MELT: Mutual Enhancement of Long-Tailed User and Item for Sequential Recommendation
--SIGIR 2023 Full Paper-

Kibum Kim, Dongmin Hyun, Sukwon Yun, Chanyoung Park

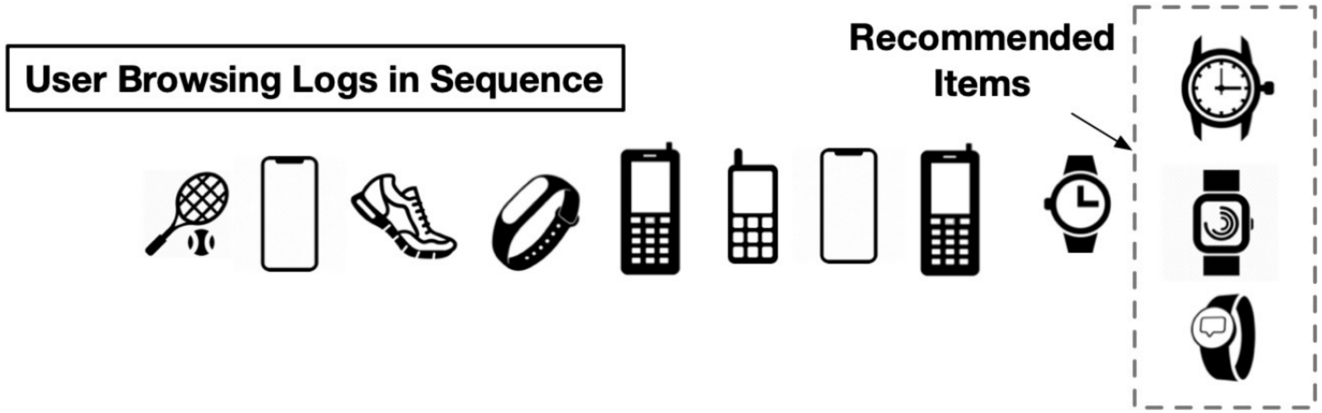
Presenter: Kibum Kim

Master Student
Department of Industrial & Systems Engineering
KAIST

TABLE OF CONTENTS

Background

WHAT IS SEQUENTIAL RECOMMENDATION?



[E.g., Item sequence [1]]

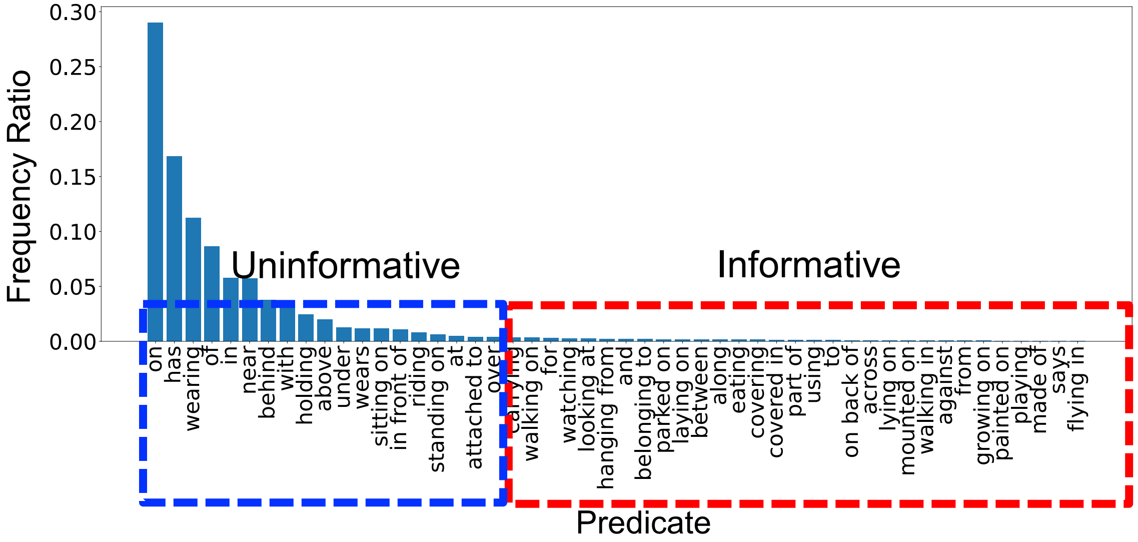
- **General Recommendation**
  - No Sequence
  - Predict a future item with static preference
  - E.g. predicted item – Phone

- **Sequential Recommendation**
  - Consider Sequence
  - Predict a future item with dynamic preference
  - E.g. predicted item – Watch

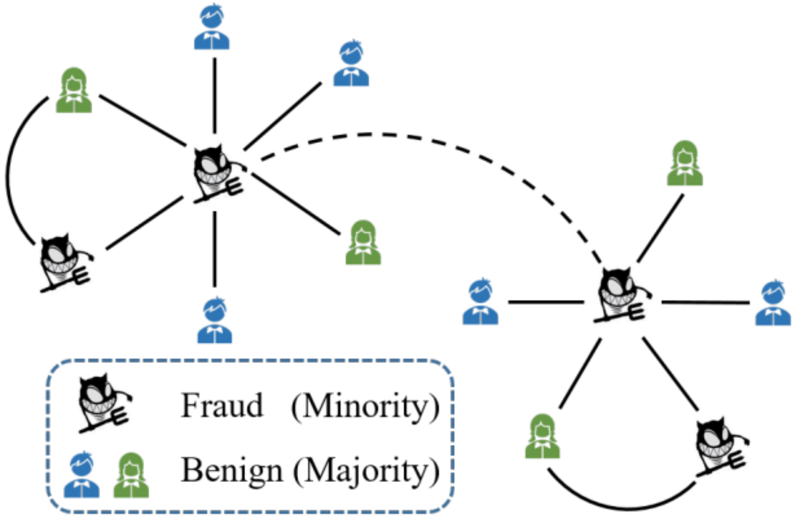
[1] Déjà vu: A Contextualized Temporal Attention Mechanism for Sequential Recommendation. Wu et al. WWW'20

WHAT IS LONG-TAILED PROBLEM?

- Following the long-tailed distribution, the learned model is biased towards head instances
  - The learned model struggles to accurately predict the tail instances
  - Reason: The representation of tail instances are not generalized well



[Long-tailed predicate distribution on Scene Graph Generation [1]]



[Fraud detection on GNN [2]]

However, the tail instances are considered as the important ones

Scene Graph Generation: Informative predicate / Fraud Detection: Fraud

[1] Unbiased Heterogeneous Scene Graph Generation with Relation-aware Message Passing Neural Network. Yoon et al. AAAI'23

[2] Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. Liu et al. WWW'21

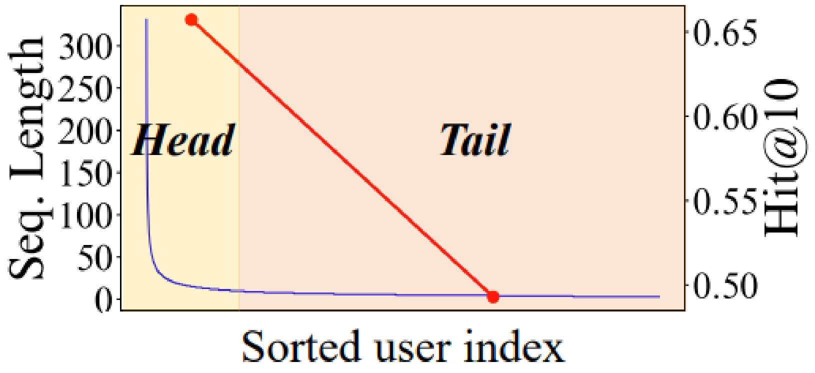
Motivation

LONG-TAILED USER AND ITEM PROBLEM

Long-tailed user problem: The users with few interactions take a significant portion (Blue line) while showing inferior performance (Red line)

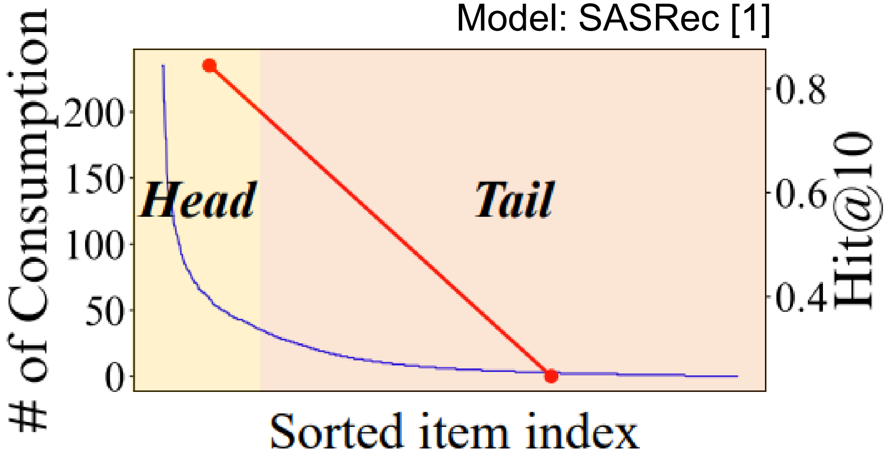
- Why important? The revenue would be increased significantly if tail users becomes head users

Model: SASRec [1]



[Long-tailed user distribution on Amazon Music data]

Model: SASRec [1]



[Long-tailed item distribution on Amazon Music data]

