Incremental Learning for Multi-Interest Sequential Recommendation

Zhikai Wang (Cloudcatcher.888@sjtu.edu.cn)

Yanyan Shen (shenyy@sjtu.edu.cn)

Shanghai Jiao Tong University

ICDE2023

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

Outline

- Background
- Problem Setting
- The IMSR Framework
- Experiments
- Conclusion

Background

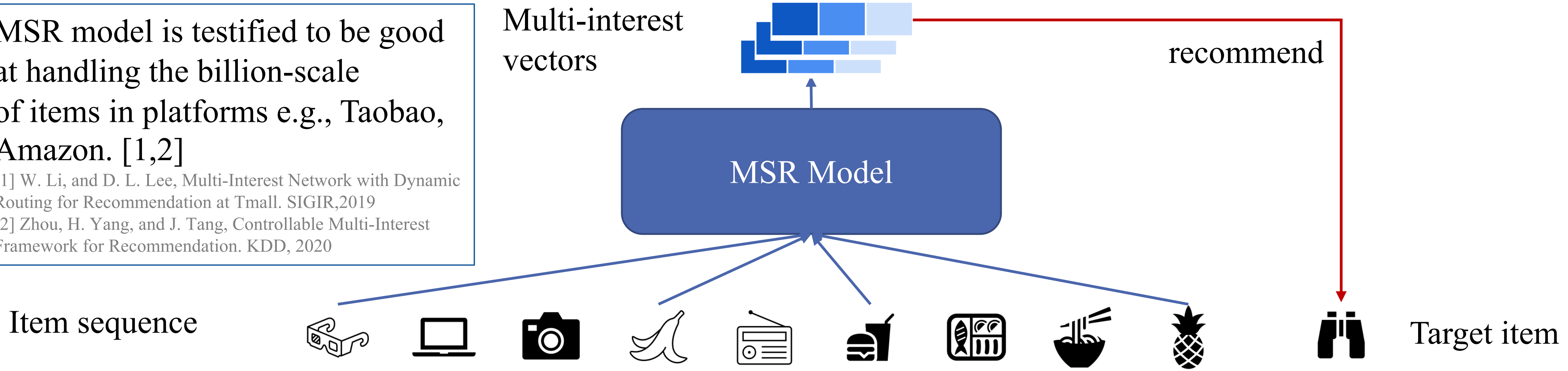
What is Multi-interest Sequential Recommendation?

- Sequential recommendation (SR):
  - Typically extract one user preference vector from the interaction sequence.
- Multi-interest sequential recommendation (MSR):
  - Recent works argue that user usually has multiple interests beneath interaction sequence.

➡ Explicitly compute multiple interest vectors based on user interaction sequence.

MSR model is testified to be good at handling the billion-scale of items in platforms e.g., Taobao, Amazon. [1,2]

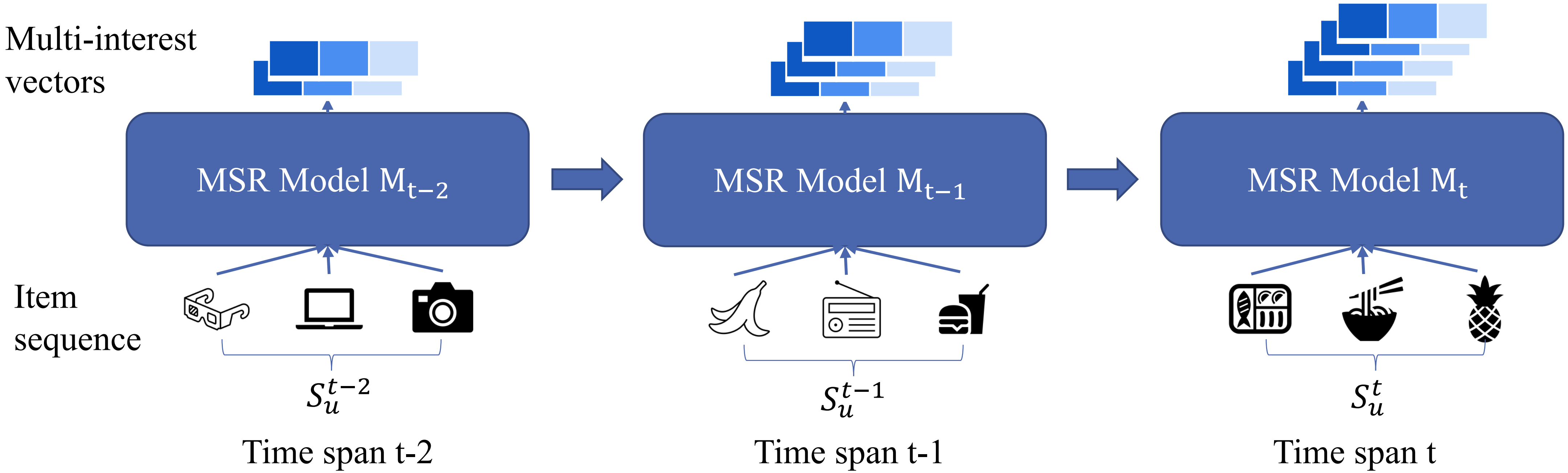
[1] W. Li, and D. L. Lee, Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. SIGIR,2019
[2] Zhou, H. Yang, and J. Tang, Controllable Multi-Interest Framework for Recommendation. KDD, 2020



