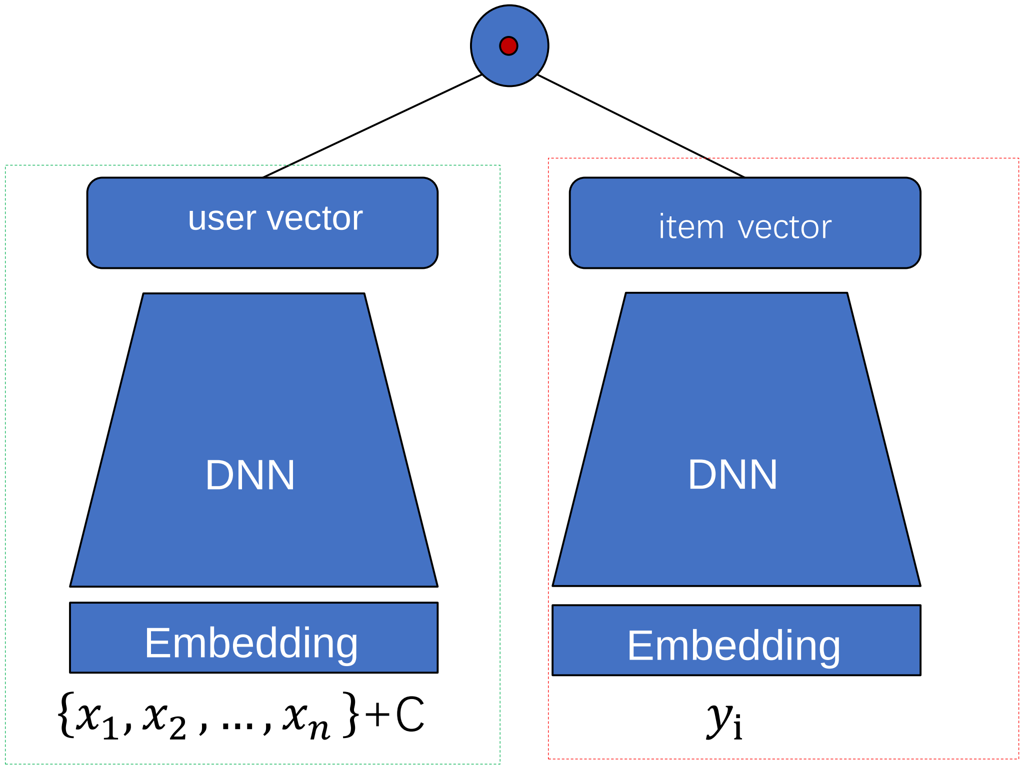
Outline

- Motivation
- Related Work
- PeterRec
- Experiments

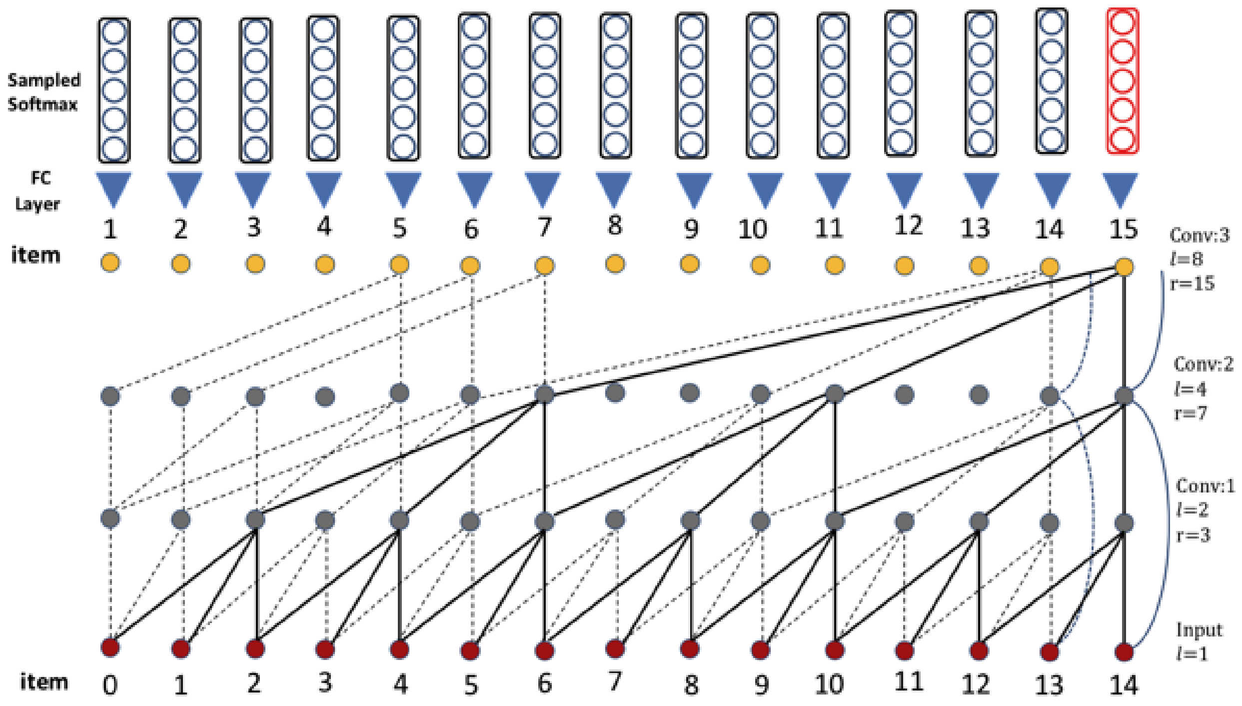
- # Recommendation Background:

(1) Content & Context Recommendation

(2) Session-based Recommendation: recommending the next item based on previously recorded user interactions.



A DSSM (Non-sequential) RS model (Supervised Learning)



Sequential NextItNet (Self-supervised Learning)

Why sequential recommendation?

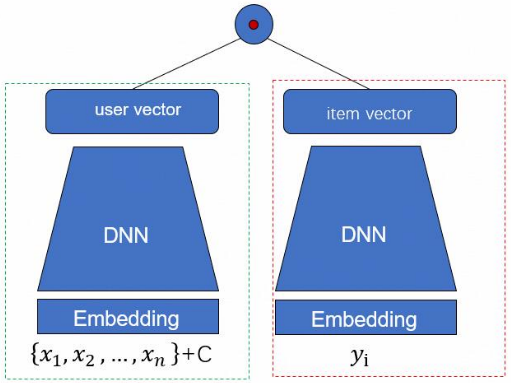
- Short-videos (Tik Tok, Weishi, Kuaishou)

  Music (Tencent music, Yahoo! Music) & News

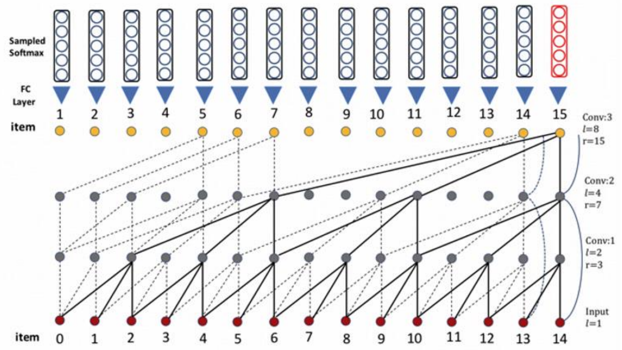
  Movie clips (You Tube, Netflix)

NonSeq Rec vs. Seq Rec:

- Only Static vs. Dynamic Preference
- Manual Feature Engineering vs. Manual-free Features
- Supervised Learning vs. Unsupervised (self-supervised) Learning



A DSSM (Non-sequential) RS model (Supervised Learning)



Sequential NextItNet (Self-supervised Learning)

Transfer Learning Background

TL aims to extract the knowledge from one or more source tasks and applies the knowledge to a target task.