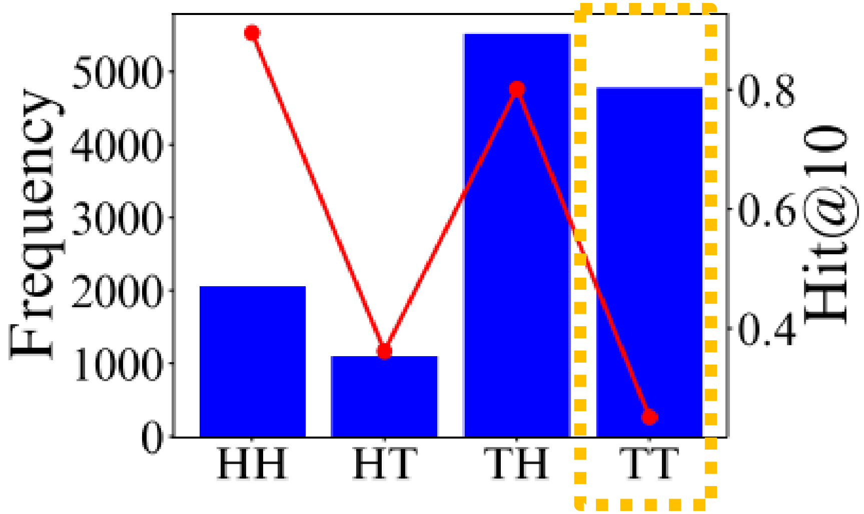
Long-tailed item problem: The model learned with long-tailed distribution (Blue line) is hard to predict the appropriate tail items[1] (Red line).

- Why important? Recommending appropriate tail items makes users stay on their online systems → Increase revenue

[1] Self-Attentive Sequential Recommendation. Kang et al. ICDM'18

IMPORTANCE OF JOINT CONSIDERATION

- In previous page, we observe that long-tailed user & item problems occur in sequential recommendation
- The more important thing is that the tail users consumed with tail items[1] (**TT**) take a large portion
  - What it worse, TT group shows the inferior performance (Red line)



[Necessity of Joint Consideration]

Existing works cannot effectively improve the performance of TT group!

They only focus on either long-tailed user or item problems

1. The users with few interactions consume the tail item in test set.

EXPLORATION OF TT GROUP

- **Straightforward approach to tackle the TT group**
  - Naive combination of the two existing works
  - Model for long-tailed user problem (ASReP [1]) + Model for long-tailed item problem (CITIES [2])

- **Limitation**
  - 1) The naïve combination (CITIES+ASReP) causes the conflict between methods → Degrading performance
  - 2) What it worse, its complexity linearly increases.

Model	Hit@10 on TT			
	Music	Beauty	Automotive	Behance
ASReP	0.246	0.160	0.114	0.287
CITIES	0.279	0.191	0.130	0.298
ASReP+CITIES	0.273 (-2.2%)	0.169 (-11.5%)	0.124 (-4.6%)	0.296 (-0.7%)
MELT	**0.312** (+11.8%)	**0.197** (+3.1%)	**0.149** (+14.6%)	**0.371** (+24.5%)

The naïve combination cannot effectively tackle the TT group

→ It motivate us to propose the new framework

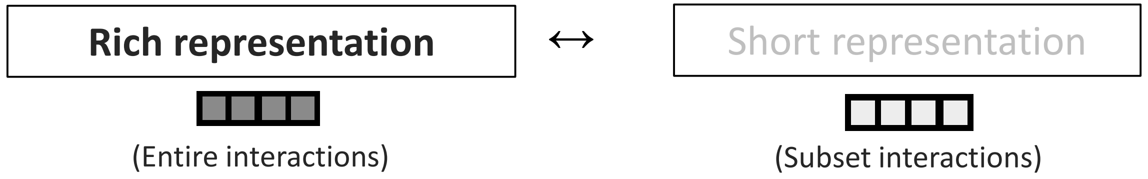
[1] Augmenting Sequential Rec

[2] CITIES: Contextual Inference of Tail-Item Embeddings for Sequential Recommendation. Jang et al. ICDM'20

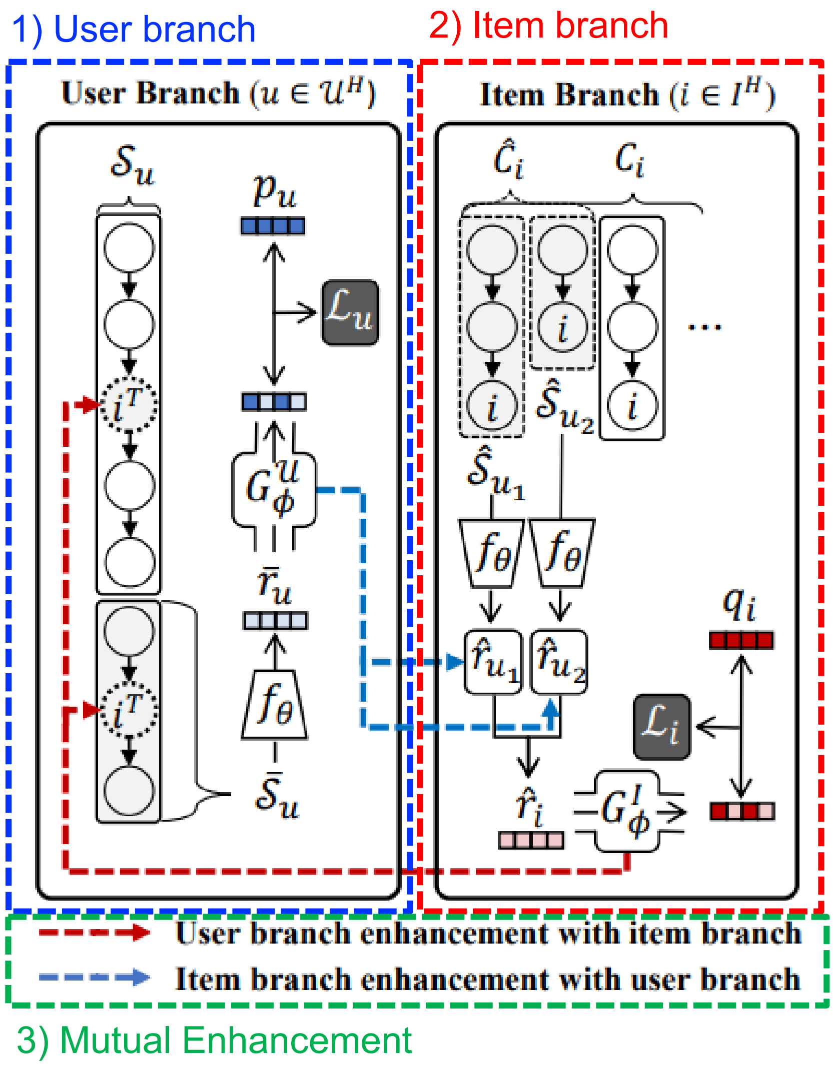
Method

METHOD: OVERALL FRAMEWORK

- Bilateral Branches
  - 1) **User Branch**: Alleviate the long-tailed user problem
  - 2) **Item Branch**: Alleviate the long-tailed item problem
    - Key idea: Using head instances, utilize the knowledge gap



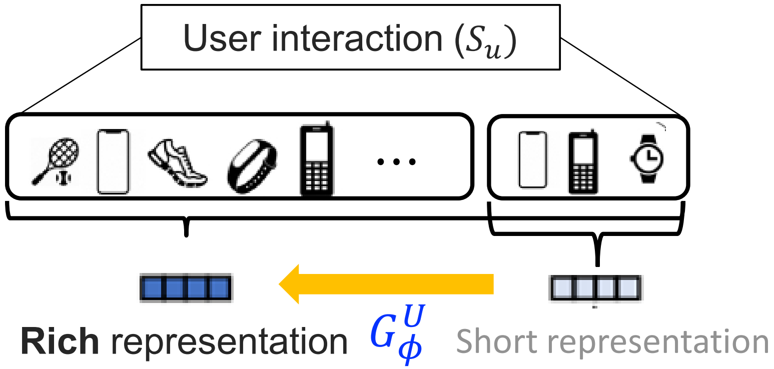
- **3) Mutual Enhancement**
  - Effectively alleviate the long-tailed problem for each branch (Conflict ↓)

