10-2023

Multi-representation Variational Autoencoder via iterative latent attention and implicit differentiation

Nhu Thuat TRAN
Singapore Management University, nttran.2020@phdcs.smu.edu.sg

Hady Wirawan LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Multi-Representation Variational Autoencoder via Iterative Latent Attention and Implicit Differentiation

Nhu-Thuat Tran
School of Computing and Information Systems
Singapore Management University
nttran.2020@phdcs.smu.edu.sg

Hady W. Lauw
School of Computing and Information Systems
Singapore Management University
hadywlauw@smu.edu.sg

ABSTRACT

Variational Autoencoder (VAE) offers a non-linear probabilistic modeling of user's preferences. While it has achieved remarkable performance at collaborative filtering, it typically samples a single vector for representing user's preferences, which may be insufficient to capture the user's diverse interests. Existing solutions extend VAE to model multiple interests of users by resorting a variant of self-attentive method, i.e., employing prototypes to group items into clusters, each capturing one topic of user's interests. Despite showing improvements, the current design could be more effective since prototypes are randomly initialized and shared across users, resulting in uninformative and non-personalized clusters.

To fill the gap, firstly, we introduce *iterative latent attention* for personalized item grouping into VAE framework to infer multiple interests of users. Secondly, we propose to incorporate *implicit differentiation* to improve training of our iterative refinement model. Thirdly, we study the self-attention to refine cluster prototypes for item grouping, which is largely ignored by existing works. Extensive experiments on three real-world datasets demonstrate stronger performance of our method over those of baselines.

CCS CONCEPTS

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

KEYWORDS

Collaborative filtering, multi-interest user modeling

ACM Reference Format:
Nhu-Thuat Tran and Hady W. Lauw. 2023. Multi-Representation Variational Autoencoder via Iterative Latent Attention and Implicit Differentiation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614980

1 INTRODUCTION

Variational AutoEncoder (VAE) has established a new pathway for Collaborative Filtering (CF) via neural network-based non-linear probabilistic modeling. Representative VAE-based CF models [19, 29] typically render a single vector for representing user's interests. We are increasingly cognizant that a user may well have varied interests, and a single representation vector may be insufficient to capture the full breadth of a user's diverse interests.

Recently, researchers have shown improvements by using multiple embedding vectors to represent a user [4, 17, 23, 48]. MacridVAE [23], the most representative VAE-based multi-interest CF model, has shown superior performance compared to MultiVAE [19], a notable VAE-based single-embedding user representation work. Even so, MacridVAE has a couple of shortcomings. For one, it employs randomly initialized prototypes to group items into several clusters, each of which capturing one user's topic of interests, which could result in uninformative clusters. For another, cluster prototypes are shared across users, which could be less effective since recommender systems are oriented towards personalization. Thus, our goal is to design a more effective VAE-based architecture for multi-interest modeling in collaborative filtering.

Existing multi-interest modeling recommendation models widely adopt two approaches, *self-attentive* method and *dynamic routing* mechanism. The former method (of which a variant is adopted by MacridVAE), relying on *attention* mechanism, typically employs a set of prototypes, rather than computing from input, to attend to different historical behaviors of users. The resulting multiple vectors are taken as multi-interest representations. As would be analyzed in Section 3, *self-attentive* method has the drawbacks of uninformative and non-personalized cluster prototypes. The latter method, inspired by dynamic routing algorithm from Capsule network [27], treats interest representations as higher-level capsules and refine their representations through routing from lower-level capsules, i.e., interacted items. The routing process potentially resolves the shortcomings of self-attentive method by iteratively refining and personalizing high-level capsules, i.e., cluster prototypes. However, iterative refinement suffers from training instabilities and increases the model complexity, making it challenging to apply to large datasets.

To resolve the shortcomings of existing works, firstly, we introduce *iterative latent attention* for personalized item grouping into VAE framework to infer multiple user's interests, which would refine cluster prototypes to produce more informative and personalized item clusters. Secondly, to reduce the complexity stemming from iterative refinement, we propose to incorporate *implicit differentiation* to ease the training process and to strive for better recommendation accuracy. Thirdly, we study the interactions between cluster prototypes via self-attention, which has received less attention by prior multi-interest modeling works. Encouraging empirical results suggest that this a direction to improve recommendation performance besides designing more effective multi-interest extractor.

Contributions. Our primary contributions are *first*, we study existing works on multi-interest modeling under a unified clustering lens to uncover their shortcomings. *Second*, we design a novel and

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614980

more effective VAE-based model called VALID, which stands for Variational Autoencoder via iterative Latent attention and Implicit Differentiation. VALID equips *iterative latent attention* and *implicit differentiation*, achieving higher recommendation accuracy than related VAE-based baselines. *Third*, we explore the interactions between cluster prototypes, which yields interesting insights. *Last but not least*, we conduct experiments on three real-world datasets to demonstrate the favorable performance of VALID.

2 RELATED WORK

VAE-based Recommender Systems. MultiVAE [19] is a notable VAE-based Collaborative Filtering (CF) model, proposing the use of multinomial likelihood and adjusting standard VAE objective for recommendation task. Subsequent works [14, 26, 28, 39, 51] have been proposed, but these works typically render a single vector for user representation, which is insufficient to capture the diversity of user's interests. MacridVAE [23] is the most representative VAE-based multi-interest model, inspiring a series of later works [9, 38, 47]. Our work generalizes MacridVAE by introducing iterative latent attention and implicit differentiation for multi-interest modeling.

