Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation

Shengyu Zhang[1*], Lingxiao Yang[1*], Dong Yao[1*], Yujie Lu[6], Fuli Feng[4], Zhou Zhao[1,2†], Tat-seng Chua[5], Fei Wu[2,3†]

[1] Zhejiang University [2] Shanghai Institute for Advanced Study of Zhejiang University
[3] Shanghai AI Laboratory [4] University of Science and Technology of China
[5] National University of Singapore [6] University of California, Santa Barbara, United States
{sy_zhang,yaodongai,zhaozhou,wufei}@zju.edu.cn
{yujielu10,fulifeng93,shawnyang1110}@gmail.com
dcscts@nus.edu.sg

ABSTRACT

Effectively representing users lie at the core of modern recommender systems. Since users' interests naturally exhibit multiple aspects, it is of increasing interest to develop multi-interest frameworks for recommendation, rather than represent each user with an overall embedding. Despite their effectiveness, existing methods solely exploit the encoder (the forward flow) to represent multiple aspects of interests. However, without explicit regularization, the interest embeddings may not be distinct from each other nor semantically reflect representative historical items. Towards this end, we propose the Re4 framework, which leverages the backward flow to reexamine each interest embedding. Specifically, Re4 encapsulates three backward flows, *i.e.*, 1) Re-contrast, which drives each interest embedding to be distinct from other interests using contrastive learning; 2) Re-attend, which ensures the interest-item correlation estimation in the forward flow to be consistent with the criterion used in final recommendation; and 3) Re-construct, which ensures that each interest embedding can semantically reflect the information of representative items that relate to the corresponding interest. We demonstrate the novel forward-backward multi-interest paradigm on ComiRec, and perform extensive experiments on three real-world datasets. Empirical studies validate that Re4 helps to learn learning distinct and effective multi-interest representations.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, Multi-interest, Backward Flow

ACM Reference Format:

Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, Fei Wu. 2022. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512094

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512094

1 INTRODUCTION

A proliferation of the Internet has resulted in an increase in information overload in people's daily lives. Recommender systems help users seek desired information and explore what they are potentially interested in, thus alleviating the information overload. It has become one of the most widely used information systems with applications such as E-commerce, news portals, and micro-video platforms. A successful recommendation framework depends on accurately describing and representing users' interests. The recent advances in neural recommender systems have convincingly demonstrated high capability in learning dense user representation for matching. Matching (also known as deep candidate generation) methods typically represent users and items with dense vectors, and leverage simple similarity functions (*e.g.*, dot product, and cosine similarity) to model user-item interactions. Typically, YoutubeDNN [10] takes user behavior sequence as input, perform mean-pooling on item embeddings in the sequence to obtain user embedding.

Despite their effectiveness, most existing works typically represent each user using an overall embedding. However, in real-world applications, users' interests exhibit multiple aspects. For example, in E-commerce platforms, a user might be simultaneously in favor of sports equipment (*e.g.*, basketball) and electronic products (*e.g.*, desktop). In the embedding hypersphere, an overall user embedding might be less effective in capturing multiple item clusters. As such, devising multi-interest representation frameworks is a promising research direction for capturing users' diverse interests. Multi-interest recommendation is still a nascent research area. Recently, MIND [26] groups users' historical behaviors into multiple clusters based on dynamic capsule routing while each interest capsule reflects a particular aspect. ComiRec [3] introduces self-attention mechanisms to extract multiple interest embeddings and a controllable factor to realize the diversity-performance tradeoff.

However, existing methods only rely on the *forward flow* (items to multi-interest), but do not consider the information from multi-interest to items, named as backward flow. Ignoring the backward flow might raise some limitations on the learned embeddings. For example, a representative attention-based multi-interest framework ComiRec-SA extracts multiple interests using different attention

[*] These authors contributed equally to this work.
[†] Corresponding Authors.

heads. The different heads of attention mechanism mainly introduce randomness into the modeling process but do not necessarily guarantee the outputs of different heads to be distinct from each other. As a result, there is no guarantee on whether the learned multiple embeddings represent multiple aspects of interest. Meanwhile, the attention weights are interpreted as the correlation between historical items and different interests. However, there is no guarantee on whether the correlation is consistent with the recommendation criterion on user-item matching. Note that we use the latter criterion for the final recommendation serving. Therefore, an effective and robust multi-interest recommendation framework requires backward flows (interests-to-items) to re-examine the relationships of learned multi-interest embeddings and historical items.

To address the aforementioned challenges, we propose to leverage the *backward flow* (multi-interests to items) and construct the **Re4** framework for multi-interest recommendation. Re4 consists of three essential components, *i.e.*, *Re-contrast*, *Re-attend*, and *Re-construct*, for effective multi-interest *Re*commendation. 1) **Re-contrast** is devised to learn multiple distinct interests that should capture different aspects of users' interests. Technically, for each interest, we firstly identify the representative items corresponding to the interest based on the similarity function used for deep candidate generation, such as dot product or cosine similarity. Then, we view the representative items as positive items for the interest. There are three kinds of negatives, *i.e.*, items in the user sequence except for the positives, randomly sampled items beyond the sequence, and other interests. Finally, we conduct contrastive learning by pulling the interest closer to the positives and pushing the interest away from the negatives. Obviously, this modeling drives each interest to be distinct from the others. 2) **Re-attend** addresses the concern on the consistency between the attention weights in the forward flow and the recommendation correlation between interests and users. The attention weights are used as the basis for item clustering and interest learning. The correlation is used for final recommendation. Therefore, ensuring consistency between them is essential. Technically, Re-attend explicitly minimizes the distance between the forward attention weights and the interest-item correlation. 3) the above two strategies focus on correlation, *i.e.*, to what extent each interest and each historical item is correlated. As a counterpart, we leverage **Re-attend** to ensure each interest representation can semantically reflect the content of representative items.

We conduct extensive experiments on three public benchmarks and demonstrate the effectiveness of Re4 against state-of-the-art multi-interest frameworks on the matching phase of recommendation. Various analysis including ablation study, hyper-parameter analysis, and case study validate the practical merits of Re4 on learning effective multi-interest representations. In summary, the main contributions of this work are threefold:

- We make an early attempt to incorporate backward flow (interests-to-items) modeling for multi-interest representation learning.
- We propose the Re4 framework and devise three backward flows, *i.e.*, Re-contrast, Re-attend, and Re-construct, to learn distinct multi-interests that can semantically reflect representative items.
- We conduct extensive experiments on three real-world datasets, validating the effectiveness of Re4 on multi-interest recommendation.