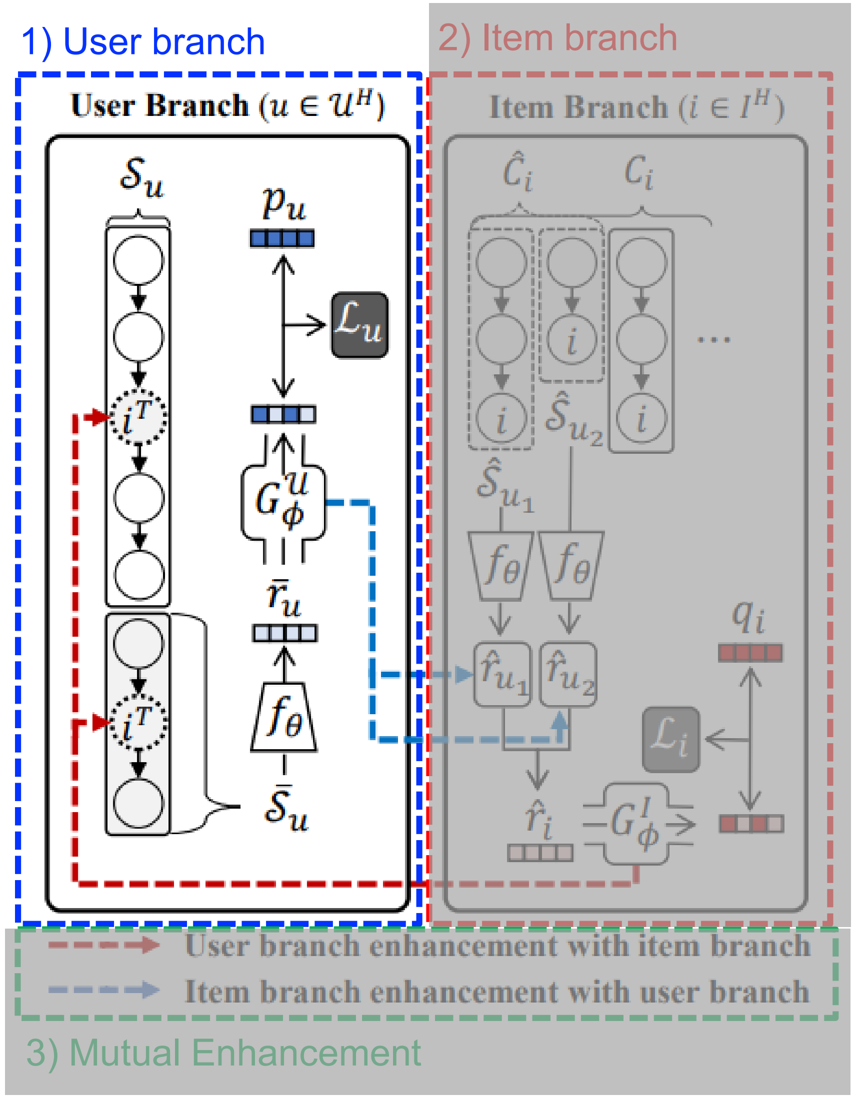


METHOD: 1) USER BRANCH - TRAIN

- **Method – Train(w/ Head users)**
  - **Rich** representation ($p_u = f_\theta(S_u)$): Encode the **entire** interactions (S_u)

    Sequential encoder (SASRec [1])
  - Short representation ($\bar{r}_u = f_\theta(\bar{S}_u)$): Encode the truncated interactions ($\bar{S}_u$)
    - We truncate the recent interactions
  - Train G_ϕ^U generator to contain the knowledge: $\|p_u - G_\phi^U(\bar{r}_u)\|^2$
    - G_ϕ^U: Generate the **Rich** representation given Short representation





[1] Self-Attentive Sequential Recommendation. Kang et al. ICDM'18

METHOD: 1) USER BRANCH - TRANSFER

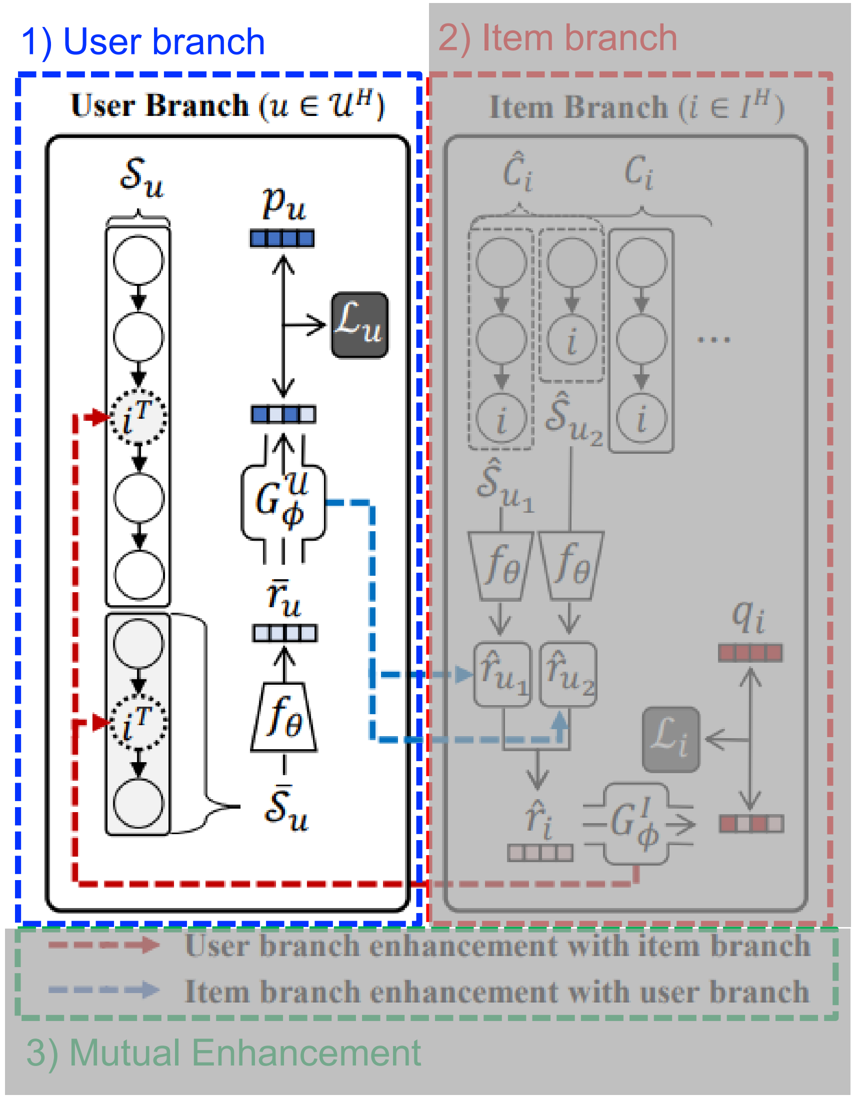
- **Method - Transfer to tail users**
  - Enhance the tail users' representation using knowledge in G_ϕ^U generator