Do MSR Models Need to be Updated Incrementally? YES

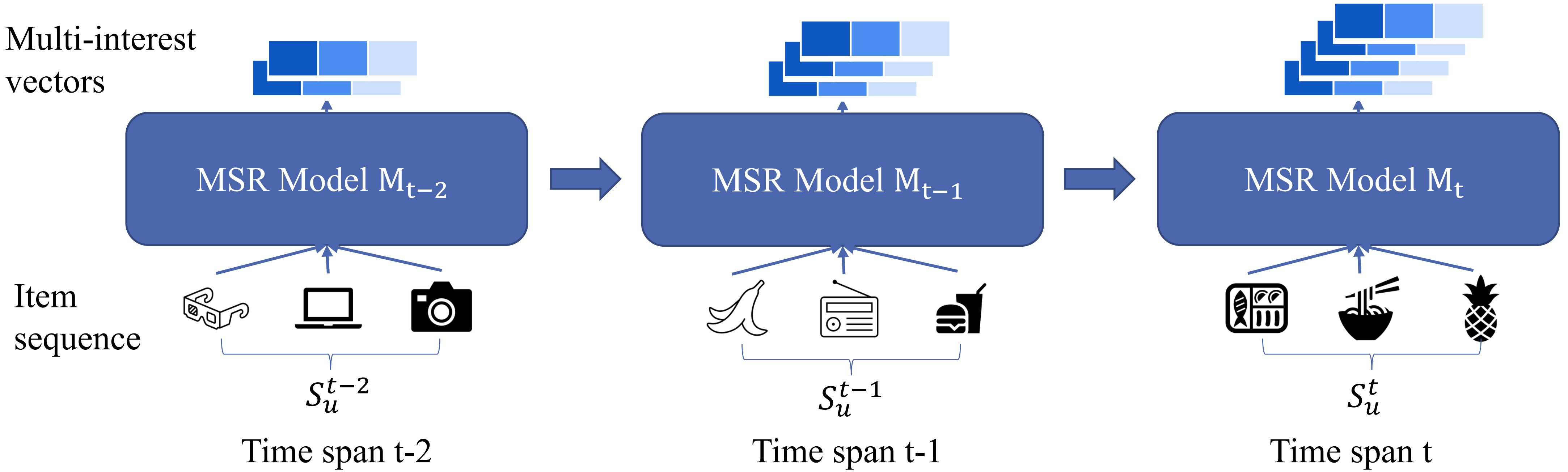
- User interaction sequence $S_u^1, S_u^2, \ldots, S_u^t, \ldots$ is collected continuously.
- Existing interests may drift & new interests may be generated.
- Retraining MSR model M_t on all historical $S_u^1, S_u^2, \ldots S_u^{t-1}$ & new data S_u^t is time-consuming.

➡ Update an MSR model using new collected data S_u^t.



How to Update an MSR Model Incrementally?

- **Design goals:**
  - Retain and slightly drift existing interest vectors to match with the newly collected interactions.
  - Detect the occurrence of new interests in each time span.
  - Expand new interests adaptively in each time span.



Why Existing Incremental Learning Methods Fail?

Type	Main idea	Limitations	Retaining	Detection	Expansion
Reservoir-based methods (e.g., [1,2])	Select historical samples from the reservoir for model updating based on prioritizing recency or the extent of being forgotten	High cost	×	×	×
Model-based methods (e.g., [3,4])	Enforce regularization terms, which regularly restrains the model parameters rather than user latent representations	Fixed model capacity	√	×	×
Model-expansion methods (e.g., [5,6])	Expand the model capacity to cope with new knowledge during incremental learning	Low self-adaptivity	√	√	×

[1] F. Mi, X. Lin, and B. Faltings, ADER: Adaptively Distilled Exemplar Replay Towards Continual Learning for Session-based Recommendation, Recsys, 2020.
[2] F. Mi and B. Faltings, Memory Augmented Neural Model for Incre-mental Session-based Recommendation. IJCAI,2020.
[3] J. Huang, Y. Chang, and X. Cheng, How to Retrain Recommender System? A Sequencial Meta-learn Method. SIGIR, 2021.
[4] D. Peng, S. J. Pan, J. Zhang, and A. Zeng, Learning an Adaptive Meta Model-Generator for Incrementally Updating Recommender Systems. KDD,2021
[5] R. Anil, S. Gadanho, and D. Huang, On the Factory Floor: ML Engineering for Industrial-Scale Ads Recommendation Models. Recsys, 2022.
[6] T. Chen, I. Goodfellow, and J. Shlens, Net2Net: Accelerating Learning via Knowledge Transfer. arXiv, 2015.

Problem Setting

Problem Setting

- **Notations** (in time span t):

Notation	Description
$S_u^t = \{i_{1,u}^t, i_{2,u}^t, \cdots, i_{n,u}^t\}$	User u's historical interaction sequence
$i_{a,u}^t$ and $e_{a,u}^t$	Target item and its embedding
$H_u^t = \{h_{1,u}^t, h_{k,u}^t, \ldots, h_{K,u}^t\}$	Multi-interest user representation with K interests

- **Incremental MSR:**
  - In each time span t, we aim to learn an MSR model M_t which can map a user's interaction sequence S_u^t into a multi-interest user representation H_u^t :

$$M_{t,} H_u^t \leftarrow \text{Incremental MSR}(S_u^t, M_{t-1}, H_u^{t-1})$$

The IMSR Framework

The IMSR Framework

Training procedure:

Phase1: Detect interests with NID.

Phase2: Retain existing interests with EIR.

Phase3: Update model on a joint loss.

Phase4: Trim redundant interests with PIT.