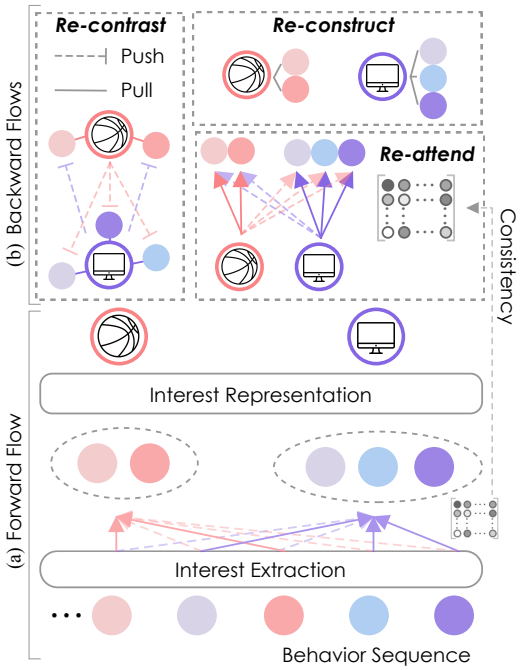


Figure 1: An illustration of leveraging backward flows for multi-interest representation learning. (a) The traditional forward flow that clusters items and extracts multiple interests. (b) The proposed backward flows, *i.e.*, *Re-contrast* which learns distinct multi-interests; *Re-construct* which permits interests' semantic reflection on representative items; and *Re-attend* which ensures the consistency between attention weights in the forward flow and recommendation correlation.

2 METHODS

In this section, we will elaborate on the building blocks of Re4, *i.e.*, the multi-interest extraction module, which is the forward flow, and three backward flow strategies. We use bold letters (*e.g.*, $\mathbf{u}$) to denote vectors, bold upper-case letters (*e.g.*, $\mathbf{W}$) to denote matrices, and letters in calligraphy font (*e.g.*, $\mathbf{P}$) to denote sets.

2.1 Multi-interest Extraction

Given a behavior sequence $X = \{x_i^u\}_{i=1,\ldots,N_x}$ of user u, the multi-interest extraction module aims to extract multiple dense vectors $\mathbf{Z}^u \in \mathbb{R}^{d \times N_z} = \{\mathbf{z}_k^u\}_{k=1,\ldots,N_z}$. x_i^u denotes the ith behavior of user u, and N_x is the length of the behavior sequence. $\mathbf{z}_k^u$ represents the kth interest embedding of user u, and N_z is a hyper-parameter indicating the number of interests. For simplicity, we will drop the superscripts occasionally and use x_i and $\mathbf{z}_k$ in place of x_i^u and $\mathbf{z}_k^u$.

Currently, there are two widely used approaches for this aim, *i.e.*, dynamic capsule routing [26] and attention mechanisms [3]. We

leverage attention mechanisms due to their effectiveness in a broad range of deep learning applications. Specifically, we first transform the behavior sequence into the dense representation using trainable item embedding table, *i.e.*, $\mathbf{X} \in \mathbb{R}^{N_x \times d} = \{\mathbf{x}_i\}_{i=1,\ldots,N_x}$. Then, we employ the additive attention technique to obtain attention weight of the kth interest on the ith item:

$$a_{k,i} = \frac{\exp\left(\mathbf{w}_k^T \tanh\left(\mathbf{W}_1 \mathbf{x}_i\right)\right)}{\sum_j \exp\left(\mathbf{w}_k^T \tanh\left(\mathbf{W}_1 \mathbf{x}_j\right)\right)}, \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d}$ is a transformation matrix shared by all interests, and $\mathbf{w}_k \in \mathbb{R}^{d_h}$ is a interest-specific transformation vector to compute interest-item correlation. $a_{k,i \in \mathbb{R}}$ indicates to what extent item x_i belongs to the interest z_k. The kth interest representation is obtained by:

$$\mathbf{z}_k = \sum_j a_{k,j} \mathbf{W}_2 \mathbf{x}_j. \tag{2}$$

2.2 Backward Flow

The multi-interest extraction module solely models the item-to-interest forward flow. We argue that interest-to-item backward flow can further enhance multi-interest representation learning. Specifically, we devise three backward flows and elaborate their details as the following.

2.2.1 Re-contrast. As shown in Equation 1-2, the attention mechanism extracts multiple interests with interest-specific transformation parameters $\mathbf{w}_k$. However, there is no guarantee that neither $\mathbf{w}_k$ nor the attention weights to be different for different interests. Each $\mathbf{w}_k$ can be interpreted as an attention head, and different attention heads are known to introduce randomness rather than diversity. Therefore, there are chances that the model learns a trivial solution where all interests are close in the embedding space *w.r.t.* items in the behavior sequence. Towards this end, we construct the Re-contrast backward flow, which leverages contrastive learning [36, 58] to learn distinct interest representations. Basically, contrastive learning is performed by pushing the anchor away from the negatives and pulling the anchor closer to the positives. Obviously, the essence of contrastive learning lies in the construction of effective positives and negatives.

Positives. Undoubtedly, the representative items corresponding to the anchor interest can be viewed as positives. In our multi-interest framework, the items in the behavior sequence with high attention weights can be interpreted as representative items. As such, we perform hard selection as the following:

$$\mathcal{P}_k = \{\mathbf{x}_j \mid a_{k,j} > \gamma_c\}, \tag{3}$$

where $\mathcal{P}_k$ denotes the set of positives which includes items with attention weight $a_{k,j}$ higher than a certain threshold γ_c. We empirically set the threshold to the uniform probability $1/N_x$.

Negatives. As a counterpart of the above positives, a straightforward solution is to view other items in the behavior sequence as negatives:

$$\bar{\mathcal{N}}_k = \{\mathbf{x}_j \mid a_{k,j} \leq \gamma_c\}, \tag{4}$$

However, note that our primary goal is to diversify learned interests. The above learning can make each interest representation more discriminative *w.r.t.* items that are not representative, but not necessarily make each interest different from each other. Therefore, we propose to add other interests as negatives. Moreover, we add items that do not appear in the behavior sequence to stabilize contrastive learning. The final negative set can be obtained by:

$$\mathcal{N}_k = \bar{\mathcal{N}}_k \cup (\mathcal{Z} \setminus \{\mathbf{z}_k\}) \cup \tilde{\mathcal{N}}_k, \tag{5}$$

where $\mathcal{Z}$ denotes the set of interest representations, and $\tilde{\mathcal{N}}_k$ is the set of items representations that are sampled beyond the behavior sequence. With the positives $\mathbf{z}^+_{k,i} \in \mathcal{P}_k$ and negatives $\mathbf{z}^-_{k,j} \in \mathcal{N}_k$, we employ InfoNCE [36] as the objective:

$$\mathcal{L}_{CL} = -\sum_i log \frac{\exp\left(\bar{\mathbf{z}}_k \cdot \bar{\mathbf{z}}^+_{k,i}/\tau\right)}{\exp\left(\bar{\mathbf{z}}_k \cdot \bar{\mathbf{z}}^+_{k,i}/\tau\right) + \sum_j \exp\left(\bar{\mathbf{z}}_k \cdot \bar{\mathbf{z}}^-_{k,j}/\tau\right)}. \tag{6}$$

where $(\bar{\mathbf{z}}_k, \bar{\mathbf{z}}^+_{k,i}, \bar{\mathbf{z}}^-_{k,j})$ are L2 normalized vectors of $(\mathbf{z}_k, \mathbf{z}^+_{k,i}, \mathbf{z}^-_{k,j})$, and τ is temperature hyper-parameter [58].