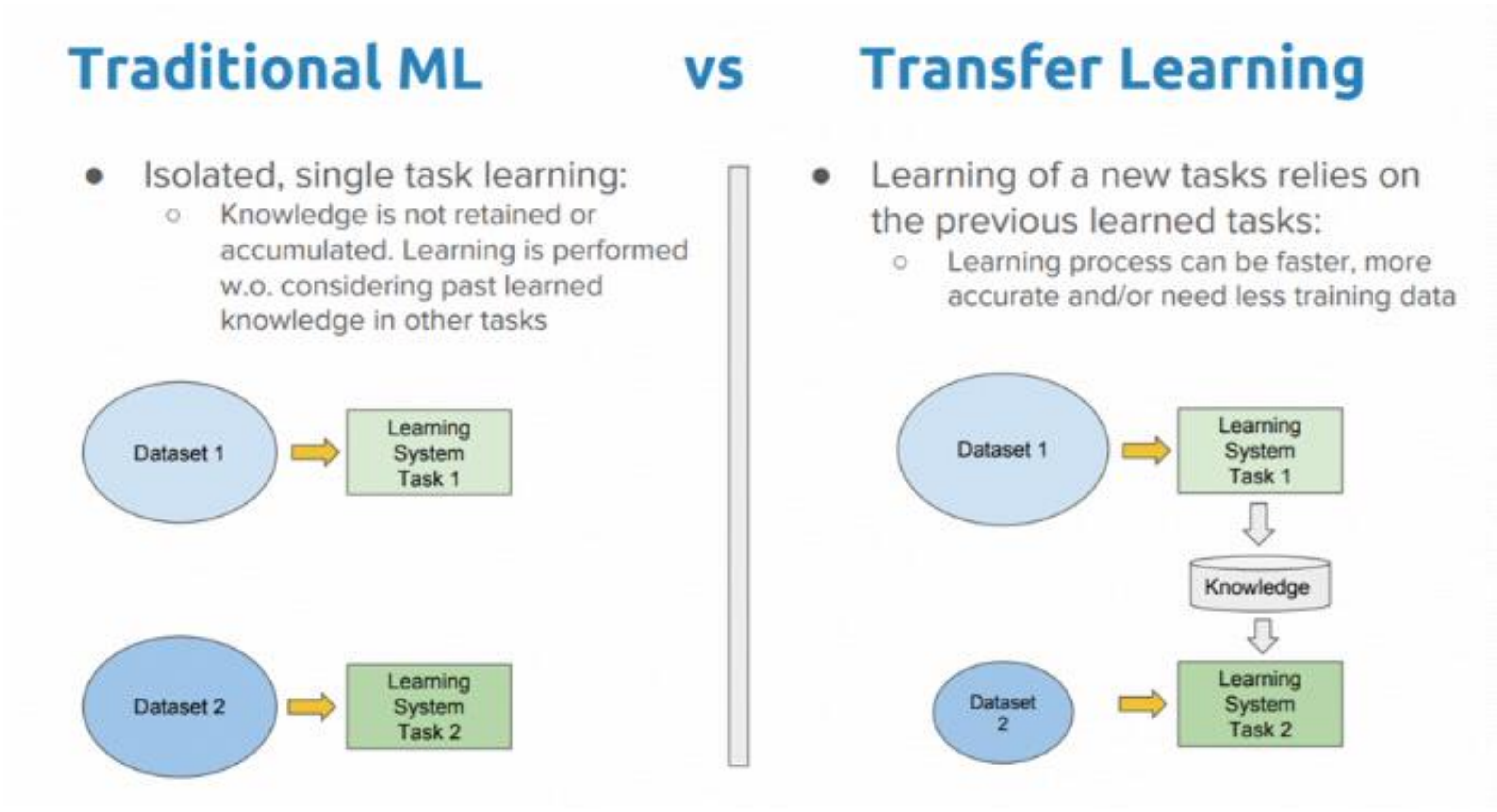
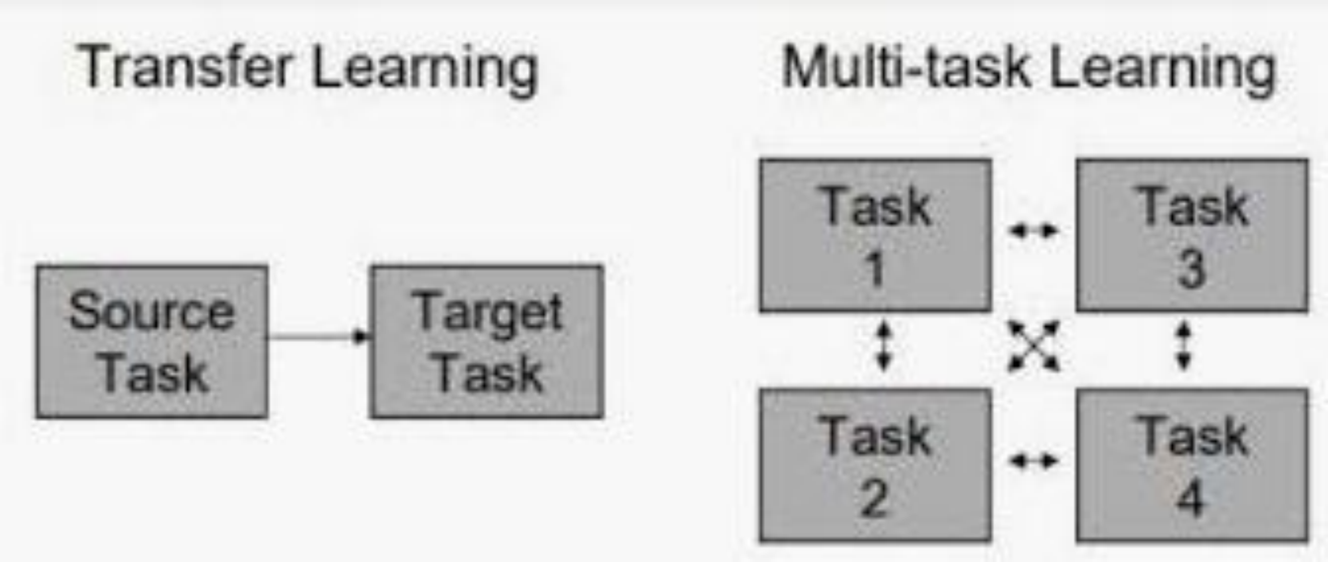


Figure: A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, online

Transfer Learning (TL) vs Multi-task Learning (MTL)

	Training	Testing
Transfer Learning	Task 1	Task 2
Multi-task Learning	Task 1 … Task N	Task 1 … Task N



[1] ICML 2018: Advances in transfer, multitask, and semi-supervised learning, online