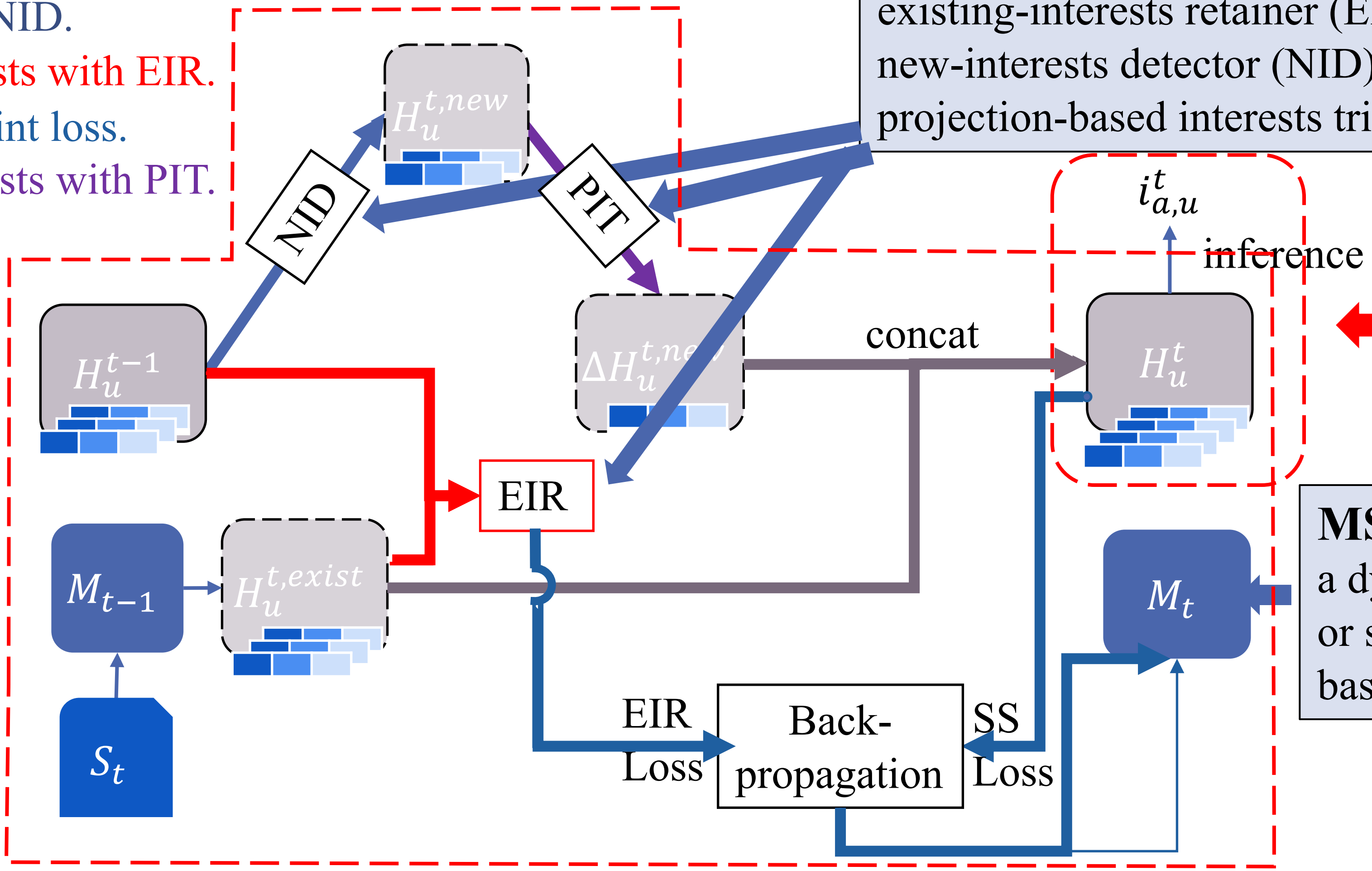
Phase5: Concatenate interests.

Three modules for incremental learning:

existing-interests retainer (EIR)

new-interests detector (NID)

projection-based interests trimmer (PIT)



Inference procedure

MSR model M_t: a dynamic routing or self-attention-based MSR model.

Existing-interests Retainer (EIR)

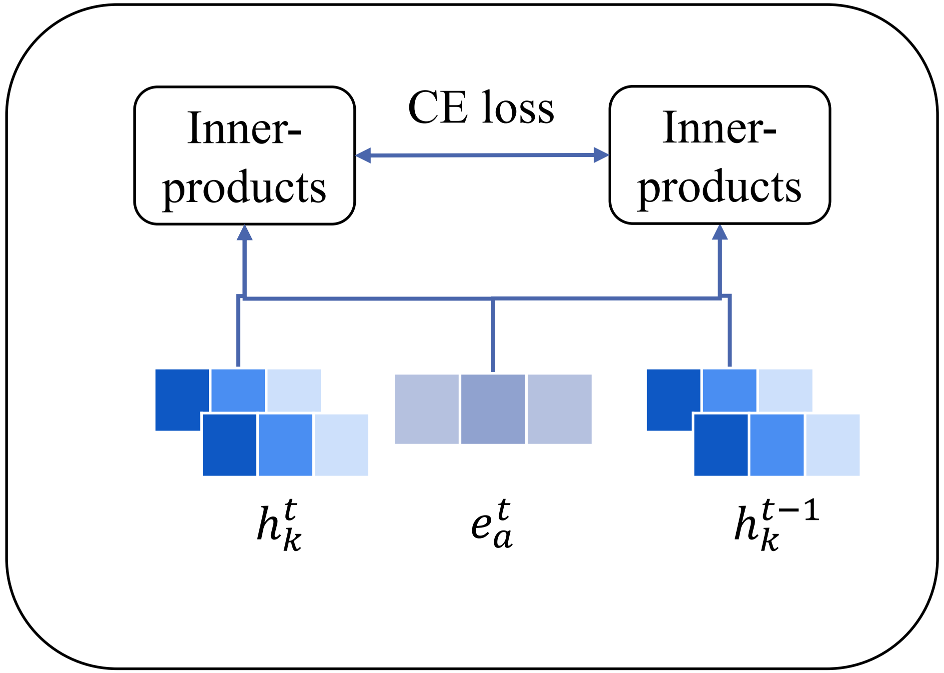
- Encourage the inner-product between each existing interest h_k^t at time span t and target item embedding e_a^t to be close to that between existing interest h_k^{t-1} at time span $t-1$ and e_a^t using cross entropy loss [1]:

$$L_{k,u}^t = L_{CE}(\sigma\left(\frac{e_a^t {h_k^t}^\top}{\tau}\right), \sigma\left(\frac{e_a^t {h_k^{t-1}}^\top}{\tau}\right))$$

- Sum over all existing interests H_k^t to obtain the total EIR loss:

$$L_{EIR}^t = \sum_{S_u^t \in S_t} \sum_{k=1}^{|H_k^t|} L_{k,u}^t$$

EIR: Existing Interests Retainer



[1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” arXiv, 2015.