2.2.2 Re-attend. In the multi-interest extraction module, the attention weight $a_{k,i}$ is interpreted as the correlation between the kth interest and the ith item in the behavior sequence. However, such a correlation is not consistent with how matching models make recommendations. Typically, a matching model leverages dot product or cosine similarity to estimate the probability of users clicking on items. As such, to make the correlation computation in the forward flow consistent with the correlation measurement in the final recommendation, we construct the Re-attend backward flow. We first compute the correlation between interests and historical items using the recommendation measurement ϕ:

$$\tilde{a}_{k,i} = \phi(\mathbf{z}_k, \mathbf{x}_i), \tag{7}$$

where ϕ is determined according to the recommender system and we use dot product in experiments. The Re-attend loss function can be written as:

$$\mathcal{L}_{Att} = \sum_k \sum_i L_{CE}(a_{k,i}, \tilde{a}_{k,i}). \tag{8}$$

where L_{CE} denotes the cross-entropy loss function.

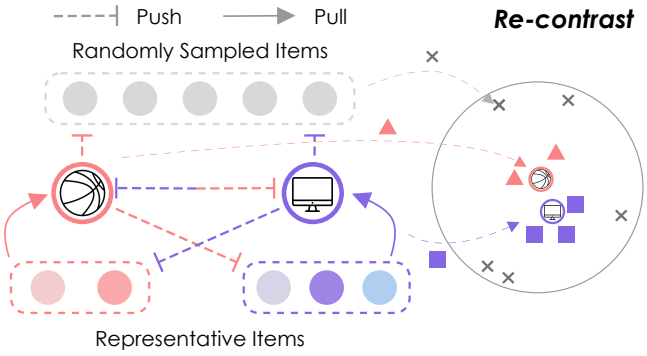


Figure 2: Schema of the Re-contrast backward flow, which aims to learn distinct multi-interest representations.

Table 1: Dataset Statistics.

Dataset	#Users	#Items	#Interactions	#Density
Book	603, 668	367, 982	8, 898, 041	0.00004
MovieLens	6, 040	3, 707	1, 000, 209	0.04467
Gowalla	29, 858	40, 981	1, 027, 370	0.00084

2.2.3 Re-construct. The above two backward flows are concerned with correlations, *i.e.*, to what extent interest-interest and interest-item are correlated. However, they neglect whether interest representations can reflect the content of representative items. To permit such a semantic reflection, we construct the Re-construct backward flow. We leverage self-attention mechanism for reconstruction, which is formulated as:

$$\mathbf{C}_k = \text{Upsample}\left(\mathbf{z}_k\right), \tag{9}$$

$$\beta_{k,i,j} = \frac{\exp\left(\bar{\mathbf{w}}_j^T \tanh\left(\bar{\mathbf{W}}_3 \mathbf{c}_{k,i}\right)\right)}{\sum_m \exp\left(\bar{\mathbf{w}}_j^T \tanh\left(\bar{\mathbf{W}}_3 \mathbf{c}_{k,m}\right)\right)}, \tag{10}$$

$$\hat{\mathbf{x}}_{k,j} = \sum_i \beta_{k,i,j} \bar{\mathbf{W}}_5 \mathbf{c}_{k,i}, \tag{11}$$

where the Upsample function is a linear projection $\bar{\mathbf{W}}_4 \in \mathbb{R}^{N_x d_b \times d}$ followed by a reshape operation to transform the linearly projected vector to a matrix $\mathbf{C}_k \in \mathbb{R}^{N_x \times d_b}$. d_b is the hidden size in Re-construct backward flow. $\mathbf{c}_{k,i}$ is the ith unit in $\mathbf{C}_k$. $\bar{\mathbf{W}}_3 \in \mathbb{R}^{d_b \times d_b}$, $\bar{\mathbf{W}}_5 \in \mathbb{R}^{d \times d_b}$, and $\bar{\mathbf{w}}_j \in \mathbb{R}^{d_b}$ are learnable transformations, and each input item x_j has a corresponding $\bar{\mathbf{w}}_j$. Different from auto-encoders [23, 30, 42] which reconstruct all inputs, we propose to reconstruct representative items corresponding to the interest. Specifically, we take the positive set $\mathcal{P}_k$ constructed by Equation 3 as the representative items for the kth interest. Therefore, the loss function of the Re-construct backward flow can be written as:

$$\mathcal{L}_{CT} = \sum_k \sum_j \mathbb{1}\left(\mathbf{x}_j \in \mathbb{P}_k\right) \left\|\hat{\mathbf{x}}_{k,j} - \mathbf{x}_j\right\|_F^2. \tag{12}$$

where $\mathbb{1}$ is an indicator function which returns 1 when the condition is true and returns 0 otherwise. We empirically find that semantic reflection can help the interest presentation to have a fine-grained understanding of items, and thus leads to boost recommendation metrics that consider rank positions.

2.3 Training and Inference

We follow the training and inference paradigm of ComiRec [3]. At the training stage, given multiple interest representations $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1,\ldots,N_z}$ and the target item embedding $\mathbf{y}$, we obtain the interest embedding that is the most related to the target item through:

$$\hat{\mathbf{z}} = \mathbf{Z}\left[:, \text{argmax}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{y}\right)\right], \tag{13}$$

Then, we adopt the negative log-likelihood objective:

$$\mathcal{L}_{Rec} = -\log p_\theta(y \mid X), \tag{14}$$

$$\text{where } p_\theta(y \mid X) = \frac{\exp\left(\hat{\mathbf{z}}^T \mathbf{y}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\hat{\mathbf{z}}^T \mathbf{y}'\right)}, \tag{15}$$

where y' denotes a randomly sampled item. During the matching phase of recommendation, it can be impractical to sum over the entire item gallery $\mathcal{Y}$ as in the denominator. We adopt the sample softmax objective [1, 22]. Therefore, the final training loss function is:

$$\mathcal{L}_{Re4} = \mathcal{L}_{Rec} + \lambda_{CL}\mathcal{CL} + \lambda_{Att}\mathcal{L}_{Att} + \lambda_{CT}\mathcal{L}_{CT}. \tag{16}$$

At inference, the recommendation probability of item y for user u with multiple interests $\mathbf{Z}^u = \{\mathbf{z}_k^u\}_{k=1,\ldots,N_z}$ is:

$$p_{u,y} = \max\left\{\mathbf{y}^T \mathbf{z}_k^u\right\}_{k=1,\ldots,N_z} \tag{17}$$

Finally, the top-N items are obtained with $p_{u,y}$ as the basis.