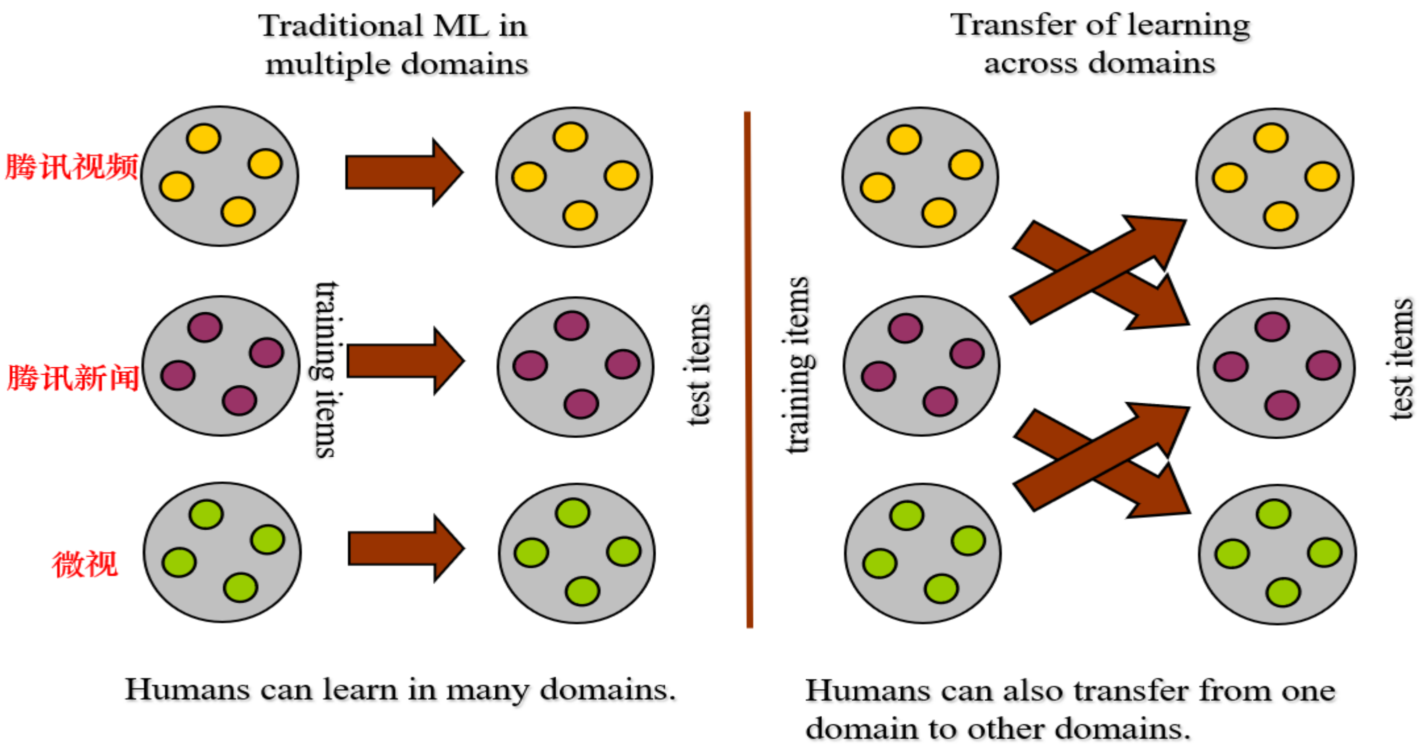
TL vs MTL

- Two-stage training vs joint training
- One objective vs multiple objectives
- Care only target vs. care all objectives

Transfer Learning (TL) for Recommender System (RS)

Motivation:

- User representation may be generic, since their preference tends to be similar across different recommendation task. That is, user's engagement in previous platforms may be important training signals for other systems.
- Traditional ML models usually fail to when modeling new or cold users due to lack of interaction data



[1]

[1] figure is from online, url is missing

Outline

- Motivation
- Related Work
- PeterRec
- Experiments

Transfer Learning (TL) for Recommender System (RS)

Task description

Source data: (u, x^u), $where\ \ x^u = \{x_1^u, x_2^u, \ldots x_n^u\}$,

where x_t^u denotes the t−th interacted item of user u

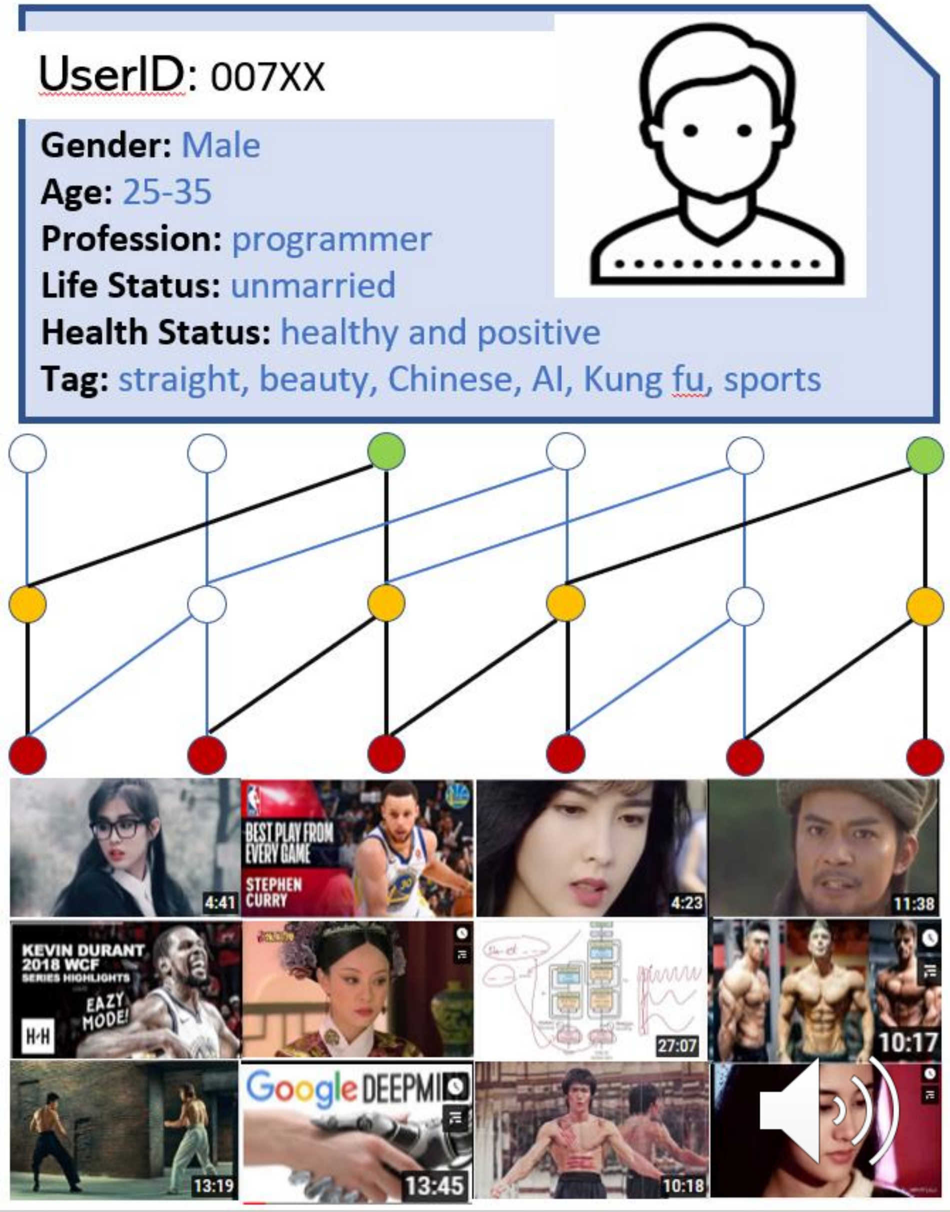
Target data: (u, y) where y is the supervise label in the target dataset