If no user embedding→ p_u= r_u

$$p_u^+ = \boldsymbol{G_\phi^U}\,(\boldsymbol{r_u}) + \beta p_u$$

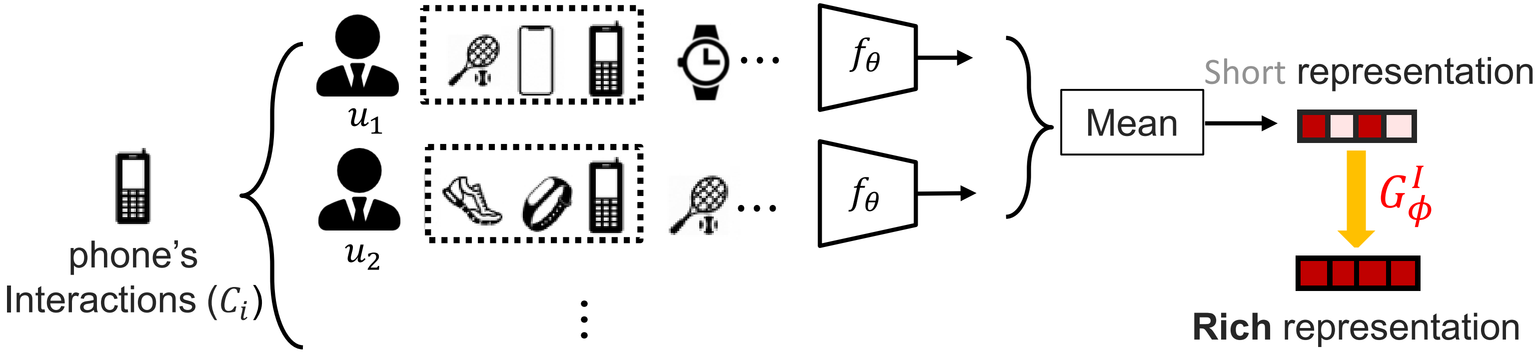
Rich representation

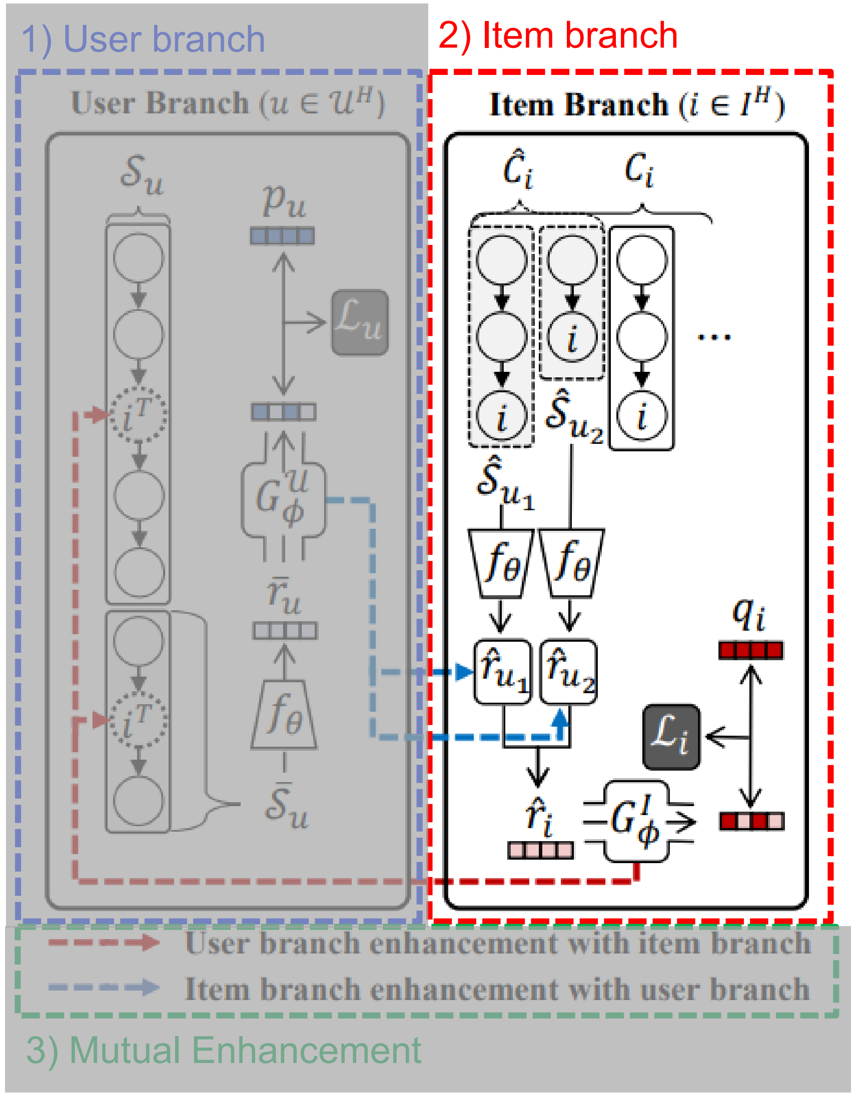
- $r_u = f_\theta(S_u)$: Encode the entire tail user's interaction
- β: Control the weight for original user representation
- p_u: User embedding
  - If there is no user embedding, p_u=$f_\theta(S_u)$



METHOD: 2) ITEM BRANCH - TRAIN

- **Method – Train(w/ Head items)**
  - **Rich** representation (q_i): Item embedding trained by entire interactions
  - Short representation ($\hat{r}_i = \mathrm{Mean}(f_\theta(\hat{S}_u), \hat{S}_u \in \hat{C}_i)$: Mean representation of user subsequences ($\hat{S}_u$) included in the subset ($\hat{C}_i$) that ended with item i
  - Train G^I_ϕ generator to contain the knowledge : $\left\| q_i - G^I_\phi(\hat{r}_i) \right\|^2$
    - G^I_ϕ: Generate the **Rich** representation given Short representation





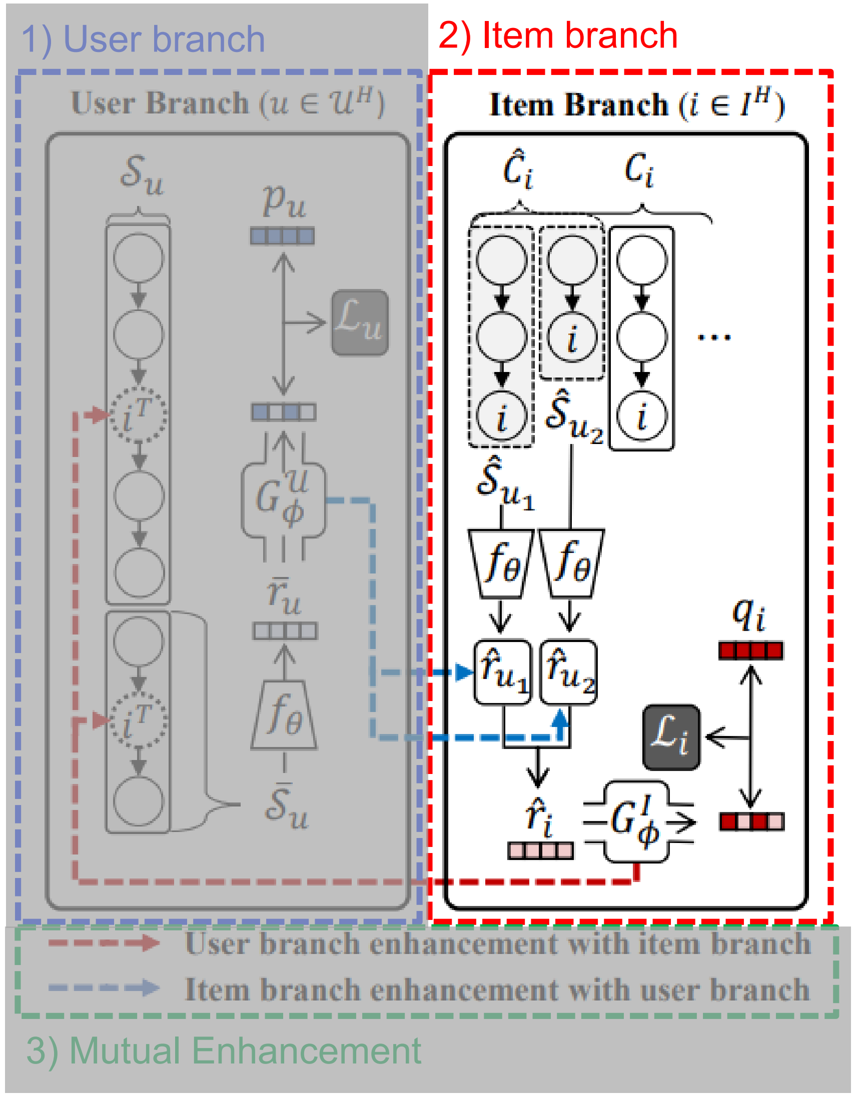
METHOD: 2) ITEM BRANCH - TRANSFER

- **Method - Transfer to tail items**
  - Enhance the tail items' representation using knowledge in G_ϕ^I generator

$$q_i^+ = \boldsymbol{G_\phi^I}\,(\boldsymbol{r_i}) + \gamma q_i$$

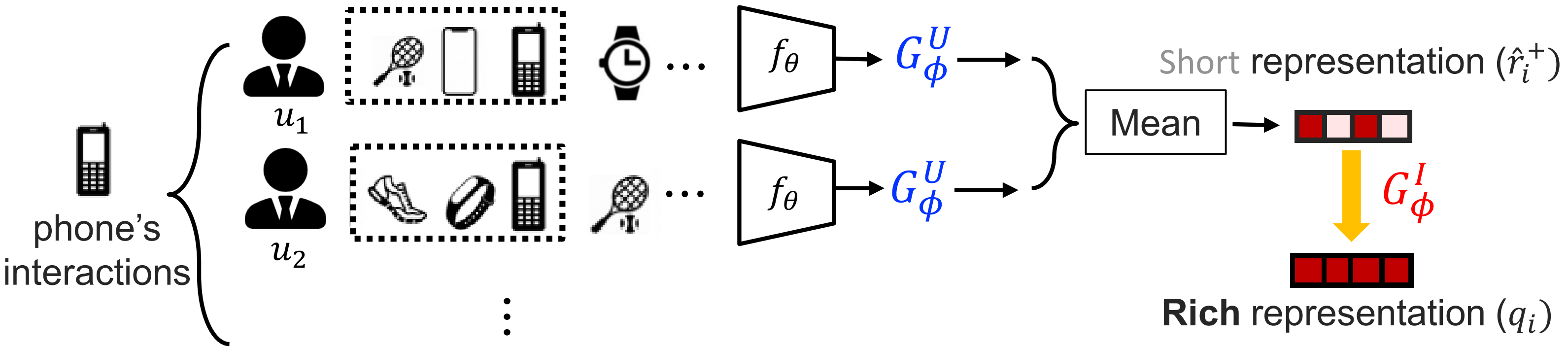
Rich representation

- $r_i = \mathrm{Mean}(f_\theta(\hat{S}_u)), \hat{S}_u \in C_i$: Encode the entire tail item's interaction
- γ: Control the weight for original item representation
- q_i: Item embedding



METHOD: 3) MUTUAL ENHANCEMENT (USER → ITEM)

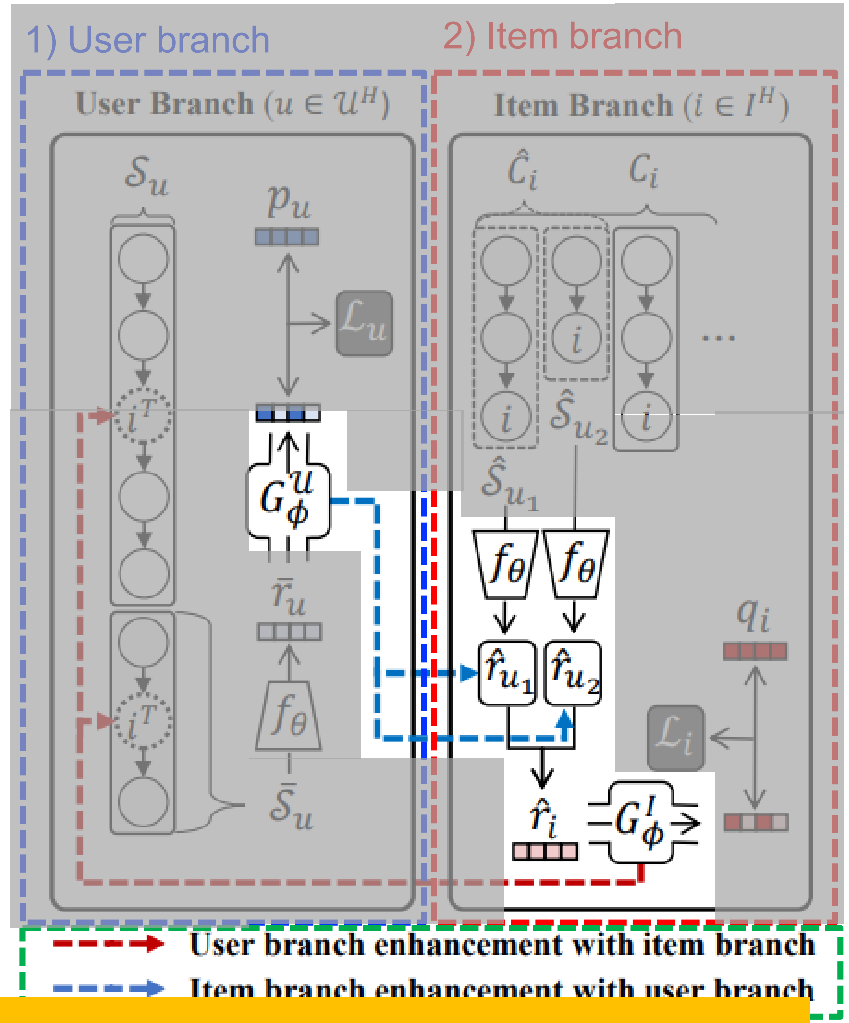
- **1) User branch → Item branch**: Enhance the G_ϕ^I knowledge in item branch using G_ϕ^U knowledge in user branch
  - **Item branch**