3 RELATED WORKS

Neural Recommender Systems. Neural recommendation models incorporate neural networks for user-item interaction modeling or user/item representation learning. Neural networks are graceful to capture the non-linear feature interactions between users and items [12, 15, 20, 50, 51, 55, 61, 62, 75], and can hereby boost traditional collaborative filtering methods. Typically, neural collaborative filtering (NCF) [17] leverages both a generic matrix factorization component and a non-linear MLP for interaction modeling to jointly enhance recommendation. As for user/item representation learning, there are roughly two lines of works, *i.e.*, graph-based modeling [2, 16, 44, 52, 53, 79], sequence-based modeling [32–34, 39, 47, 48, 60, 66, 70]. Sequence-based recommendation models have the advantage of modeling dynamic user interest by extracting user representation from the newest behavior sequence. Youtube-DNN [10] is one of the earliest works on sequential recommendation which leverages mean-pooling to obtain users' representation. Inspired by sequence modeling in the generic domain (*e.g.*, natural language processing and video processing [7, 63, 67–69, 71–73, 76–78]), this work is followed by a lot of advanced techniques such as Recurrent Neural Networks [11, 18, 19, 27, 38, 65], attention mechanisms [3, 43, 49, 64], dynamic capsule routing [3, 26], and memory networks [6, 21].

Multi-interest Recommendation. To capture diverse interests of users, there is increasing attention on multi-interest representation learning [3–5, 26, 57, 59], related to disentangled representation in the generic domain [28, 54]. MIND [26] takes the initiative to represent users with multiple interests. They devise the multi-interest extractor based on dynamic capsule routing [40]. ComiRec [3] is a state-of-the-art that leverages self-attention for multi-interest modeling. They also introduce a controllable factor for recommendation diversity-accuracy tradeoffs. We follow these works to devise generic multi-interest modeling for the matching phase, and propose to model the backward flow (interests-to-items). [4] utilizes auxiliary time information to better extract multiple interests. [5] construct pre-defined four kinds of representation (user-level, item-level, neighbor-assisted, and category-level) for video recommendation. [59] focuses on the ranking phase of recommendation, and devises a target-item-aware multi-interest extraction layer.

Backward Flow in Recommendation. Generally, leveraging backward flow (output-to-input) is not a new paradigm for deep learning. Auto-encoders [23, 30, 41, 42, 56] and dual learning

Table 2: A comparison between the proposed Re4 framework and state-of-the-art matching baselines on three public benchmarks. Re4 mostly achieves performance gains over baselines, and the performance improvement is more substantial on the larger-scale dataset with a large item gallery (Amazon) where users' interests are more likely to be diverse.

Datasets	Metric	POP	Y-DNN	GRU4Rec	MIND	ComiRec-SA	ComiRec-DR	Re4	Improv.
Amazon	R@20	0.0137	0.0457	0.0406	0.0486	0.0549	0.0531	**0.0771**	40.44%
	R@50	0.0240	0.0731	0.0650	0.0764	0.0847	0.0811	**0.1155**	36.36%
	NDCG@20	0.0226	0.0767	0.0680	0.0793	0.0899	0.0918	**0.1304**	42.05%
	NDCG@50	0.0394	0.1208	0.1037	0.1223	0.1356	0.1352	**0.1883**	38.86%
	HR@20	0.0302	0.1029	0.0894	0.1062	0.1140	0.1201	**0.1627**	35.47%
	HR@50	0.0523	0.1589	0.1370	0.1610	0.1720	0.1758	**0.2326**	32.31%
MovieLens	R@20	0.0006	0.1115	**0.1286**	0.1033	0.1189	0.1223	0.1117	-13.14%
	R@50	0.0016	0.2191	**0.2428**	0.1994	0.1949	0.2263	0.2048	-15.65%
	NDCG@20	0.0057	0.3671	0.3971	0.3325	0.3131	0.3913	**0.4581**	15.36%
	NDCG@50	0.0135	0.4035	0.4157	0.3683	0.3396	0.4039	**0.6067**	45.95%
	HR@20	0.0186	0.7318	0.7831	0.7020	0.7550	0.7714	**0.8048**	2.77%
	HR@50	0.0452	0.8858	0.8990	0.8593	0.8874	0.8801	**0.9288**	3.31%
Gowalla	R@20	0.0028	0.1127	0.1273	0.1218	0.1277	0.1153	**0.1386**	8.54%
	R@50	0.0054	0.1926	0.2043	0.2049	0.2072	0.1831	**0.2203**	6.32%
	NDCG@20	0.0073	0.2378	0.2803	0.2565	0.2736	0.2534	**0.3141**	12.06%
	NDCG@50	0.0135	0.3638	0.4002	0.3888	0.4019	0.3621	**0.4412**	9.78%
	HR@20	0.0104	0.3443	0.3814	0.3627	0.3838	0.3429	**0.4206**	9.59%
	HR@50	0.0224	0.5010	0.5251	0.5301	0.5288	0.5355	**0.5697**	6.39%

[13, 25, 45, 74, 80] share similar ideas. [23] removes unnecessary connections between neurons in existing fully-connected auto-encoders and enhances recommendation based on the optimized encoder structures. [45] views preference prediction and review generation as two dual tasks, and interchangeably predicts each one using the other along with user and item id information. To the best of our knowledge, there is no prior attempt to explore the backward flow in multi-interest recommendation. Moreover, we propose several practical strategies that boost multi-interest representation learning in semantics and in correlation with corresponding representative items.

4 EXPERIMENTS

4.1 Experimental Setup

We conduct experiments on real-world datasets to answer three main research questions:

- **RQ1**: How does Re4 perform compared to state-of-the-art models?
- **RQ2**: How do the different re-examination strategies and the number of interests affect Re4?
- **RQ3**: How do the learned multi-interest representations benefit from the backward flow?

Datasets We consider three real-world recommendation benchmarks, of which the statistics are shown in Table 1.