Example

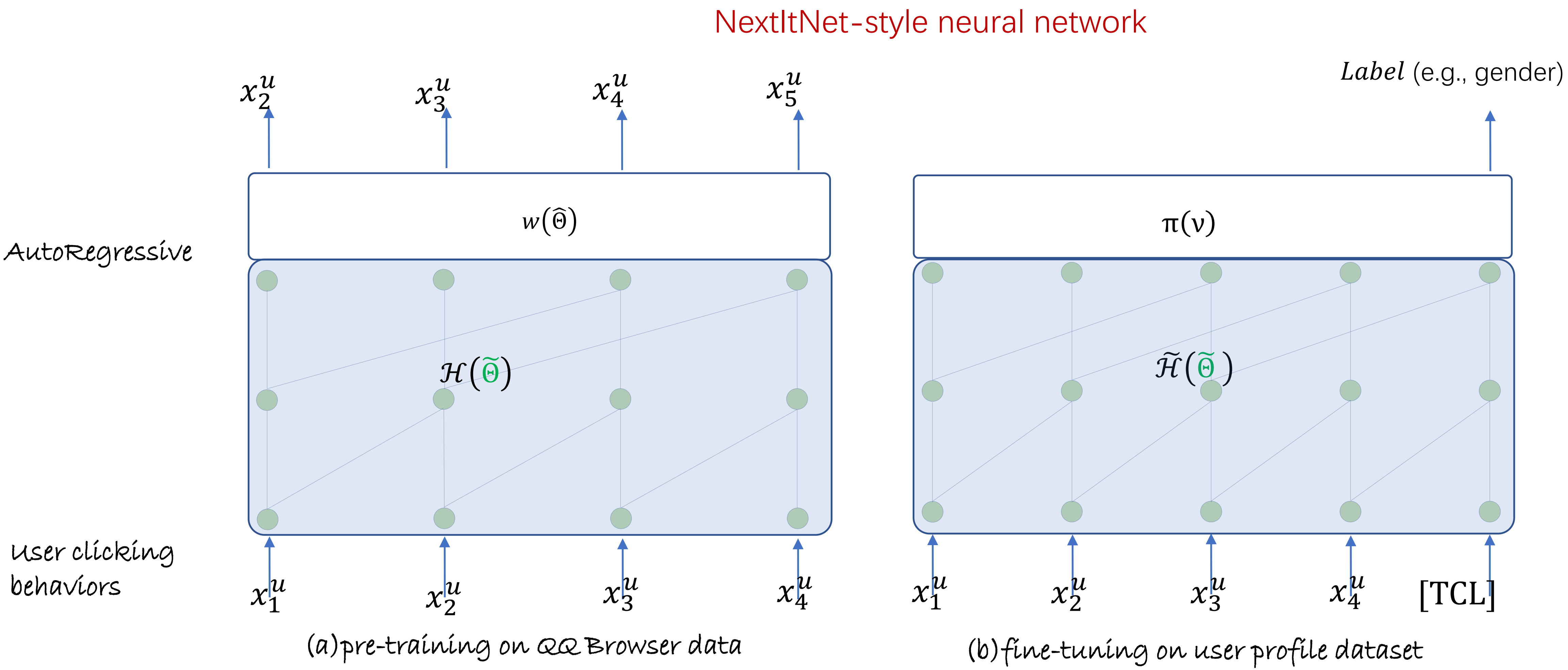
Source data: user's watching activities in Tencent QQ Browser

Target data: user's watching activities in Kandian, but users are cold or new here

or user's profile labels e.g. age, gender, lifestatus, etc.



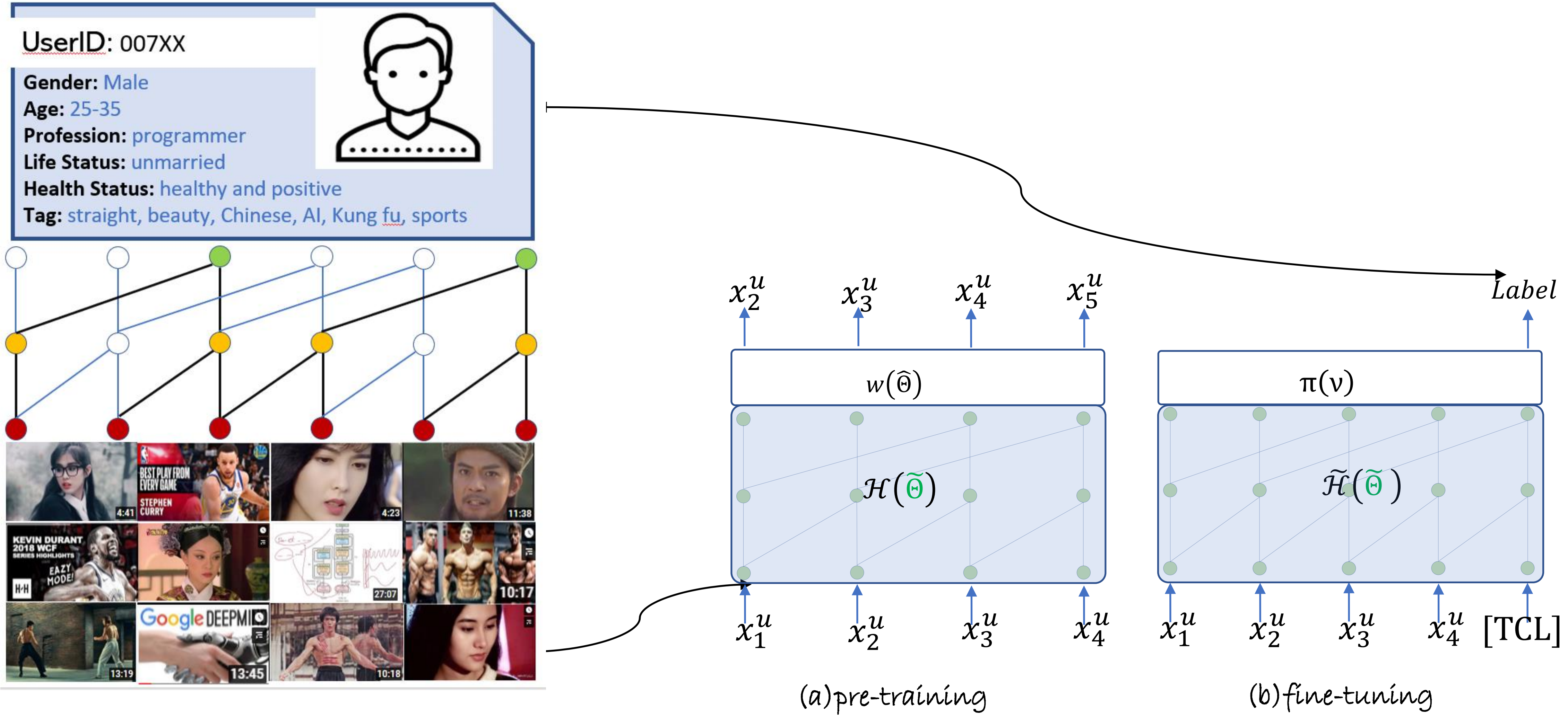
PeterRec Architecture



NextItNet: A Simple Convolutional Generative Network for Next Item Recommendation. WSDM2019, Yuan et al.