New-interests Detector (NID)

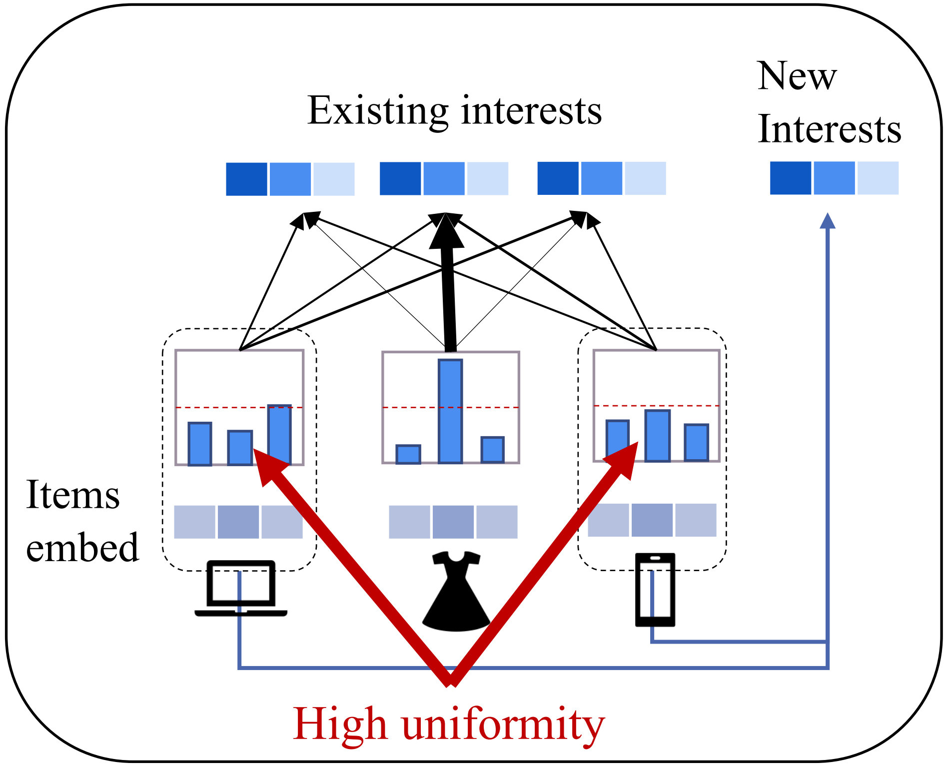
- Calculate the probability vector $[s_{i,1}, \ldots, s_{i,K^t}]$ of item i belonging to all existing interests $h_1^{t-1}, \ldots, h_{K^t}^{t-1}$ using inner product $s_{i,k} = e_i {h_k^{t-1}}^\top$
- Intuitively, the uncertainty of item i belonging to existing interests is proportional to the **uniformity** of $[s_{i,1}, \ldots, s_{i,K^t}]$ and can be measured by Query Sparsity Measurement[1]:

$$\text{Uncertainty}(i) = \frac{1}{K^t}\sum_{k=1}^{K^t} s_{i,k} - \ln\sum_{k=1}^{K^t} e^{s_{i,k}} + lnK^t$$

- New interests are detected if the average uncertainty of items in S_u^t exceeds a threshold c_1:

$$\overline{\text{Uncertainty}(i)} > c_1, i \in S_u^t$$

NID: New Interests Detector

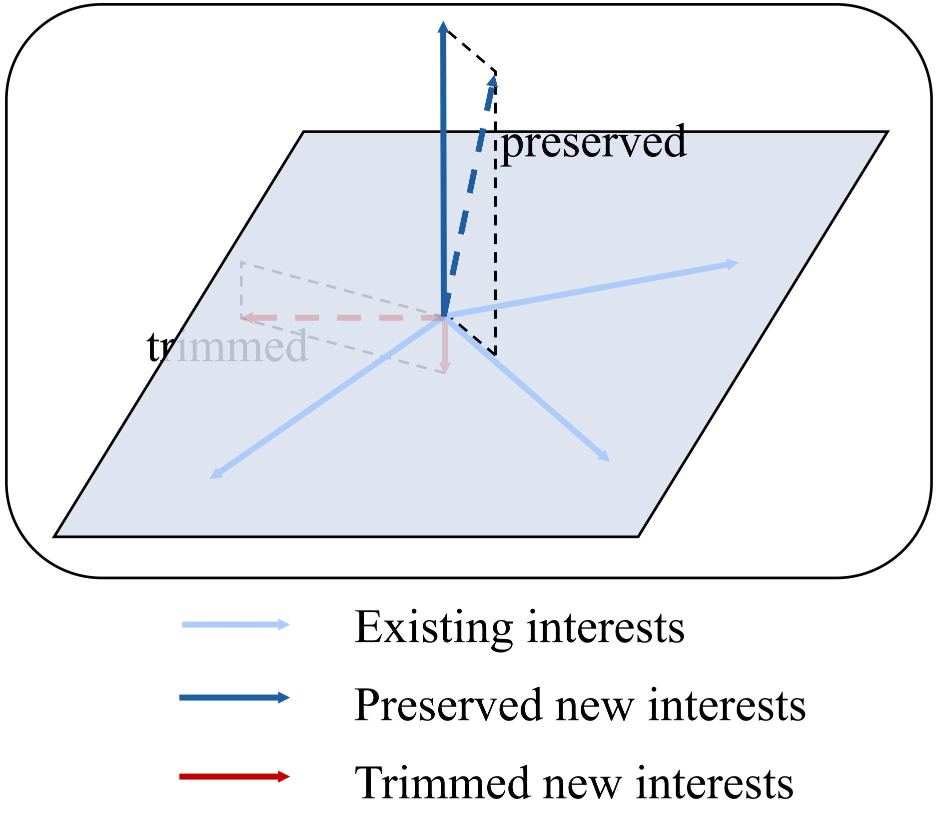


[1] Haoyi Zhou and Shanghang Zhang etc., Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting, AAAI2021

Projection-based Interest Trimmer (PIT)