- **Amazon**[*]. Amazon Review dataset [35] is one of the most widely used recommendation benchmarks. We choose the largest category Amazon Book for evaluation. We keep the last 20 behaviors to construct the behavior sequence for each user.
- **MovieLens**[†]. MovieLens-1M is a widely used public benchmark on movie ratings.
- **Gowalla.** We use the 10-core setting [14] of the check-in dataset [29] released by the Gowalla platform.

Evaluation Protocal & Metrics. For a fair comparison, we employ the evaluation framework of ComiRec [3], which can demonstrate the generalization capability of all models by assessing them according to unseen user behavior sequences. Details can be found in [3]. The ***matching*** phase of recommendation is our primary focus, and we select the datasets, comparison methods, and evaluation metrics accordingly. For the matching phase, *Recall*, *Normalized Discounted Cumulative Gain* (NDCG)[‡], and *Hit Rate* are three broadly used metrics. Metrics are computed based on the top 20/50 recommended candidates (*e.g.*, Recall@20). Higher scores demonstrate better recommendation performance for all metrics.

Baselines We follow ComiRec [3] to construct matching baselines and also take their method for comparison. Since the evaluation protocol requires modeling unseen users and unseen behavior sequences, we do not consider factorization-based and graph-based methods.

- **POP.** POP always recommends the most popular items to users.

[*]http://jmcauley.ucsd.edu/data/amazon/

[†]https://grouplens.org/datasets/movielens/1m/

[‡]For a fair comparison with the state-of-the-art baseline ComiRec, we strictly follow their metric implementation. The normalization term IDCG used in their paper is calculated *w.r.t.* recalled positive items rather than all positive items.

- **YouTube DNN[10].** Y-DNN is a successful industrial recommender that takes the behavior sequence as input, and pools the embedded items to have user representation.
- **GRU4Rec [19].** GRU4Rec is a representative model that uses the Gated Recurrent Unit [9] to model the sequential dependencies between items.
- **MIND [26].** MIND is one of the earliest multi-interest frameworks, and employs dynamic capsule routing to extract multiple interest embeddings.
- **ComiRec-DR [3].** ComiRec-DR is the state-of-the-art multi-interest framework that also uses dynamic routing and introduces a controllable factor for recommendation diversity-accuracy tradeoffs.
- **ComiRec-SA [3].** ComiRec-SA uses self-attention mechanisms for multi-interest modeling, which is the base model of Re4.

Implementation Details We use Adam [24] for optimization with learning rate of 0.003/0.005 for Books/Yelp and Gowalla, $\beta_1 = 0.9$, $\beta_2 = 0.99, \epsilon = 1{\times}10^{-8}$, weight decay of $1{\times}1e{-}5$. All models are with embedding size $d = 64$. The hidden size in the forward flow is set to $d_h = 256$, and the hidden size in the backward flow is set to $d_b = 32$. The temperature in Re-contrast is set to $\tau = 0.02$. The default interest number in experiments is set to $N_z = 8$. We search for loss coefficients λ_{Att}, λ_{Att}, and λ_{Att} in the range $\{0.01, 0.1, 1, 10\}$. Re4 is implemented with Pytorch [37] 1.6.0 and Python 3.8.5.

4.2 Overall Performance (RQ1)

Table 2 shows the recommendation results on three datasets. We have the following observations:

- Always recommending popular items (POP) without considering users' interests is less effective on three datasets. These results show that neural models cannot easily achieve better results by fitting trivial findings (*e.g.*, recommending items that have more interactions).
- Traditional sequential recommenders (Y-DNN and GRU4Rec) substantially outperform POP. Sequential recommenders have the advantage of taking users' latest behavior sequence into consideration, and can hereby make dynamic recommendations for users with interactions that are unseen during training. GRU4Rec consistently outperforms Y-DNN, demonstrating the superiority of modeling sequential dependencies between items. Note that Y-DNN solely mean-pools historical items. However, both Y-DNN and GRU4Rec represent users with an overall user embedding, which might easily lead to suboptimal results when items are various, and users' interests are diverse.
- Multi-interest frameworks (*e.g.*, MIND, ComiRec-SA) generally achieve better performance than baselines that uses single embedding to represent users on large-scale datasets (*i.e.*, Amazon, and Gowalla). Such improvement basically indicates that when items are diversified and users' interests are hereby more likely to be diverse, multi-interest framework is a more effective way to represent users. Not surprisingly, on MovieLens-1M with limited items (3, 900), multi-interest baselines cannot beat traditional recommenders. Among multi-interest baselines, ComiRec-SA with self-attention mechanisms achieves the best performance results on three datasets. It makes sense that attention mechanisms have been demonstrated as effective in numerous deep learning tasks [8, 31, 46]. The different attention heads in ComiRec-SA introduce randomness and are interpreted as interest encoders for different interest aspects. The attention weights of each interest on historical items are hereby interpreted as the correlation of items and interests.
- Re4 consistently yields the best performance on three datasets in most cases. Remarkably, Re4 improves the best performing baselines by 42.05%, 15.36%, and 12.06% in terms of NDCG@20 on Amazon Book, MovieLens, and Gowalla, respectively. By leveraging the backward flows, the learned multi-interests become more distinct from each other, and better correlated with the corresponding representative items. Interestingly, different from other multi-interest frameworks that cannot beat GRU4Rec, Re4 improves GRU4Rec *w.r.t.* NDCG and Hit Rate. On MovieLens with limited items, the larger improvement *w.r.t.* NDCG probably indicates that the backward flow helps to learn fine-grained multi-interests (*e.g.*, basketball, and football), which are harder to learn than coarse-grained interests (*e.g.*, sports equipment, and electronic products). With fine-grained multi-interests, Re4 places positive test items in front of the others in the ranking list with confidence, and thus achieving higher NDCG. Nevertheless, Re4 obtain larger performance gains on larger datasets Amazon, demonstrating the practical merits of Re4.

Table 3: Analysis of the number of interests N_z, which is a hyper-parameter defined in Section 2.1.