What can be done by PeterRec

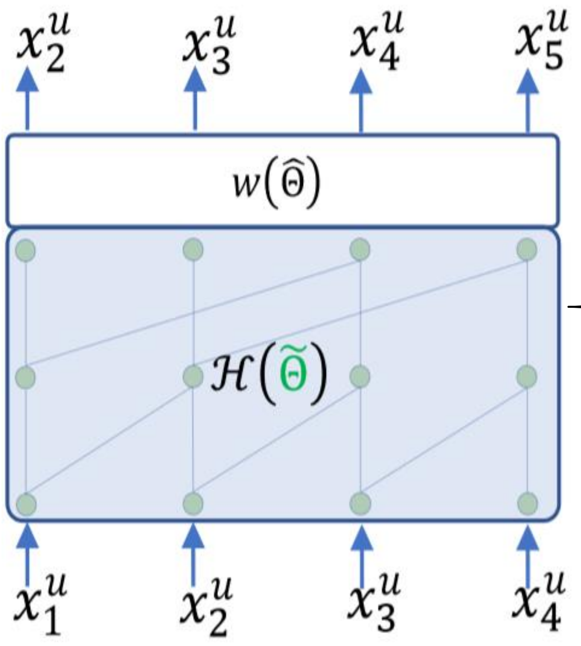
- Cold-start problem, e.g., ads rec
- User profile prediction, e.g., gender prediction



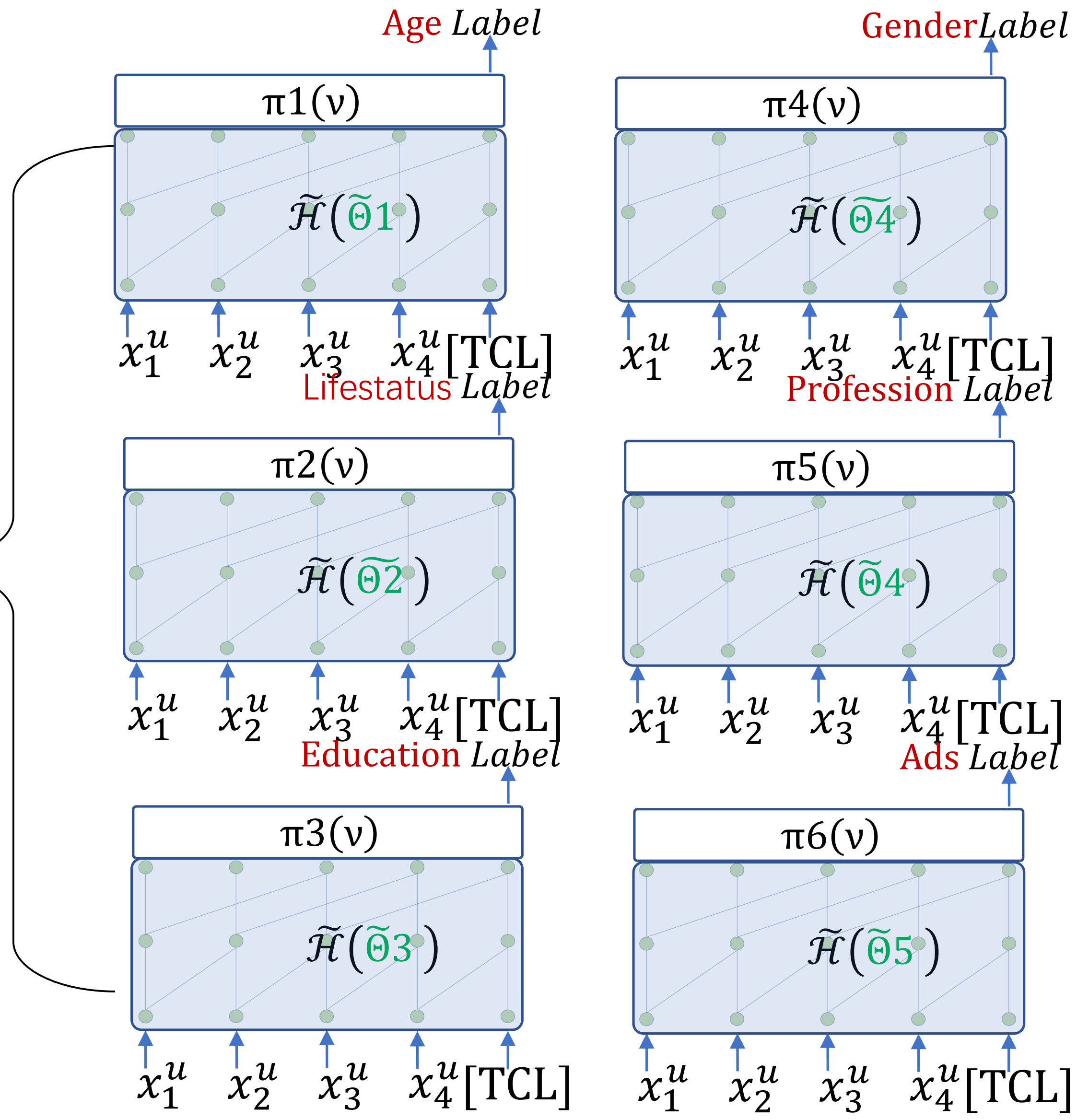
Problems we meet when a number of tasks are required.

Training a separate model for each downstream task is parameter-inefficient since both pretraining & finetuning models are very large.

The number of finetuned models is as many as the number of downstream tasks.
100 tasks=100 finetuned models