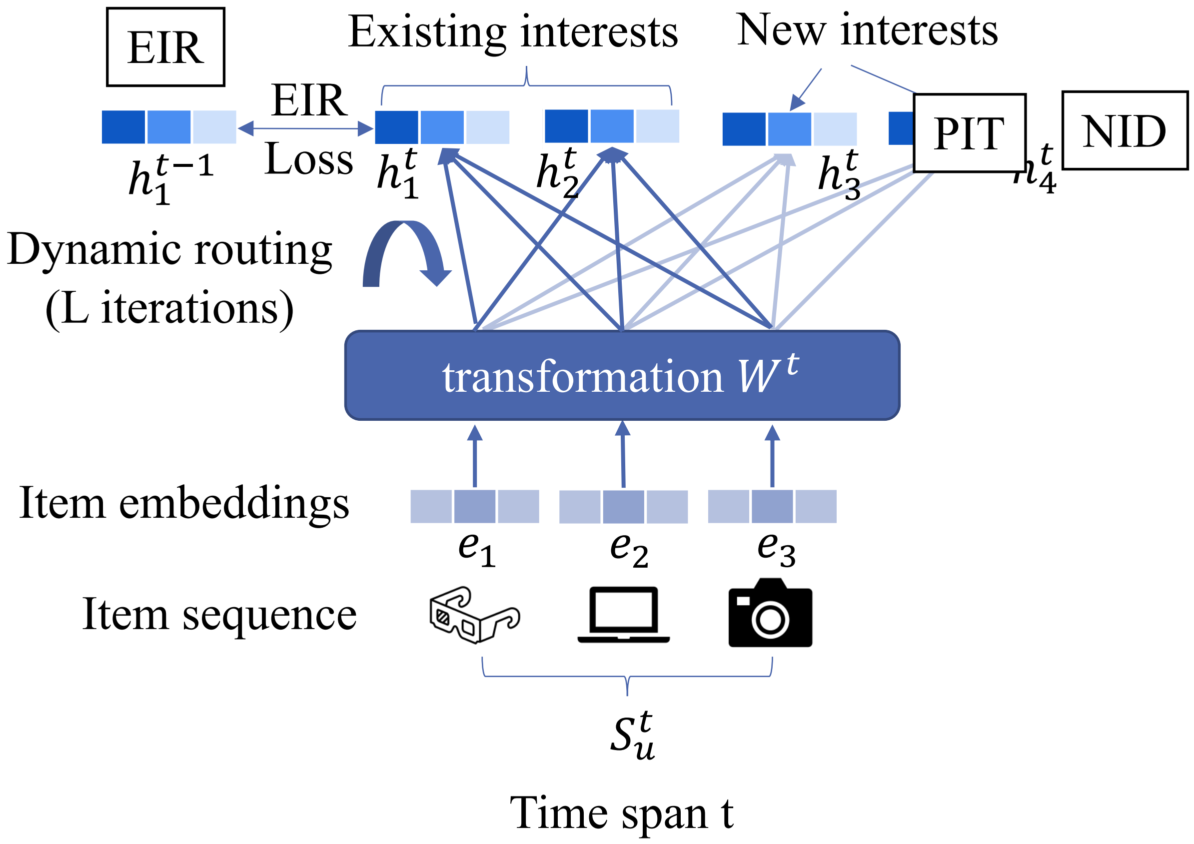
- **Step 1: Obtain new interests via projection:**
  - Learn δK new interests for each user (omit u, t):
  $$H^{new} = (h_1^{new}, \dots, h_{\delta K}^{new})$$
  - Project new interests on existing interests hyperplane:
  $$h_k^{proj} = M_{exist} M_{exist}^\top \left(M_{exist} M_{exist}^\top\right)^{-1} h_k^{new}$$
  - Preserve the orthogonal part:
  $$\Delta h_k^{new} = h_k^{new} - h_k^{proj}$$
- **Step 2: Trimming insignificant interests:**
  - Remove new interest vectors with small L2-norm:
  $$\|\Delta h_k^{new}\|_{L_2} < c_2$$

PIT: Projection-based Interest Trimmer

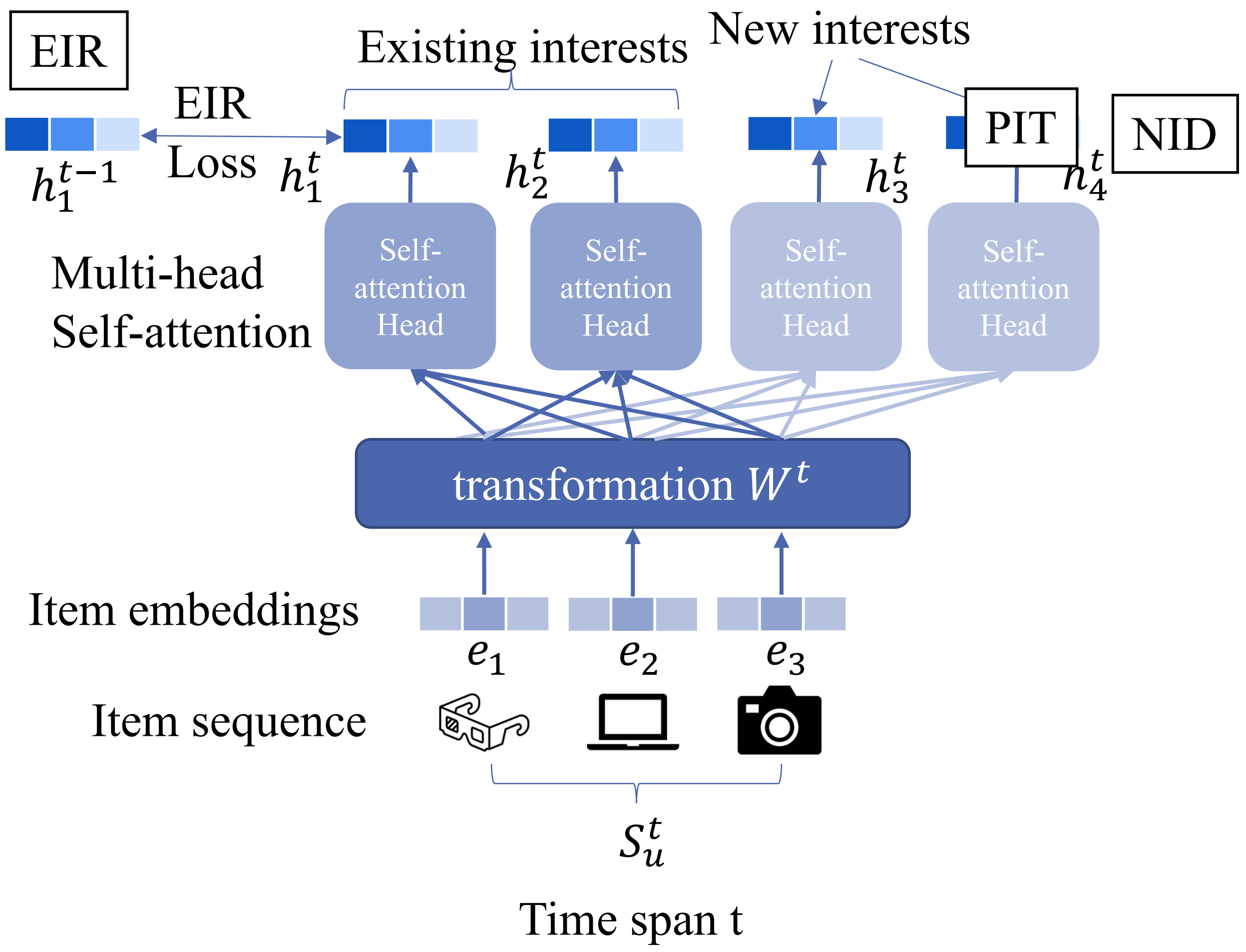


Apply IMSR to Incrementally Update Two MSR Models

- **Dynamic routing based MSR (DR):**



- **Self-attention based MSR (SA):**



Training and Inference Procedure

- **Training in time span t:**
  - update the model parameters using L_{SS}^{t} and L_{EIR}^{t} :