Model	Metric	$N_z = 2$	$N_z = 4$	$N_z = 6$	$N_z = 8$
Amazon	R@20	0.0728	0.0745	0.0769	**0.0771**
	R@50	0.1033	0.1105	**0.1156**	0.1155
	N@20	0.1239	0.1277	0.1298	**0.1304**
	N@50	0.1704	0.1812	0.1879	**0.1883**
	H@20	0.1494	0.1575	0.1606	**0.1627**
	H@50	0.2063	0.2220	0.2311	**0.2326**
MovieLens	R@20	0.1007	0.1121	**0.1128**	0.1117
	R@50	0.1831	**0.2083**	0.2060	0.2048
	N@20	0.4335	0.4615	**0.4648**	0.4581
	N@50	0.5761	**0.6177**	0.6070	0.6067
	H@20	0.7748	0.7765	0.7980	**0.8048**
	H@50	0.9040	0.9189	0.9238	**0.9288**

4.3 In-depth Analysis (RQ2)

4.3.1 Analysis of the number of user interest embeddings. To have a comprehensive analysis of how the number of user interest embeddings affects the performance of Re4, we conduct experiments on a large-scale dataset Amazon with diverse items (*i.e.*, 313, 966) and a small-scale dataset MovieLens with limited items (*i.e.*, 3, 900). According to the results shown in Table 3, we have several findings:

Table 4: Ablation studies by progressively adding proposed backward flows to the base model.

Model	R@20	N@20	H@20	R@50	N@50	H@50
Base	0.128	0.274	0.384	0.207	0.402	0.529
$+\mathcal{L}_{Att}$	0.133	0.287	0.400	0.220	0.427	0.573
$+\mathcal{L}_{CL}$	0.135	0.296	0.412	**0.222**	0.433	**0.575**
$+\mathcal{L}_{CT}$	**0.139**	**0.314**	**0.421**	0.220	**0.441**	0.570

Table 5: Analysis of the positive/negative selection threshold γ_c in Re-contrast, as defined in Equation 3.

γ_c	R@20	R@50	N@20	N@50	H@20	H@50
1/2	0.126	0.209	0.272	0.408	0.383	0.547
1/4	0.128	0.221	0.277	0.425	0.387	0.567
1/16	0.134	0.222	0.291	0.433	0.406	0.582
1/32	0.126	0.208	0.276	0.407	0.387	0.549
Ada	0.139	0.220	0.314	0.441	0.421	0.570

- Increasing the amount of interest embeddings mostly leads to performance gains (*e.g.*, $2 \rightarrow 4 \rightarrow 6 \rightarrow 8$ on Amazon and $2 \rightarrow 4, 6, 8$ on MovieLens). When increasing the amount of interest embeddings, a multi-interest framework probably yields trivial multi-interest embeddings that are similar to each other, or incorrect interest embeddings that are semantically irrelevant to their corresponding representative items. Such ways of modeling hinder performance gains. As such, the performance gain basically demonstrates that Re4 can help to learn effective multi-interest embeddings through the backward flow, *i.e.*, Re-contrast, Re-construct, and Re-attend.
- When increasing the number of interests, Re4 consistently improves the performance on the Amazon dataset while there are minor performance drops *w.r.t.* several metrics on the MovieLens dataset. This result generally indicates the practical merits of Re4 for large-scale recommender systems where the information overload problem is more severe. This finding is consistent with the results shown in Table 2 and analyzed in Section 4.2.
- Although there is a performance drop on small-scale dataset MovieLens when the number of interest increases, *i.e.*, $4 \rightarrow 6, 6 \rightarrow 8$, the change is relatively lower than the performance gain obtained in $2 \rightarrow 4$. Jointly analyzing the results of larger-scale dataset Amazon, the performance of Re4 is generally less sensitive to over-increasing the hyper-parameter, *i.e.*, the number of interests. This observation further demonstrates the practical merits of Re4.

4.3.2 Ablation Studies. To investigate how different backward flows (*i.e.*, Re-contrast, Re-construct, and Re-attend) affect the performance of Re4, we conduct ablation studies by progressively adding three strategies to the base model. The results on the Gowalla dataset are shown in Table 5.

Table 6: Analysis of how loss coefficients λ_{Att}, λ_{Att}, and λ_{Att} affect the performance of Re4. For each experiment, we add one loss function to the base model and alter the corresponding coefficient.

Model	Coeff.	H@20	R@20	N@20
Base	NA	0.3838	0.1277	0.2736
w. $\mathcal{L}_{Att}$	1	0.3868	0.1287	0.2759
	0.1	0.4042	0.1302	0.2843
	0.01	0.4002	0.1334	0.2875
w. $\mathcal{L}_{CL}$	1	0.4039	0.1328	0.2846
	0.1	0.4049	0.1335	0.2874
	0.01	0.4056	0.1351	0.2905
w. $\mathcal{L}_{CT}$	1	0.4042	0.1341	0.2861
	0.1	0.4066	0.1325	0.2883
	0.01	0.3995	0.1331	0.2862

- $\mathcal{L}_{Att}$ denotes the loss function of the Re-attend backward flow, which explicitly guarantees that the dot product similarities between interests and items are consistent with the attention weights in the forward flow. The performance gain demonstrates the importance of such consistency. We attribute the improvement to that many matching models, including the base model ComiRec-SA, make recommendations based on the dot product similarities between interest embeddings and item embeddings.
- $\mathcal{L}_{CL}$ refers to the Re-contrast backward flow, which drives interest embeddings to be distinct from each other. Adding $\mathcal{L}_{CL}$ leads to consistent recommendation performance improvement, validating the necessity of the Re-contrast backward flow for the multi-interest framework.
- $\mathcal{L}_{CT}$ is designed to ensure that each interest embedding can semantically reflect the corresponding representative items. Adding $\mathcal{L}_{CT}$ leads to more gains *w.r.t.* NDCG. The reason might be that semantic reflection can help the interest embeddings to have a fine-grained understanding of the representative items rather than the coarse-grained understanding such as the correlation. Fine-grained understanding is further graceful for distinguishing candidates at the top ranks, and gives accurate items higher ranks. Another evidence for the analysis is that the improvement on Top 20 candidates are higher than Top 50 candidates. This merit is essential since, in many real-world recommender systems, users generally give fewer chances for items that are left behind.

4.3.3 Analysis of Loss Coefficients. Since Re4 introduces new loss functions, we take an analysis of how loss coefficients affect the performance of the corresponding backward flow. Specifically, for each experiment, we solely add one loss function among $\mathcal{L}_{Att}, \mathcal{L}_{CL}, \mathcal{L}_{CT}$ to the base model, and manually set its coefficient to $1, 0.1, 0.001$. The results on Gowalla dataset are shown in Table 6. When altering the loss coefficients, there are minor performance changes in most cases. In other words, the recommendation

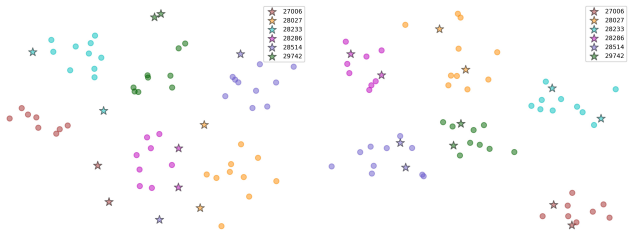
performance is insensitive to the hyper-parameters. An exception is the $\mathcal{L}_{Att}$ loss function, which leads to less effective performance when the corresponding coefficient is high $\lambda_{Att} = 1$. The reason might be that over-regularization on the attention weights probably limits the expressive power of attention mechanism in the forward flow. Still, all variants of Re4 perform better than the base model, demonstrating the merits of three backward flows.