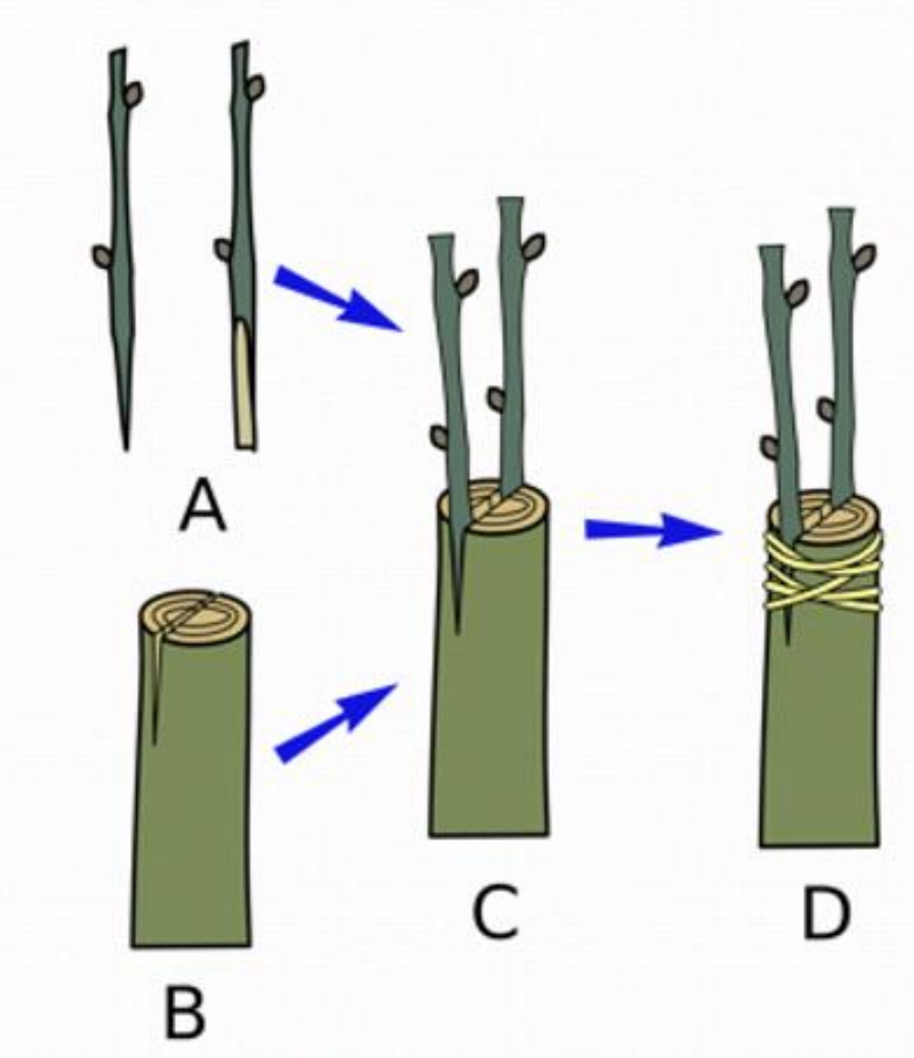


pretraining



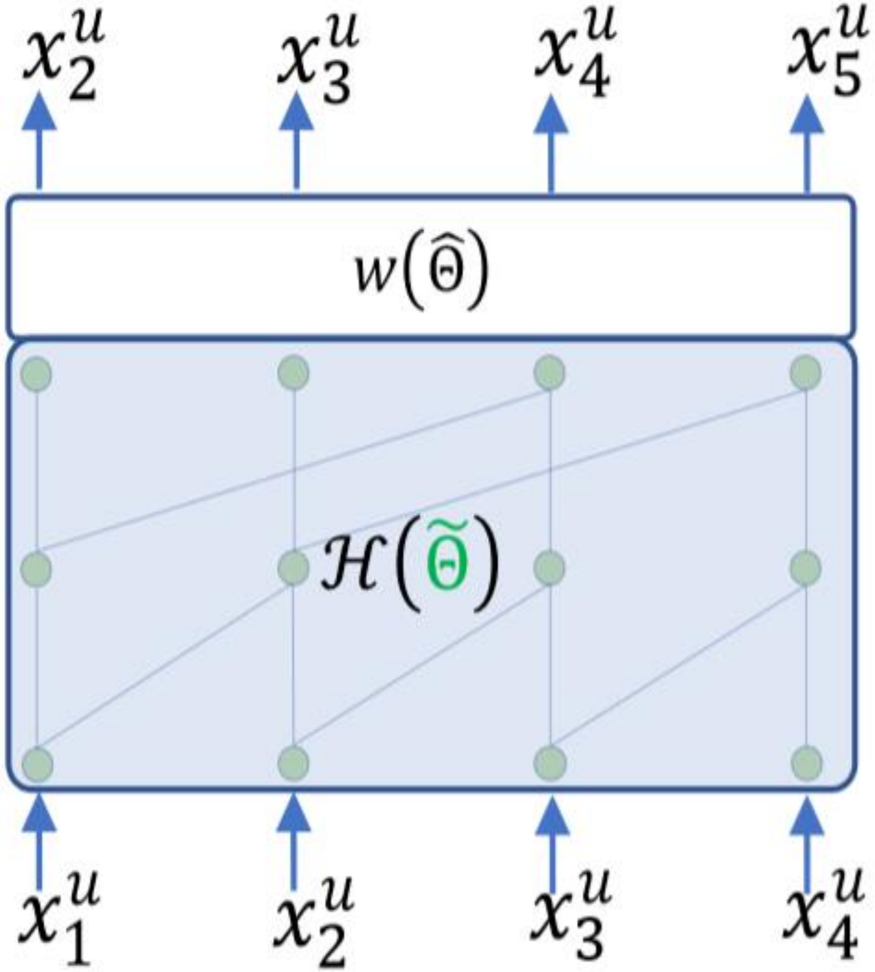
Six fine-tuning models

Taking inspiration from grafting



Tree

A: branch of plum
B: Tree of peach
C: insertion
D: grow together

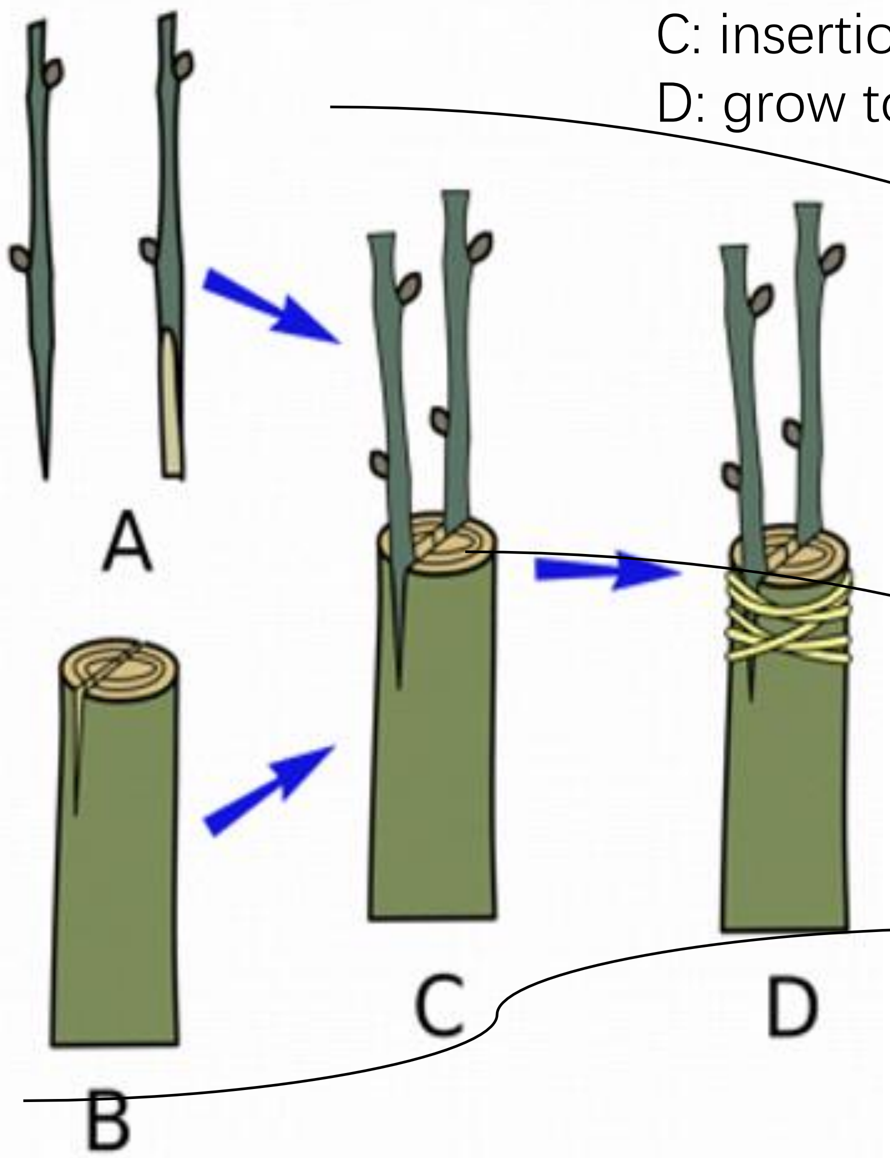


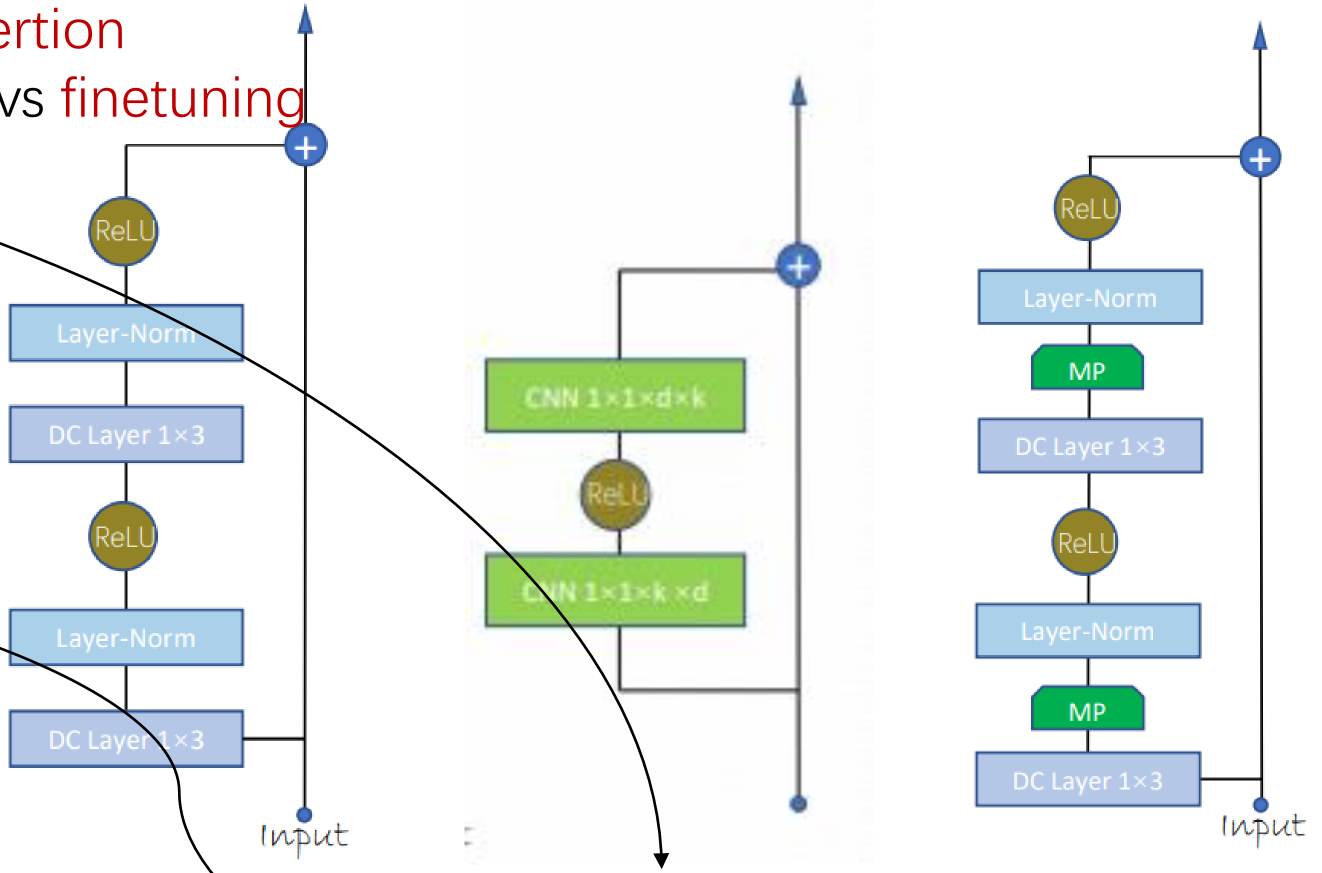
Pretrained model

Pretrained model is treated as the peach Tree.

Grafting for plants.

A: branch of plum vs MP