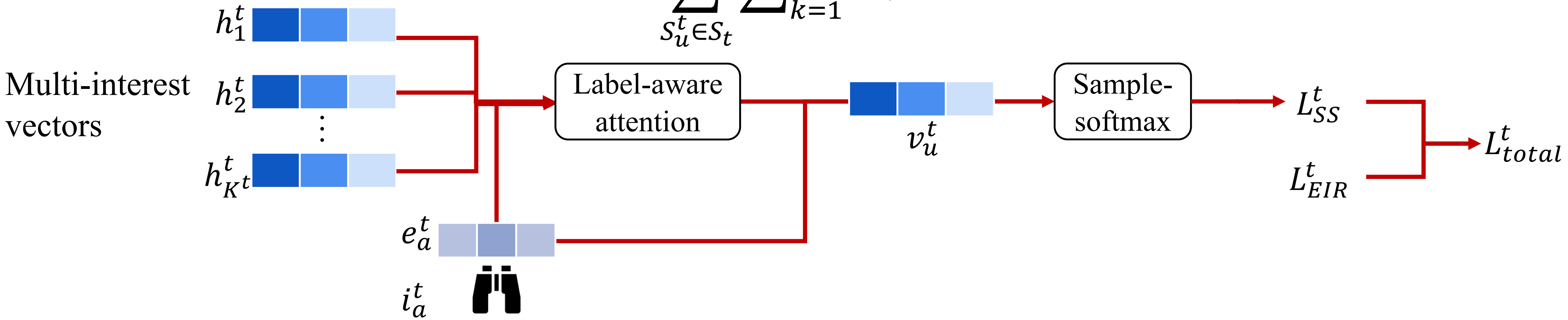
$$L_{total}^{t} = L_{SS}^{t} + L_{EIR}^{t}$$

Label aware attention $v_u^t = \sum_{k=1}^{K_u^t} \text{softmax}\left(e_a^t {h_k^t}^{\top}\right) h_k^t$

Sample-softmax $L_{SS}^{t} = \sum_{S_t} log \frac{\exp(v_u^t {e_a^t}^{\top})}{\sum_{i \in I'} \exp(v_u^t {e_i^t}^{\top})}$

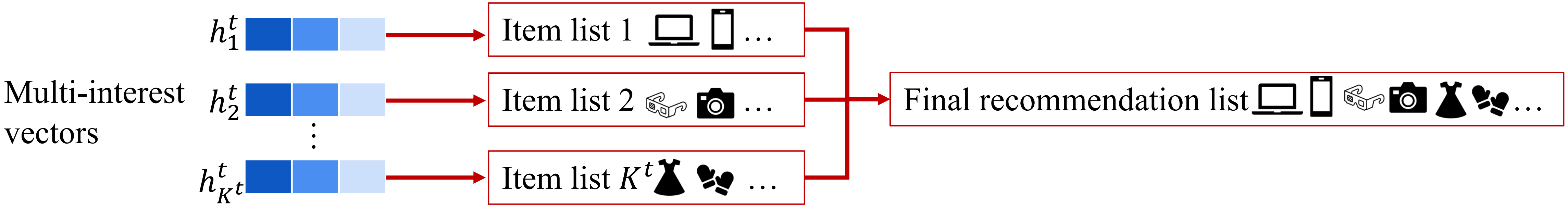
($I' \subset I \backslash \{i_a^t\}$ is a small sampled subset of I)

$$L_{EIR}^{t} = \sum_{S_u^t \in S_t} \sum_{k=1}^{K_u^{t-1}} L_{k,u}^{t}$$



Training and Inference Procedure

- **Inference in time span t:**
  - Allocate the top-N items according to inner-product score $h_1^t {e_a^t}^\top$ for each interest.
  - Union the item lists allocated by each interests to obtain the final recommendation list.



Experiments

Experiment Settings

- **Datasets & Splitting:**

Dataset	#users	#items	#interactions pre-training	1	2	3	4	5	6
Electronics	87,912	234,621	1,689,188	224,421	428,149	329,194	129,482	481,491	196,451
Clothing	285,464	376,859	5,748,920	864,371	574,922	957,329	1,134,792	943,422	1,274,084
Books	459,133	313,966	8,898,041	1,345,234	1,324,545	1,852,324	1,593,281	1,349,281	1,433,376
Taobao	976,779	1,708,530	85,384,110	12,329,481	14,481,123	22,129,123	9,329,128	14,238,129	12,877,126

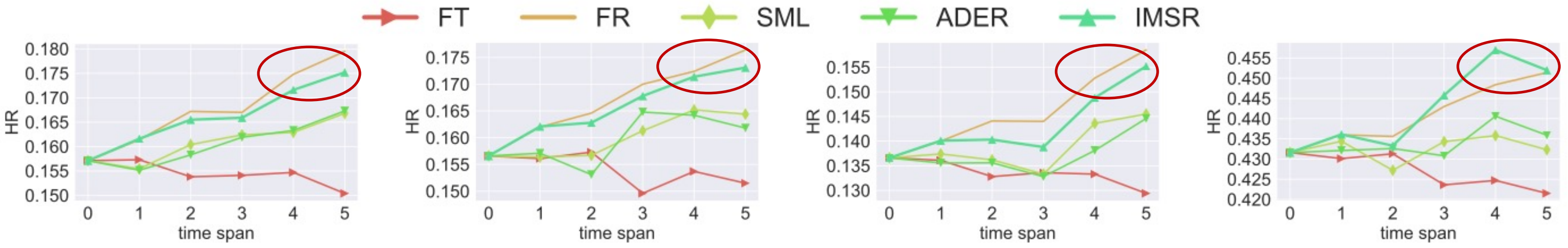
- **Evaluation metrics:** Hit Ratio, NDCG on Top 20
- **Base models:** DR models: MIND [1], ComiRec-DR [2], SA model: ComiRec-SA [2].
- **Compared Learning Methods:** Full Retraining (FR), Fine-tuning (FT), SML (model-based IL, 2021), ADER (reservoir-based IL, 2020).
- **Hyperparameter setting:** $K_u^0 = 4, \delta K = 3, c_1 =$0.6(Taobao)/0.4(Else), $c_2 = 0.3$