4.3.4 Analysis of γ_c in Re-contrast. In positive/negative items selection of Re-contrast, we introduce a hyperparameter γ_c as the threshold, as defined in Equation 3. In practice, we choose an adaptive $\gamma_c = 1/N_x$ to somehow balance the number of positives and negatives. To have an analysis of the how γ_c affects the performance of Re4, we conduct experiments with $\gamma_c = \{1/2, 1/4, 1/16, 1/32\}$. Note that large $\gamma_c = 1/2$ and small $\gamma_c = 1/32$ correspond skewed cases with significantly more negatives and positives, respectively. $\gamma_c = 1/16$ yields a relatively proper ratio of positives to negatives, leading to better performance than the skewed cases. The adopted adaptive strategy $\gamma_c = 1/N_x$ achieves better performance than the fixed value $\gamma_c = 1/16$ in most cases.

4.4 Effect on Representation (RQ3)

Case Study. We are interested in how the proposed backward flows facilitate multi-interest representation learning in the embedding space. As such, we train the two-interest version of both the base model (ComiRec-SA) and Re4. We randomly sample six users and their corresponding items, and obtain users' multi-interest embeddings and item embeddings. We perform t-SNE transformation onto these embeddings and plot the results in Figure 3a and Figure 3b for Base and Re4, respectively. Both users and items are randomly sampled from the test set and are unseen during training, which helps to better reveal the generalization ability. We have the following findings:

- Overall, users with their corresponding items exhibit more noticeable clusters in Re4. The base model is more likely to yield trivial multi-interest embeddings. For instance, the two interest embeddings of user 29742 present no significant difference *w.r.t.* the distance to their corresponding items. As for user 28514, while one interest embedding (top) is close to its corresponding items, another interest embedding (bottom) is far away from the items. These inferior results probably verify the analysis in Section 4.3.1. As a remedy, the proposed Re-contrast backward flow can drive multiple interest embedding to be discrepant while the Re-attend and Re-construct backward flows can ensure interest embeddings are closed to their representative items both in correlation and in semantics.
- For Re4, two interest embeddings with their closest items mostly exhibit two fine-grained clusters. For example, for user 28027, 28286, and 28233, the two interests' representations are not only distinct from each other in the embedding space, but also with exclusive test items around. Meanwhile, each user's multiple interests exhibit a larger distance with other users' items and interests than the distance with items and interests of the same user. These results jointly demonstrate that Re4 learns effective multiple embeddings that can represent different aspects of interests.



(a) Base **(b) Re4**

Figure 3: The visualization displays the multi-interests (⋆) of some randomly sampled test users, and some corresponding items (• of the same color). We perform t-SNE transformation on the multi-interest embeddings and item embeddings learned by the base model without backward flow and Re4.

Table 7: Quantitative analysis of representations learned by the Base model and Re4.

CM	CI Metric	K-means++ Base	K-means++ SSRec	User Interests Base	User Interests SSRec
K-means	INTER	20.32	23.03	26.91	32.98
	INTRA	35.10	37.14	35.80	38.65
FCM	INTER	84.50	88.89	74.58	84.72
	INTRA	36.47	38.14	37.84	39.48
X-means	INTER	26.29	31.22	29.18	35.15
	INTRA	36.07	37.78	37.47	39.05

Quantitative Result. Besides the above qualitative results, we also provide quantitative results as follows: 1) We perform clustering on the representations of all users' interests and items. We evaluate the ratio of positive items being in the cluster of their corresponding interest, *i.e.*, **INTER** (-User). 2) For each user, we perform clustering on the representation of his/her interest and items representations. We evaluate the ratio of his/her interests being in different clusters, *i.e.*, **INTRA** (-User). We use multiple cluster initialization (CI) methods and multiple clustering methods (CM). We use the negative of recsys's similarity function (dot product) as the distance metric. According to Table 7, we have similar observations as in the case study: 1) Users and their corresponding items exhibit more noticeable clusters in Re4 (with high INTER); and 2) different interest embeddings with their closest items are more likely to exhibit fine-grained clusters in Re4 (with high INTRA).

5 CONCLUSION AND FUTURE WORK

In this paper, we investigate how we can model and leverage the backward flow (interests-to-items) for multi-interest recommendation. We devise the Re4 framework that incorporates three backward flows, *i.e.*, Re-contrast, Re-construct, and Re-attend. In essence,

Re4 facilitates the multi-interest representation learning to 1) capture diverse aspects of interest; 2) semantically reflect the corresponding representative items; and 3) make the attention weights in the forward flow consistent with interest-item correlation *w.r.t.* the final recommendation. We conduct extensive experiments on three real-world datasets, providing insightful analyses on the rationality and effectiveness of Re4. This work was an initiative to construct the forward-backward paradigm for multi-interest recommendation. Remarkably, the backward flow does not affect inference, which is essential for industrial recommender systems. We believe that the novel paradigm can be inspirational to future developments. Obviously, there is more to explore on backward flow strategies. For example, we can extend Re4 to content-based recommenders and explore strategies for various auxiliary features.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (No. 62037001,61836002,62072397), National Key R&D Program of China (No. 2020YFC0832500), Zhejiang Natural Science Foundation (No. LR19F020006), and Project by Shanghai AI Laboratory (No. P22KS00111).

REFERENCES

[1] Yoshua Bengio and Jean-Sébastien Senecal. 2008. Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model. *IEEE Trans. Neural Networks* (2008).

[2] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *CoRR* (2017).

[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation.. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.*

[4] Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. 2021. Exploring Periodicity and Interactivity in Multi-Interest Framework for Sequential Recommendation.. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021.*

[5] Xusong Chen, Dong Liu, Zhiwei Xiong, and Zheng-Jun Zha. 2021. Learning and Fusing Multiple User Interest Representations for Micro-Video and Movie Recommendations. *IEEE Trans. Multim.* (2021).

[6] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks.. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.*

[7] Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In *Proc. of WWW.*

[8] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory-Networks for Machine Reading.. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016.*

[9] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.*

[10] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations.. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016.*

[11] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations.. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017.*

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual Learning for Machine Translation.. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain.*

[14] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback.. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence.*

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval.* 355–364.

[16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval.*

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering.. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017.*

[18] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations.. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018.*

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks.. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.*

[20] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web.* 193–201.