  - Enhance the user subsequences' representation based on the G_ϕ^U knowledge

$$\hat{r}_i^+ = \text{Mean}(\underbrace{G_\phi^U(\hat{r}_u)}_{\text{Rich representation}} + \beta p_u)$$

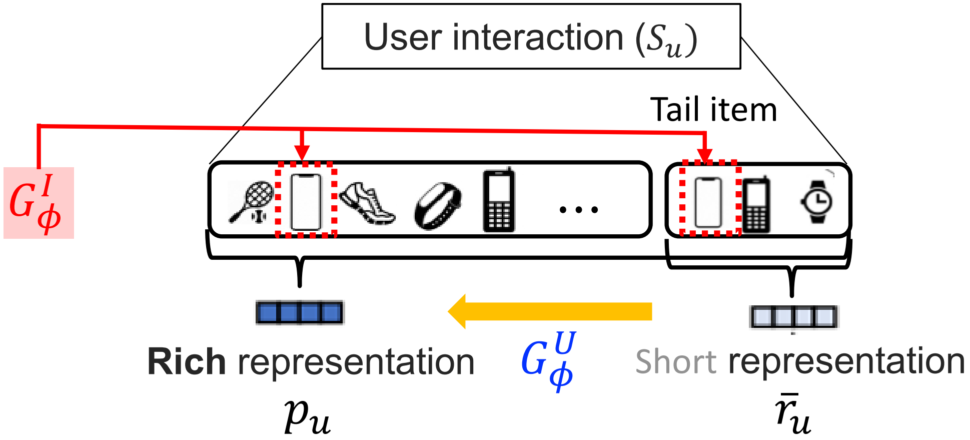
(※ Original user subsequence's representation: $\hat{r}_u$)



By training to predict q_i using the enhanced $\hat{r}_i^+$, we aim to improve the knowledge quality of G_ϕ^I

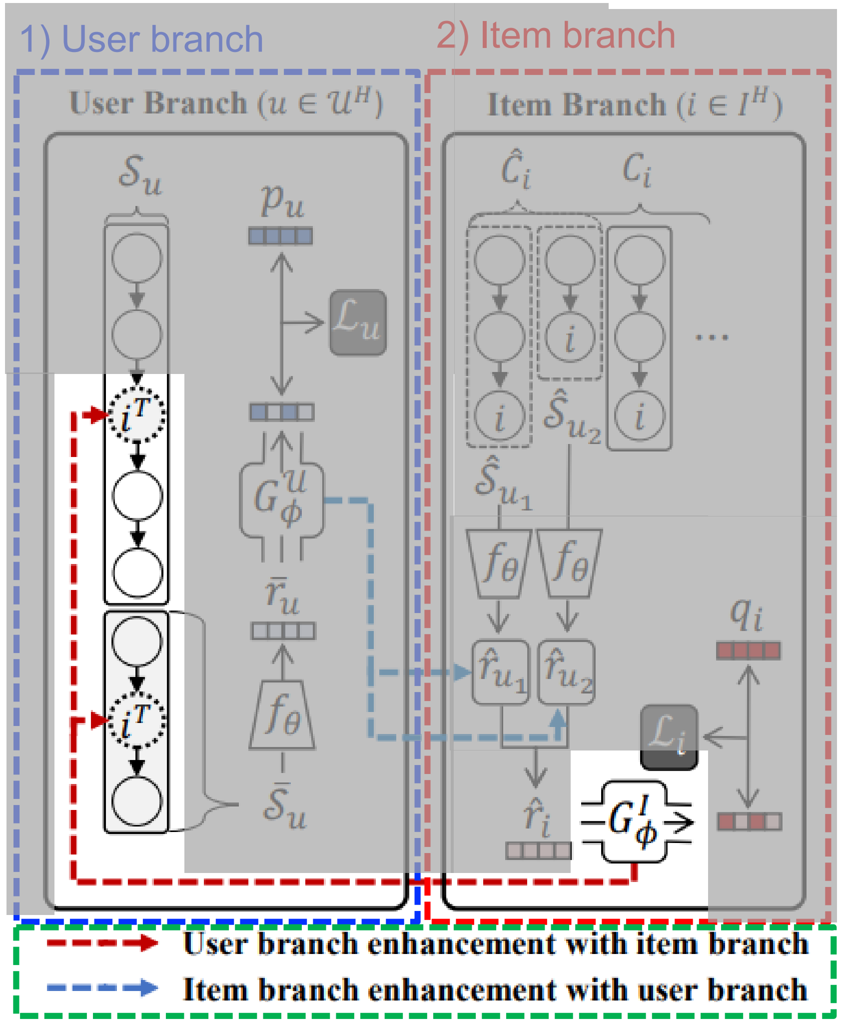
METHOD: 3) MUTUAL ENHANCEMENT (ITEM → USER)

- **2) Item branch → User branch**: Enhance the G_ϕ^U knowledge in user branch using G_ϕ^I knowledge in item branch
  - **User branch**



  - Enhance the input of sequence encoder, i.e., item embedding
    (※ Original input: q_i)

$$q_i^+ = \boldsymbol{G_\phi^I(r_i)} + \gamma q_i$$

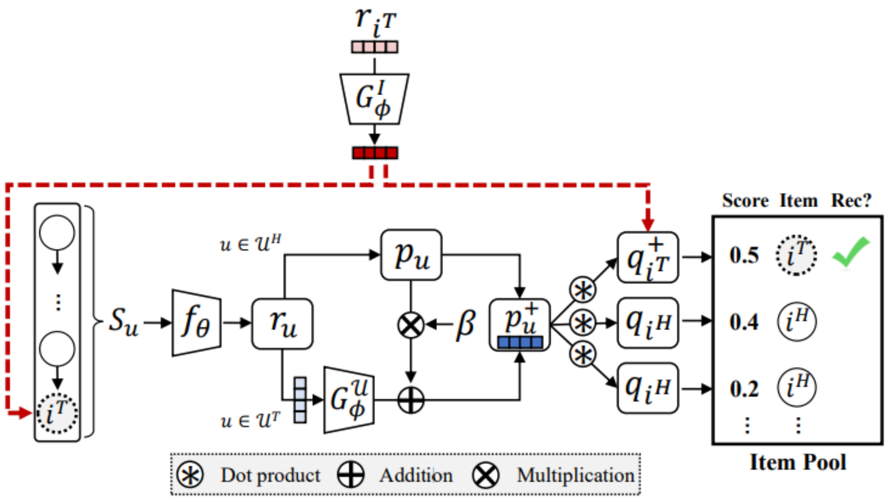


With enhanced p_u and $\bar{r}_u$, we aim to improve the knowledge quality of G_ϕ^U

METHOD: 5) TRAINING & INFERENCE

- **Model Training - loss**
  - $\mathcal{L}_{final} = \underbrace{\lambda_U \sum_{u \in U^H} \mathcal{L}_u}_{\text{User branch}} + \underbrace{\lambda_I \sum_{i \in I^H} \mathcal{L}_i}_{\text{Item branch}} + \mathcal{L}_{rec}$
  - λ_U, λ_I: Hyperparameters for controlling weight of user and item branch, respectively
  - $\mathcal{L}_{rec}$: Next item prediction loss (Cross-entropy loss)

- **Inference**
  - The tail users' representation is enhanced by G_ϕ^U knowledge (pg. 15)
  - The tail items' representation is enhanced by G_ϕ^I knowledge (pg. 17)
    - 1) Input of sequence encoder
    - 2) Recommendation in item pool
  - Top K items with the highest scores are recommended
    - Score: user representation × item representation



[Overview of inference phase]

Experiment

EXPERIMENT: EXPERIMENT SETTINGS

- **Datasets**
  - Amazon product datasets (6), Behance, Foursquare
- **Evaluation Metrics**
  - Hit@K: the ground-truth test item belongs to top K ranked items
  - NDCG@K: consider the ground-truth item's rank in top K items.