B: Tree of peach vs pretrained model
C: insertion vs insertion
D: grow together vs finetuning





Original resnet block

MP

insertion

Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation, Yuan et al SIGIR 2020

Outline

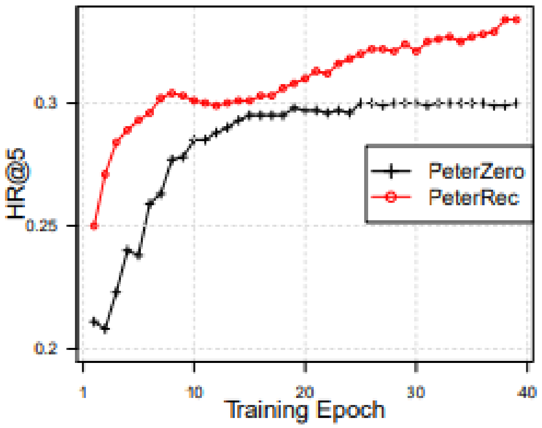
- Motivation
- Related Work
- PeterRec
- Experiments

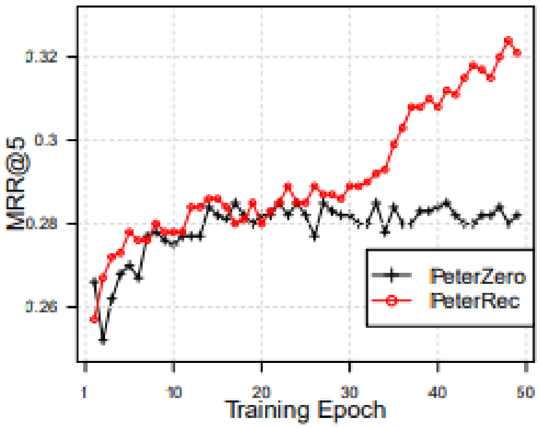
Results.

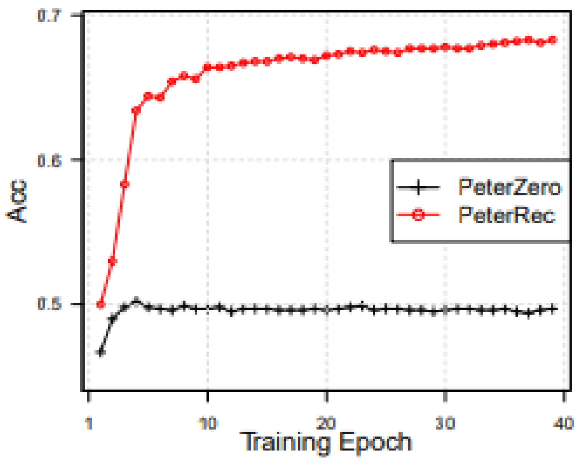
Is pretraining necessary?