[21] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks.. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[22] Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation.. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers.*

[23] Farhan Khawar, Leonard K. M. Poon, and Nevin L. Zhang. 2020. Learning the Structure of Auto-Encoding Recommenders.. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.*

[24] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization.. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[25] Jae-woong Lee, Seongmin Park, and Jongwuk Lee. 2021. Dual Unbiased Recommender Learning for Implicit Feedback.. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021.*

[26] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall.. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.*

[27] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation.. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017.*

[28] Mengze Li, Kun Kuang, Qiang Zhu, Xiaohong Chen, Qing Guo, and Fei Wu. 2020. IB-M: A Flexible Framework to Align an Interpretable Model and a Black-box Model. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, 643–649.

[29] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation.. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016.*

[30] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering.. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018.*

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks.. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*

[32] Yujie Lu, Shengyu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. 2021. Future-Aware Diverse Trends Framework for Recommendation.. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021.*

[33] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. Pi-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations.. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[34] Jarana Manotumruksa and Emine Yilmaz. 2020. Sequential-based Adversarial Optimisation for Personalised Top-N Item Recommendation.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*.

[35] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes.. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*.

[36] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* (2018).

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). 8024–8035.

[38] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks.. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*.

[39] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential Recommendation with Self-Attentive Multi-Adversarial Network.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*.

[40] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules.. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

[41] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential Variational Autoencoders for Collaborative Filtering.. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*.

[42] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering.. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*.

[43] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer.. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*.

[44] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, and et al. 2020. A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks.. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*.

[45] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation.. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[47] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and Time-aware Item Modeling for Sequential Recommendation.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*.

[48] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation.. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[49] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-Based Transactional Context Embedding for Next-Item Recommendation.. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

[50] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 1717–1725.

[51] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1288–1297.

[52] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering.. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[53] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*.

[54] Anpeng Wu, Kun Kuang, Junkun Yuan, Bo Li, Pan Zhou, Jianrong Tao, Qiang Zhu, Yueting Zhuang, and Fei Wu. 2020. Learning decomposed representation for counterfactual inference. *arXiv preprint arXiv:2006.07040* (2020).

[55] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*. 153–162.

[56] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems.. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*.

[57] Yongji Wu, Lu Yin, Defu Lian, Mingyang Yin, Neil Zhenqiang Gong, Jingren Zhou, and Hongxia Yang. 2021. Rethinking Lifelong Sequential Recommendation with Incremental Multi-Interest Attention. *CoRR* (2021).

[58] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination.. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

[59] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Multi-Interest Network for Click-through Rate Prediction.. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*.

[60] Jiahao Xun, Shengyu Zhang, Zhou Zhao, Jieming Zhu, Qi Zhang, Jingjie Li, Xiuqiang He, Xiaofei He, Tat-Seng Chua, and Fei Wu. 2021. Why Do We Click: Visual Impression-aware News Recommendation.. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*.

[61] Jiangchao Yao, Feng Wang, Kunyang Jia, Bo Han, Jingren Zhou, and Hongxia Yang. 2021. Device-Cloud Collaborative Learning for Recommendation.. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*.

[62] Jiangchao Yao, Shengyu Zhang, Yang Yao, Feng Wang, Jianxin Ma, Jianwei Zhang, Yunfei Chu, Luo Ji, Kunyang Jia, Tao Shen, and et al. 2021. Edge-Cloud Polarization and Collaboration: A Comprehensive Survey. *CoRR* (2021).

[63] Hongbin Ye, Ningyu Zhang, Shumin Deng, Hui Chen Xiang Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced Prompt-tuning for Few-shot Learning. In *Proc. of WWW*.

[64] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks.. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*.

[65] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation.. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*.

[66] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*.

[67] Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. 2020. DeVLBert: Learning Deconfounded Visio-Linguistic Representations.. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*.

[68] Shengyu Zhang, Ziqi Tan, Jin Yu, Zhou Zhao, Kun Kuang, Jie Liu, Jingren Zhou, Hongxia Yang, and Fei Wu. 2020. Poet: Product-oriented Video Captioner for E-commerce.. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*.

[69] Shengyu Zhang, Ziqi Tan, Zhou Zhao, Jin Yu, Kun Kuang, Tan Jiang, Jingren Zhou, Hongxia Yang, and Fei Wu. 2020. Comprehensive Information Integration Modeling Framework for Video Titling.. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*.

[70] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation.. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*.

[71] Wenqiao Zhang, Haochen Shi, Jiannan Guo, Shengyu Zhang, Qingpeng Cai, Juncheng Li, Sihui Luo, and Yueting Zhuang. 2021. MAGIC: Multimodal relAtional Graph adversarIal inferenCe for Diverse and Unpaired Text-based Image

Captioning. *arXiv preprint arXiv:2112.06558* (2021).
[72] Wenqiao Zhang, Haochen Shi, Siliang Tang, Jun Xiao, Qiang Yu, and Yueting Zhuang. 2021. Consensus graph representation learning for better grounded image captioning. In *Proc 35 AAAI Conf on Artificial Intelligence*.
[73] Wenqiao Zhang, Xin Eric Wang, Siliang Tang, Haizhou Shi, Haochen Shi, Jun Xiao, Yueting Zhuang, and William Yang Wang. 2020. Relational graph learning for grounded video description generation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3807–3828.
[74] Yin Zhang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, and Ed H. Chi. 2021. A Model of Two Tales: Dual Transfer Learning Framework for Improved Long-tail Item Recommendation.. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*.
[75] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2016. User Preference Learning for Online Social Recommendation. *IEEE Trans. Knowl. Data Eng.* (2016).
[76] Zhou Zhao, Shuwen Xiao, Zehan Song, Chujie Lu, Jun Xiao, and Yueting Zhuang. 2020. Open-Ended Video Question Answering via Multi-Modal Conditional Adversarial Networks. *IEEE Trans. Image Process.* (2020).
[77] Zhou Zhao, Zhu Zhang, Xinghua Jiang, and Deng Cai. 2019. Multi-Turn Video Question Answering via Hierarchical Attention Context Reinforced Networks. *IEEE Trans. Image Process.* (2019).
[78] Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhenxin Xiao, Xiaohui Yan, Jun Yu, Deng Cai, and Fei Wu. 2019. Long-Form Video Question Answering via Dynamic Hierarchical Reinforced Networks. *IEEE Trans. Image Process.* (2019).
[79] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral collaborative filtering.. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*.
[80] Fuzhen Zhuang, Zhiqiang Zhang, Mingda Qian, Chuan Shi, Xing Xie, and Qing He. 2017. Representation learning via Dual-Autoencoder for recommendation. *Neural Networks* (2017).