$$\text{Hit@K} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \mathbf{1}(Rank_{u,g_i} \leq K)$$

$$\text{NDCG@K} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \frac{\mathbf{1}(Rank_{u,g_i} \leq K)}{log_2(Rank_{u,g_i}+1)}$$

- **Data Statistics**
  - It ranges from large interactions to small interactions

Dataset	# Item	# User	# Int.	Avg $\|\mathcal{S}_u\|$
Clothing	174,484	184,050	1,068,972	4.01
Sports	83,728	83,970	589,029	5.11
Beauty	57,289	52,204	394,908	5.6
Grocery	39,264	32,126	275,256	6.6
Automotive	40,287	34,315	183,567	3.6
Music	20,356	20,165	132,595	5.11
Foursquare	13,335	43,110	306,553	5.12
Behance	32,491	28,915	712,271	22.7

[Data Statistics]

EXPERIMENT: PERFORMANCE COMPARISON

- **Observation**
  - MELT improves the performance on tail user/item groups without sacrificing the head user/item groups
  - As MELT is model-agnostic, MELT based on state-of-the-art model, FMLP [1], also improves the performance

[MELT+SASRec]

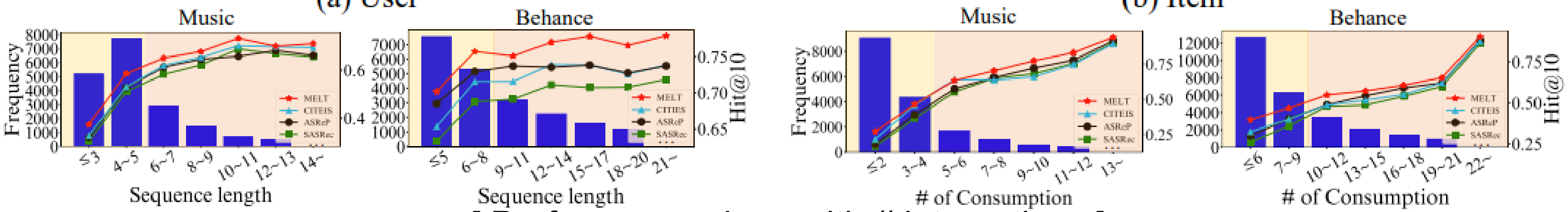
Dataset	Model	Overall		Head User		Tail User		Head Item		Tail Item		Mean		Improvement (%)	
		HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	Overall	Mean
Clothing	BERT4Rec	0.3892	0.2455	0.3852	0.2464	0.3903	0.2453	**0.8097**	0.5145	0.0128	0.0048	0.3995	0.2528	11.4	11.4
	Tail-Net	0.3769	0.2252	0.3692	0.2219	0.3789	0.2260	0.7882	0.4736	0.0087	0.0029	0.3863	0.2311	17.4	17.7
	INSERT	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-	-
	ASReP	0.4137	0.2550	0.4292	0.2698	0.4096	0.2511	0.7605	0.4894	0.1033	0.0452	0.4257	0.2639	5.7	5.3
	SASRec	0.4054	0.2491	0.4202	0.2615	0.4015	0.2458	0.7476	0.4794	0.0991	0.0429	0.4171	0.2574	8.0	7.7
	CITIES	0.4050	0.2474	0.4220	0.2601	0.4005	0.2440	0.7177	0.4621	**0.1252**	**0.0552**	0.4164	0.2554	8.3	8.1
	MELT	**0.4350**	**0.2718**	**0.4462**	**0.2803**	**0.4321**	**0.2696**	0.7864	**0.5270**	0.1205	0.0435	**0.4463**	**0.2801**	-	-
Sports	BERT4Rec	0.4886	0.3197	0.5402	0.3612	0.4782	0.3113	0.7988	0.5451	0.1627	0.0828	0.4950	0.3251	9.4	8.9
	Tail-Net	0.4334	0.2718	0.4637	0.2975	0.4274	0.2667	0.8224	0.5227	0.0247	0.0083	0.4346	0.2738	25.4	26.0
	INSERT	0.3987	0.2359	0.4167	0.2478	0.3950	0.2335	0.7403	0.4459	0.0397	0.0151	0.3979	0.2356	39.3	40.9
	ASReP	0.5034	0.3249	0.5486	0.3603	0.4944	0.3178	0.8107	0.5498	0.1805	0.0886	0.5086	0.3291	6.7	6.6
	SASRec	0.4972	0.3201	0.5439	0.3539	0.4878	0.3133	0.8110	0.5481	0.1676	0.0806	0.5026	0.3240	8.2	8.0
	CITIES	0.5102	0.3283	0.5565	0.3603	0.5008	0.3219	0.7731	0.5328	**0.2339**	**0.1134**	0.5161	0.3321	5.4	5.2
	MELT	**0.5377**	**0.3463**	**0.5848**	**0.3780**	**0.5282**	**0.3399**	**0.8429**	**0.5943**	0.2169	0.0857	**0.5432**	**0.3495**	-	-
Beauty	BERT4Rec	0.4476	0.3005	0.5083	0.3635	0.4337	0.2861	0.7777	0.5410	0.1087	0.0536	0.4571	0.3111	11.1	10.8
	Tail-Net	0.4233	0.2643	0.4736	0.3180	0.4118	0.2520	**0.7991**	0.5068	0.0375	0.0154	0.4305	0.2731	20.9	21.0
	INSERT	0.4069	0.2447	0.4387	0.2724	0.3997	0.2383	0.7764	0.4730	0.0278	0.0103	0.4107	0.2485	27.6	29.1
	ASReP	0.4680	0.3112	0.5402	0.3772	0.4515	0.2962	0.7530	0.5179	0.1755	0.0991	0.4801	0.3226	6.7	6.0
	SASRec	0.4576	0.2995	0.5351	0.3687	0.4399	0.2837	0.7396	0.5013	0.1680	0.0924	0.4707	0.3115	9.8	8.8
	CITIES	0.4599	0.3039	0.5447	0.3786	0.4406	0.2868	0.7063	0.4893	0.2071	**0.1135**	0.4747	0.3171	8.8	7.5
	MELT	**0.5012**	**0.3300**	**0.5673**	**0.3837**	**0.4861**	**0.3178**	0.7806	**0.5581**	**0.2144**	0.0959	**0.5121**	**0.3389**	-	-
Grocery	BERT4Rec	0.4590	0.3168	0.5506	0.4152	0.4380	0.2942	0.7872	0.5692	0.1166	0.0534	0.4731	0.3330	6.6	6.1
	Tail-Net	0.4274	0.2812	0.5160	0.3772	0.4070	0.2591	0.7998	0.5360	0.0387	0.0152	0.4404	0.2969	16.7	16.0
	INSERT	0.4236	0.2651	0.4878	0.3121	0.4088	0.2543	**0.8165**	0.5147	0.0135	0.0047	0.4317	0.2715	20.1	21.6
	ASReP	0.4622	0.3079	0.5575	0.4036	0.4403	0.2859	0.7544	0.5235	0.1573	0.0829	0.4774	0.3240	7.4	6.7
	SASRec	0.4556	0.3036	0.5539	0.3957	0.4330	0.2824	0.7499	0.5206	0.1486	0.0771	0.4714	0.3190	8.9	8.2
	CITIES	0.4588	0.3080	0.5593	0.4005	0.4357	0.2867	0.7245	0.5159	0.1815	**0.0910**	0.4753	0.3235	7.9	7.1
	MELT	**0.4910**	**0.3360**	**0.5802**	**0.4235**	**0.4705**	**0.3159**	0.7805	**0.5783**	**0.1888**	0.0831	**0.5050**	**0.3502**	-	-

(a) User

[MELT+FMLP]

Data	Model	Overall		Tail User		Tail Item	
		HR@10	ND@10	HR@10	ND@10	HR@10	ND@10
Clothing	FMLP	0.392	0.242	0.389	0.239	0.127	0.062
	+CITIES	0.402	0.249	0.396	0.245	0.158	0.078
	+MELT	**0.434**	**0.274**	**0.426**	**0.269**	**0.206**	**0.091**
Sports	FMLP	0.513	0.337	0.502	0.329	0.220	0.118
	+CITIES	0.523	0.343	0.512	0.335	0.239	**0.128**
	+MELT	**0.542**	**0.356**	**0.531**	**0.349**	**0.274**	0.126
Beauty	FMLP	0.474	0.323	0.458	0.307	0.204	0.124
	+CITIES	0.491	0.334	0.475	0.319	0.224	**0.135**
	+MELT	**0.500**	**0.341**	**0.483**	**0.326**	**0.241**	0.127
Grocery	FMLP	0.462	0.314	0.441	0.293	0.169	0.095
	+CITIES	0.479	0.328	0.459	0.308	0.188	**0.105**
	+MELT	**0.488**	**0.333**	**0.467**	**0.313**	**0.206**	0.097
Automotive	FMLP	0.355	0.228	0.337	0.208	0.135	0.075
	+CITIES	0.384	0.249	0.375	0.243	0.149	0.084
	+MELT	**0.390**	**0.253**	**0.380**	**0.246**	**0.174**	**0.086**
Music	FMLP	0.553	0.394	0.518	0.366	0.275	0.141
	+CITIES	0.567	0.406	0.531	0.378	0.301	**0.177**
	+MELT	**0.590**	**0.407**	**0.555**	**0.379**	**0.341**	0.173
Foursquare	FMLP	0.917	0.773	0.901	0.743	0.350	**0.241**
	+CITIES	0.920	0.820	0.904	0.795	0.338	0.224
	+MELT	**0.925**	**0.821**	**0.910**	**0.797**	**0.409**	0.197
Behance	FMLP	0.729	0.531	0.704	0.506	0.394	0.272
	+CITIES	0.736	0.539	0.713	0.516	0.435	**0.285**
	+MELT	**0.751**	**0.549**	**0.731**	**0.529**	**0.474**	0.273