[1] W. Li, and D. L. Lee, Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. SIGIR,2019
[2] Zhou, H. Yang, and J. Tang, Controllable Multi-Interest Framework for Recommendation. KDD, 2020

Performance Comparison

Base model	Learning method	Electionics HR	Electionics NDCG	Electionics RI	Clothings HR	Clothings NDCG	Clothings RI	Books HR	Books NDCG	Books RI	Taobao HR	Taobao NDCG	Taobao RI
MIND	FR	16.03	16.43	11.15	16.23	15.98	10.57	13.82	11.95	10.47	43.29	24.90	2.63
	FT	14.75	14.46	-	14.45	14.68	-	12.34	10.98	-	42.09	24.35	-
	SML	15.41	15.17	4.71	15.27	14.81	3.26	13.12	11.12	3.97	42.88	24.58	1.54
	ADER	15.64	14.98	4.84	15.62	15.20	5.76	12.92	11.48	4.64	42.90	24.24	1.05
	IMSR	15.81*	15.71*	**7.93**	15.81*	15.71*	**8.19**	13.99*	11.94*	**11.18**	43.94*	25.66*	**4.76**
ComiRec-DR	FR	17.00	16.79	9.85	16.91	16.75	9.82	14.79	12.79	12.06	44.29	25.87	4.23
	FT	15.41	15.35	-	15.36	15.28	-	13.30	11.30	-	42.62	24.68	-
	SML	16.16	15.85	4.09	16.08	15.77	3.92	13.92	11.85	4.74	43.28	24.89	1.28
	ADER	16.12	15.90	4.10	16.02	15.84	3.96	13.73	11.96	4.43	43.44	25.00	1.68
	IMSR	16.80*	16.48*	**8.20**	16.74*	16.47*	**8.38**	14.46*	12.48*	**9.51**	44.48*	26.00*	**4.72**
ComiRec-SA	FR	17.15	16.95	10.82	16.74	16.87	8.83	14.86	12.85	11.66	44.31	25.75	4.54
	FT	15.31	15.46	-	15.49	15.39	-	13.46	11.35	-	42.44	24.58	-
	SML	15.96	15.99	3.83	15.90	15.88	2.89	13.78	11.71	2.72	43.17	24.83	1.47
	ADER	16.32	15.88	4.63	16.14	15.88	3.67	13.55	11.98	2.87	43.43	25.00	2.12
	IMSR	16.97*	16.32*	**8.19**	16.94*	16.56*	**8.45**	14.38*	12.49*	**8.30**	44.58*	26.11*	**5.48**

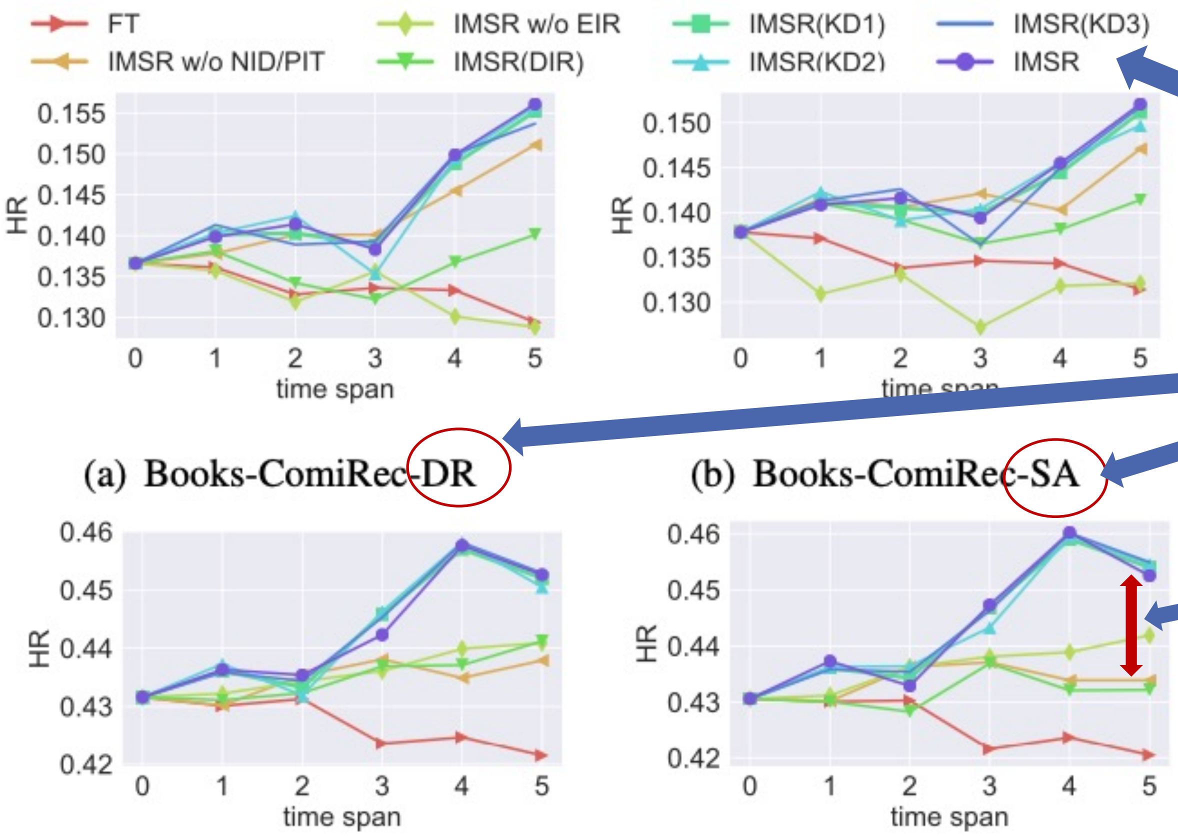
IMSR achieves 3.77%, 3.89%, 4.21%, 4.76% relative improvements on NDCG compared to the second best incremental learning methods on four datasets.