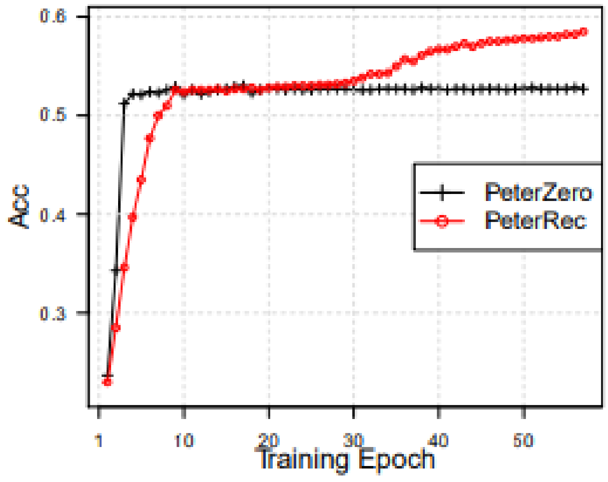
PeterZero: no pretraining

PeterRec: with pretraining









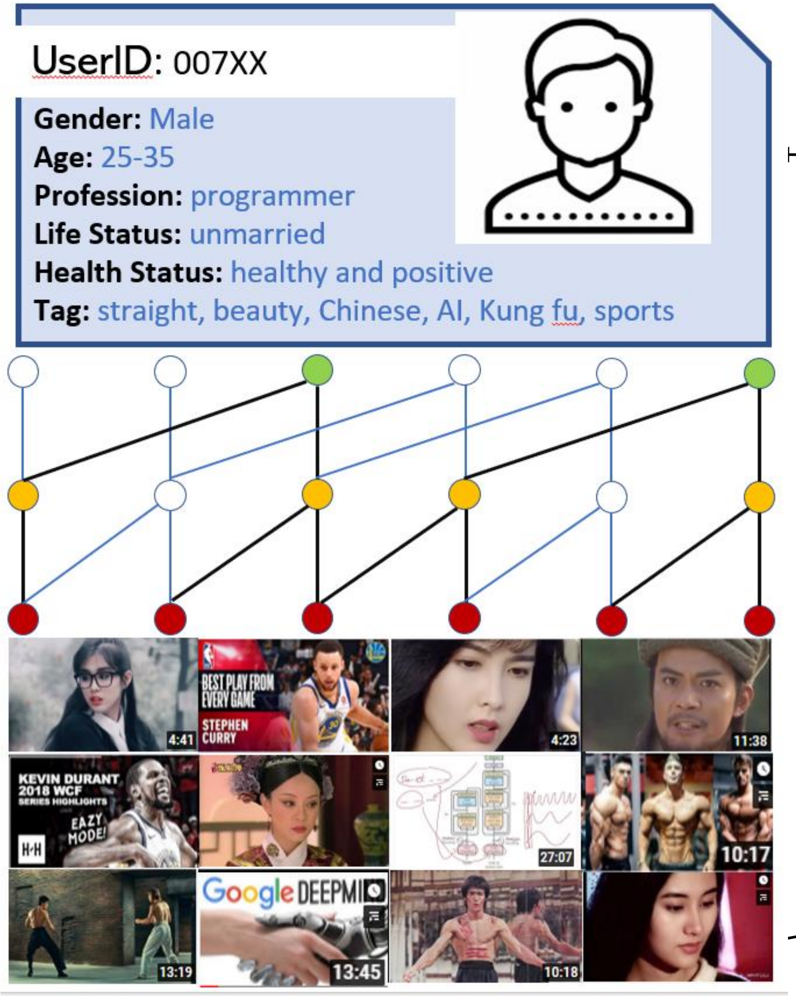
Results.

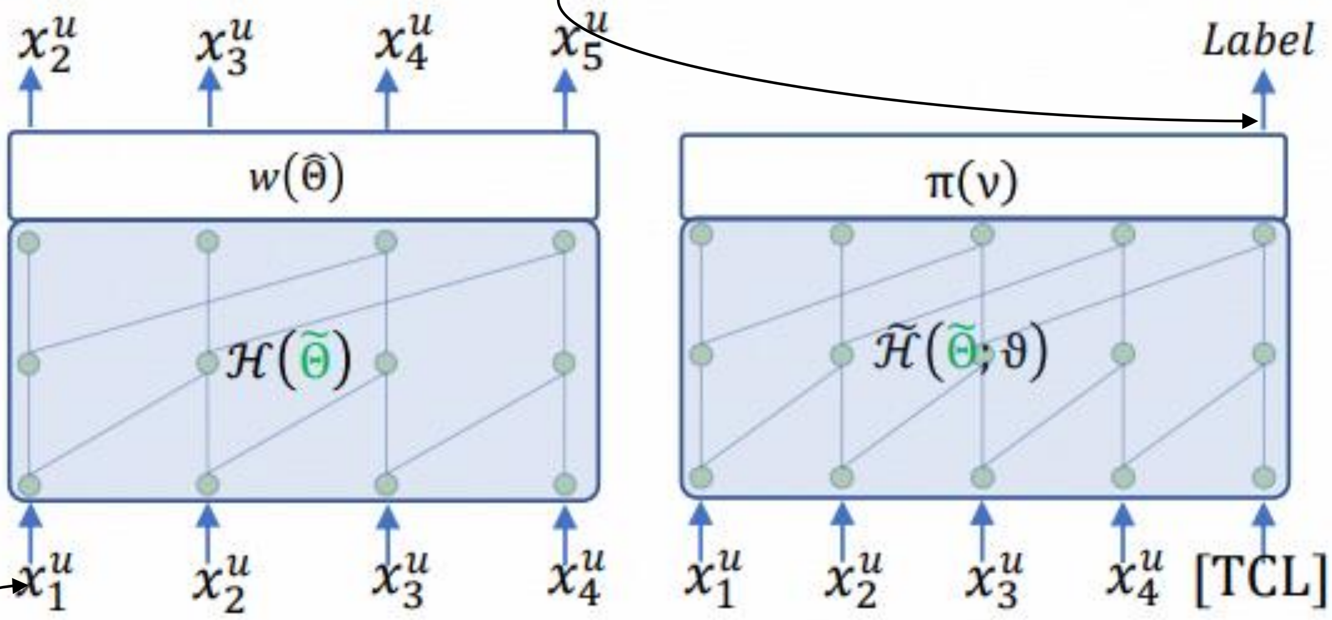
Table 3: Performance comparison (with the non-causal CNN architectures). The number of fine-tuned parameters (ϑ and ν) of PeterRec accounts for 9.4%, 2.7%, 0.16%, 0.16%, 0.16% of FineAll on the five datasets from left to right.

Model	ColdRec-1	ColdRec-2	GenEst	AgeEst	LifeEst
FineCLS	0.295	0.293	0.900	0.679	0.606
FineLast	0.330	0.310	0.902	0.682	0.608
FineAll	0.352	0.338	0.905	0.714	0.615
PeterRec	0.351	0.339	0.906	0.714	0.615

What can be done by Peterrec

Adolescent mental health - for parents
Payment capacity - for bank
Advertising – for company
More





Example : if we have the video watch be- haviors of a teenager, we may know whether he has depression or propensity for violence by PeterRec without resorting to much fea- ture engineering and human-labeled data.


Q&A