(b) Item



[Performance along with # interactions]

EXPERIMENT: FINE-GRAINED PERFORMANCE

- Fine-grained Performance Comparison
  - HU/HI: the head users (HU) consumed with a head item[1] (HI), …
- MELT jointly alleviates the long-tailed user and item problems
  - 1) MELT vs ASReP(long-tail user): Outperforms on HU/TI and TU/TI groups
  - 2) MELT vs CITIES(long-tail item): Outperform on TU/HI and TU/TI groups
  - 3) MELT vs ASReP+CITIES(long-tail user+item): Outperforms on TU/TI groups → It shows the effectiveness of mutual enhancement manner

Data	Model	HU/HI	HU/TI	TU/HI	TU/TI	Mean
Music	SASRec	0.912	0.325	0.828	0.230	0.574
	ASReP	0.915	0.339	0.832	0.246	0.583
	CITIES	0.896	0.405	0.807	0.279	0.597
	ASReP+CITIES	0.915	0.368	0.828	0.273	0.596
	MELT	**0.928**	**0.418**	**0.884**	**0.312**	**0.636**
Beauty	SASRec	0.807	0.240	0.724	0.152	0.481
	ASReP	0.811	0.245	0.739	0.160	0.489
	CITIES	0.786	0.282	0.687	0.191	0.487
	ASReP+CITIES	**0.820**	0.245	0.743	0.169	0.494
	MELT	0.818	**0.291**	**0.775**	**0.197**	**0.520**
Automotive	SASRec	0.701	0.133	0.646	0.108	0.397
	ASReP	0.724	0.146	0.667	0.114	0.413
	CITIES	0.675	0.158	0.625	0.130	0.397
	ASReP+CITIES	0.726	0.147	0.668	0.124	0.416
	MELT	**0.748**	**0.184**	**0.724**	**0.149**	**0.451**
Behance	SASRec	0.816	0.320	0.766	0.234	0.534
	ASReP	0.840	0.351	0.789	0.287	0.567
	CITIES	0.828	0.377	0.778	0.298	0.570
	ASReP+CITIES	0.839	0.349	0.792	0.296	0.569
	MELT	**0.854**	**0.458**	**0.813**	**0.371**	**0.624**

1. The consumed item is the test item

EXPERIMENT: ABLATION STUDY

Data	Row	U	I	M	C	Over.	HU	TU	HI	TI	Mean
Music	1					0.534	0.660	0.497	0.851	0.248	0.564
	2	✓				0.541	0.659	0.506	0.891	0.225	0.570
	3		✓			0.578	0.698	0.543	0.848	0.331	0.605
	4	✓	✓			0.582	0.699	0.548	0.877	0.316	0.610
	5	✓	✓	✓		0.597	**0.711**	0.563	0.891	0.330	0.623
	6	✓	✓	✓	✓	**0.600**	0.709	**0.567**	**0.896**	**0.332**	**0.626**
Beauty	1					0.458	0.540	0.452	0.753	0.177	0.480
	2	✓				0.485	0.545	0.472	**0.806**	0.156	0.495
	3		✓			0.471	0.540	0.459	0.724	0.212	0.484
	4	✓	✓			0.490	0.550	0.476	0.777	0.195	0.499
	5	✓	✓	✓		0.498	0.565	0.483	0.775	**0.213**	0.509
	6	✓	✓	✓	✓	**0.502**	**0.566**	**0.487**	0.783	**0.213**	**0.512**

[Ablation Study]

HU/TU: Head user/Tail user
HI/TI: Head item/Tail item

- 1) Effectiveness of User branch (U) and Item Branch (I): 1 row (SASRec [1]) ↔ 2/3 row
- 2) Observe the conflict as User branch and Item branch are combined: 3 row ↔ 4 row on TI performance
- 3) Effectiveness of Mutual Enhancement (Conflict ↓): 4 row ↔ 5 row on TU&TI performance

[1] Self-Attentive Sequential Recommendation. Kang et al. ICDM'18

CONCLUSION

- Based on the empirical discovery, the jointly addressing the long-tailed user and item problems is non-trivial.
  - Naïve combination of existing works cannot effectively enhance the performance of TT group.

- We propose a novel framework named MELT, which consists of bilateral branches trained in a mutually enhancing manner and the curriculum learning is adopted.

- The extensive experiment on 8 datasets demonstrates the effectiveness of alleviating the both long-tailed user and item problems.

Thank you!

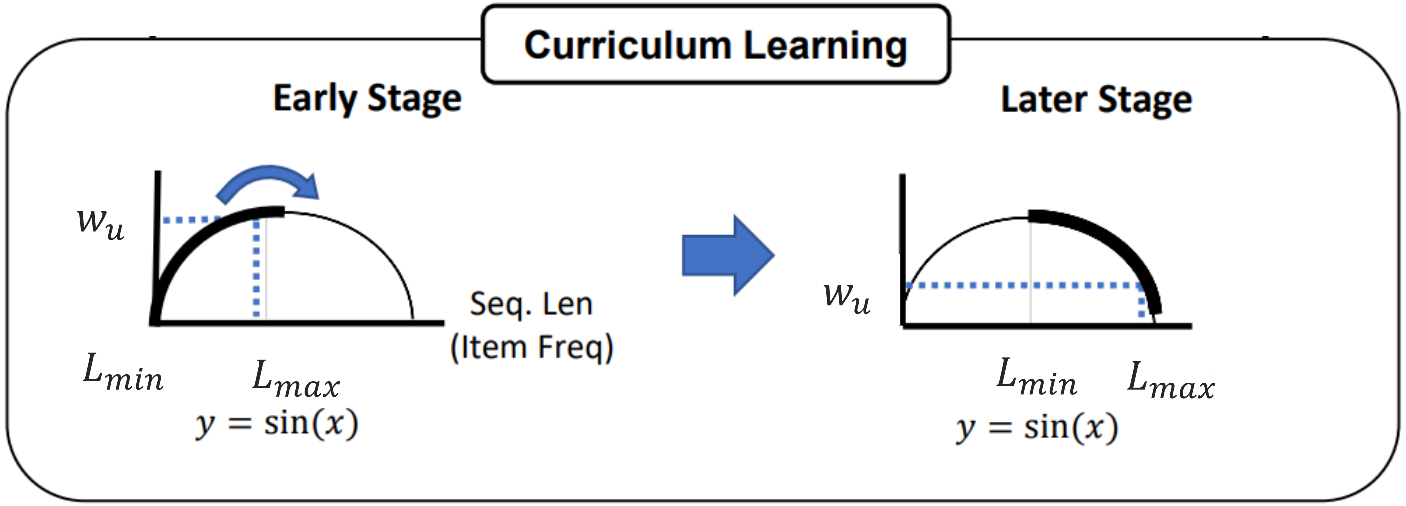
Source Code (Github)

Full Paper

APPENDIX: CURRICULUM METHOD

- **Curriculum learning-based Training**
  - The number of interactions differs among head users.
  - Early stage: mainly learned by large interaction (easy), Later stage: mainly learned by small interaction (hard)
  - User branch의 loss: $w_u \left\|p_u - G_\phi^U(\bar{r}_u)\right\|^2$

$$w_u = \sin\left(\frac{\pi}{2} \cdot \frac{e}{e_{max}} + \frac{\pi}{2} \cdot \frac{|S_u| - L_{min}}{L_{max} - L_{min}}\right)$$



APPENDIX - 1

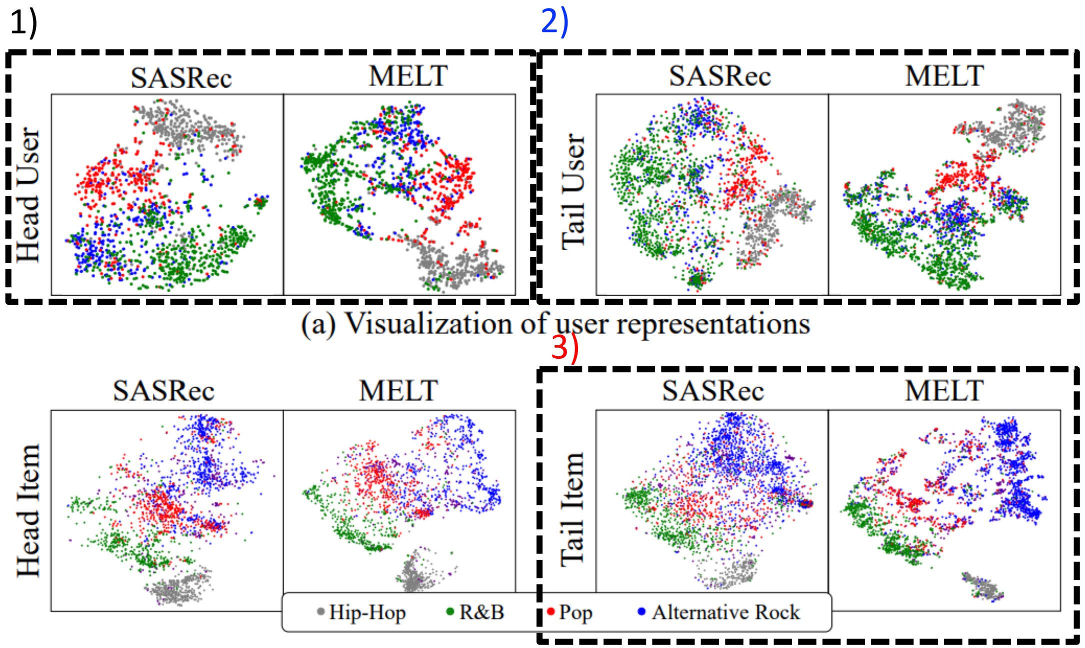
- The performance with the others datasets