(a) Electronics-HR (b) Clothing-HR (c) Books-HR (d) Taobao-HR

IMSR is slightly inferior to FR by only using the newly collected interactions.

Effects of Different Modules in IMSR



(a) Books-ComiRec-DR (b) Books-ComiRec-SA

(c) Taobao-ComiRec-DR (d) Taobao-ComiRec-SA

All components in IMSR contribute to the recommendation performance.

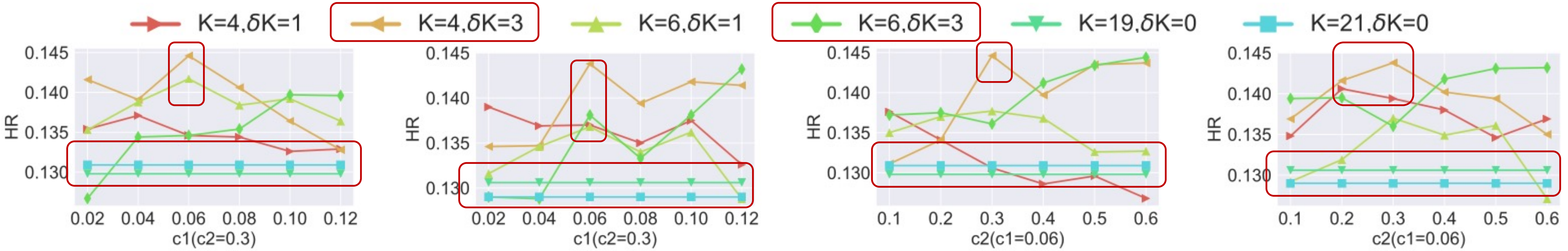
The effectiveness of all three components is insensitive to the base model.

On Taobao, the effectiveness of the NID and PIT is more significant.
→ users in Taobao develop new interests fast due to the richness of item categories.

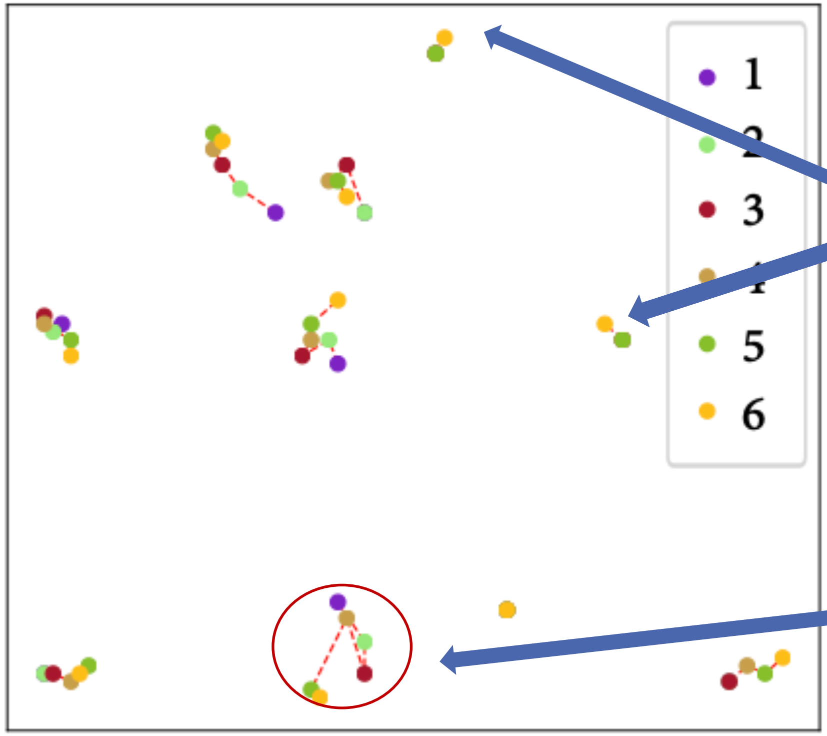
	dataset	t_0	t_6
average interests number of all users	Books	4	5.6
	Taobao	4	9.2

Parameter Sensitivity

- Hyperparameters c_1 and c_2:
  - c_1 controls the new interests detecting sensitivity and c_2 controls the strictness of redundant interest trimming. The model achieves the highest performance with moderate values of c_1 and c_2 in most cases.
- Interests number K and δK:
  - Best performance is achieved when K=6, δK=3 on *Taobao*, K=4, δK=3 on *Books*.
  - K=19, δK=0 (refer to creating all the interest vectors in advance) is far below the performance of IMSR with K = 4,δK = 3, which confirms using interests expansion strategy is more effective than creating too many interests in advance.



Interest Visualization



(b) t-SNE visualization of one user interest evolution among different time spans (colors).

Different interests are located in different places, which reflects the effectiveness of PIT and NID in preventing learning redundant interests.

Same interest in different time spans linked with red dashes locate closely, which shows that EIR prevents dramatical drift of existing interests.

Conclusions & Future work

- **New Framework IMSR** for incremental multi-interest sequential recommendation :
  - **Retain** and slightly drift existing interest vectors. → EIR
  - **Detect** the occurrence of new interests automatically. → NID
  - **Expand** new interests adaptively. → PIT
- **Prominent Performance** on two kinds of multi-interest sequential recommendation models
  - Apply IMSR on two kinds of MSR models: dynamic-routing-based and self-attention-based models.
  - Conduct extensive experiments on four real datasets where our proposed framework achieve superior performance compared with the existing incremental learning methods.
- **Future Work:**
  - How to compress user interest vectors under memory space limit?
  - How to remove interests being inactive for a long while?

Thank you !

- E-Mail: Cloudcatcher.888@sjtu.edu.cn
- Github: https://github.com/Cloudcatcher888/IMSR
- Homesite: https://cloudcatcher888.github.io

Q&A