Dataset	Model	Overall		Head User		Tail User		Head Item		Tail Item		Mean		Improvement (%)	
		HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	Overall	Mean
Automotive	BERT4Rec	0.3495	0.2093	0.3762	0.2240	0.3433	0.2059	**0.8055**	0.4839	0.0041	0.0013	0.3823	0.2288	17.9	15.7
	Tail-Net	0.3521	0.2118	0.3819	0.2322	0.3453	0.2071	0.7893	0.4818	0.0211	0.0074	0.3844	0.2321	16.9	14.6
	INSERT	0.3516	0.2113	0.3796	0.2296	0.3451	0.2070	0.7886	0.4807	0.0206	0.0072	0.3835	0.2311	17.1	15.0
	ASReP	0.3603	0.2229	0.4084	0.2586	0.3492	0.2147	0.6780	0.4342	0.1197	0.0629	0.3888	0.2426	13.0	11.9
	SASRec	0.3472	0.2154	0.3911	0.2464	0.3371	0.2083	0.6570	0.4229	0.1126	0.0583	0.3745	0.2340	17.1	16.2
	CITIES	0.3505	0.2222	0.3927	0.2518	0.3408	0.2153	0.6348	0.4242	0.1352	**0.0692**	0.3759	0.2401	15.1	14.7
	MELT	**0.4024**	**0.2566**	**0.4397**	**0.2850**	**0.3938**	**0.2501**	0.7284	**0.5098**	**0.1556**	0.0649	**0.4294**	**0.2775**	-	-
Music	BERT4Rec	0.5203	0.3628	0.6443	0.4567	0.4835	0.3349	0.8448	0.6335	0.2270	0.1181	0.5499	0.3858	13.9	12.5
	Tail-Net	0.4791	0.3072	0.5812	0.3746	0.4487	0.2871	0.8729	0.5921	0.1231	0.0496	0.5065	0.3259	27.9	26.4
	INSERT	0.4400	0.2724	0.5077	0.3103	0.4198	0.2611	0.8346	0.5384	0.0832	0.0319	0.4613	0.2854	41.1	40.9
	ASReP	0.5443	0.3830	0.6677	0.4724	0.5076	0.3564	0.8547	0.6341	0.2637	0.1559	0.5734	0.4047	8.4	7.6
	SASRec	0.5344	0.3717	0.6602	0.4643	0.4970	0.3442	0.8512	0.6269	0.2481	0.1410	0.5641	0.3941	11.0	9.8
	CITIES	0.5537	0.3879	0.6850	0.4841	0.5147	0.3593	0.8310	0.6428	0.3030	**0.1574**	0.5834	0.4109	6.8	5.8
	MELT	**0.5997**	**0.4058**	**0.7091**	**0.4846**	**0.5672**	**0.3824**	**0.8961**	**0.6835**	**0.3318**	0.1548	**0.6261**	**0.4263**	-	-
Foursquare	BERT4Rec	0.8939	0.8058	0.9198	0.8406	0.8756	0.7813	0.9445	0.8551	0.1670	0.0975	0.7267	0.6436	2.8	8.6
	Tail-Net	0.8768	0.7828	0.9081	0.8210	0.8548	0.7560	0.9339	0.8358	0.0581	0.0226	0.6887	0.6089	5.3	14.7
	INSERT	0.8490	0.7215	0.8998	0.7784	0.8398	0.7112	0.9326	0.8051	0.2677	0.1400	0.7350	0.6087	11.3	10.7
	ASReP	0.9243	0.8189	0.9444	0.8527	0.9102	0.7952	**0.9656**	0.8616	0.3314	0.2066	0.7879	0.6790	0.3	1.4
	SASRec	0.9193	0.8124	0.9409	0.8474	0.9040	0.7878	0.9624	0.8556	0.2996	0.1918	0.7767	0.6707	0.9	2.8
	CITIES	0.9209	0.8191	0.9427	0.8540	0.9056	0.7944	0.9608	0.8609	0.3474	**0.2180**	0.7891	**0.6818**	0.5	1.1
	MELT	**0.9271**	**0.8210**	**0.9471**	**0.8541**	**0.9131**	**0.7977**	0.9628	**0.8644**	**0.4156**	0.1966	**0.8097**	0.6782	-	-
Behance	BERT4Rec	0.6359	0.4388	0.6490	0.4576	0.6225	0.4197	0.7856	0.5447	0.0328	0.0123	0.5225	0.3586	20.3	32.3
	Tail-Net	0.6039	0.4241	0.6189	0.4450	0.5885	0.4027	0.7452	0.5259	0.0344	0.0137	0.4968	0.3468	25.8	38.2
	INSERT	0.5613	0.3454	0.5542	0.3381	0.5685	0.3529	0.6998	0.4308	0.0032	0.0013	0.4564	0.2808	42.6	58.1
	ASReP	0.7163	0.5177	0.7411	0.5416	0.6910	0.4934	0.8147	0.5964	0.3197	0.2006	0.6416	0.4580	4.8	6.0
	SASRec	0.6891	0.4908	0.7158	0.5146	0.6618	0.4666	0.7912	0.5748	0.2778	0.1526	0.6117	0.4272	9.6	12.2
	CITIES	0.7107	0.5156	0.7370	0.5386	0.6839	0.4921	0.8030	0.5938	0.3386	0.2003	0.6406	0.4562	5.4	6.3
	MELT	**0.7505**	**0.5424**	**0.7736**	**0.5642**	**0.7268**	**0.5201**	**0.8336**	**0.6267**	**0.4154**	**0.2025**	**0.6874**	**0.4784**	-	-

APPENDIX: VISUALIZATION

- Visualization on representation of Head user/item, Tail user/item
  - User: Visualization on user representation ($f_\theta(S_u)$) / Color: Category of the test item
  - Item: Visualization on item embedding (q_i) / Color: Item category

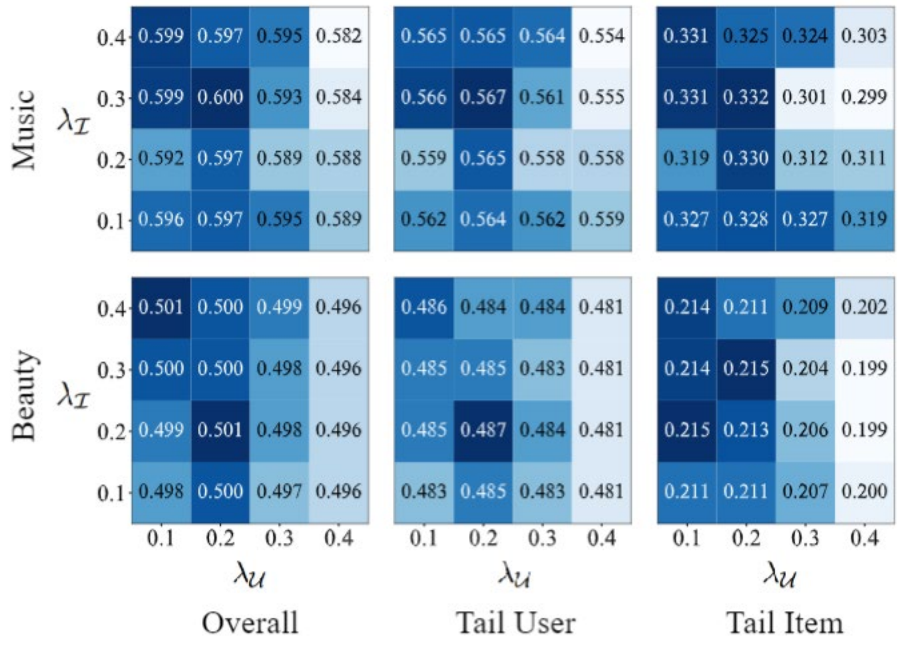
- Closely grouped within Cluster
  - 1) Head user: the benefit of tail item embeddings as user representation relies on the item embeddings
  - 2) Tail user: the effectiveness of the knowledge (G_ϕ^U) in the user branch
  - 3) Tail item: the effectiveness of the knowledge (G_ϕ^I) in the item branch



[Dataset: Amazon Music]

APPENDIX - 2

- Sensitivity Analysis of λ_U and λ_I
  - λ_U: Weight for user branch / λ_I: Weight for item branch
- Best performance over $\lambda_U = 0.2, \lambda_I = 0.3$ on overall, tail user, and tail item groups
  - It simplifies the tuning process
  - The small values for λ_U and λ_I produces the best performance → it acts as regularizers of the recommendation loss



APPENDIX - 3

- Performance over different numbers of layers
  - Default: 1 layer MLP
- Observation
  - MELT generally performs well even with a single-layer feed-forward network
  - Designing the complex generators is not beneficial