Multi-Interest User Modeling. Representative works that employed *self-attentive* method include MacridVAE [23] for collaborative filtering and ComiRec-SA [4] for sequential recommendation. [38, 47] leverage external information to improve MacridVAE. A series of works improve ComiRec-SA for sequential recommendation [7, 16, 18, 31, 33, 43, 45, 53, 54]. Regarding *dynamic routing* approach, DGCF [48] and MIND [17] are notable for routing-based multi-interest modeling, which are inspired by dynamic routing from Capsule network [27]. A string of subsequent works following this direction are [4, 5, 7, 37, 49]. Another less common direction for multi-interest modeling is to include multiple user vectors [2, 34, 50].

Our work shares the same spirit with MacridVAE and DGCF and is distinct in several respects. Firstly, we employ *iterative latent attention* for personalized item grouping to derive multiple user's interests, generalizing MacridVAE. Secondly, we study *implicit differentiation* to resolve training instabilities caused by iterative refinement in our model. This study not only improves the training, resulting in better accuracy, but also brings insight on the connection between clustering and multi-interest modeling. Thirdly, we model interactions between cluster prototypes by self-attention.

Iterative Representation Learning. A wide range of models have employed iterative refinement to induce a set latent representations from inputs. Typically, starting from an initial guess (e.g., random), these models iterate multiple rounds over input to embed informative information to transform the initial guess into the desired solution. Capsule network and variants [27, 40] iteratively update representation of higher-level capsules based on lower-level capsules, as applied to image reconstruction, classification and segmentation. Slot Attention [22] and its extension [15] leverage iterative attention mechanism to produce a set of vectors capturing objects from input images or videos. Perceiver [11, 12] uses multiple cross-attention and self-attention blocks repeatedly to efficiently learn representations from inputs with a large number of elements and apply to multiple modalities without much change to the architecture. Perceiver has inspired many works to apply iterative refinement procedure into various tasks, e.g., graph modeling [1], speech processing [41], vision-language modeling [35], robotic manipulation [30], autoregressive modeling [10].

These works share in common a soft clustering [3] structure, in which each cluster is associated with one component from inputs, e.g., objects from images. Correspondingly, multi-interest modeling induces the clusters underlying user's historical interactions, and each cluster represents a certain topic of user interest. The semantic connection between these two disciplines motivate us to study a novel iterative approach for multi-interest user modeling.

3 PRELIMINARIES

Problem Formulation. Our problem setting includes a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$ with $M = |\mathcal{U}|$ and $N = |\mathcal{I}|$ being the number of users and items, respectively. Let $\mathbf{R} \in \{0,1\}^{M \times N}$ represent the interactions between users and items. Let $\mathbf{r}^u \in \mathbf{R}, \mathbf{r}^u_l = 1$ indicate user u interacted with item l, otherwise $\mathbf{r}^u_l = 0$. The goal is to predict the likelihood of interactions between user u and item l that u has not interacted. To achieve this goal, the key is to effectively capture user preferences from her adopted items. Let $\mathcal{I}^u$ be the set of all items that user u interacted, i.e., $\mathbf{r}^u_l = 1$. Given $\mathcal{I}^u$ as inputs, multi-interest extractor produces a set of K vectors $\mathbf{z}^u = \{\mathbf{z}^u_k\}^K_{k=1}$, $\mathbf{z}^u_k \in \mathbb{R}^d$, to capture K topics of interest for user u.

Multi-Interest Modeling under Clustering Lens. Two most popular methods to derive multiple interests of user for recommendation are *self-attentive method* and *dynamic routing*. We study these approaches under the clustering lens to understand how they work and draw the connection to our proposed model.

Under clustering lens, the problem of multi-interest modeling in recommender systems is to induce K groups of items underlying user u's adopted items set $\mathcal{I}^u$ with $L = ||\mathcal{I}^u||$. Mathematically, let $\boldsymbol{\theta}^u = \{\boldsymbol{\theta}^u_k\}^K_{k=1}$ represent K centroids/prototypes of K clusters for user u; $\boldsymbol{\phi}^u = \{\boldsymbol{\phi}^u_{lk}\}^L_{l=1}$ denote the assignment of each item $l \in \mathcal{I}^u$ to cluster k. Each interest of user is obtained by aggregating item representations assigned to the corresponding cluster. Let $\mathbf{z}^u_k$ denote the k-th interest representation of user u and $\mathbf{H}_l$ is representation items l, we have $\mathbf{z}^u_k = \sum^L_{l=1} \boldsymbol{\phi}^u_{lk} \mathbf{H}_l$. Existing works mainly differ by how they derive cluster prototypes $\boldsymbol{\theta}^u$ and cluster assignment $\boldsymbol{\phi}^u$.

Self-Attentive Method. Existing works mainly employ the idea from [20], which is described by the following equations:

$$\begin{aligned} \mathbf{A} &= \underset{K}{\text{softmax}}(\mathbf{W}_2 tanh(\mathbf{W}_1 \mathbf{H}^T))^T \\ \mathbf{V} &= \mathbf{A}^T \mathbf{H} \end{aligned} \tag{1}$$

$\mathbf{H} \in \mathbb{R}^{L \times d}$ contains representations of adopted items. $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ projects $\mathbf{H}$ to latent space. $\mathbf{W}_2 \in \mathbb{R}^{K \times d}$ is interpreted as K cluster prototypes[1], i.e., $\boldsymbol{\theta}^u$ in clustering framework. $\mathbf{A} \in \mathbb{R}^{L \times K}$ represents the weights of assigning items to clusters, i.e., $\mathbf{A}_{lk}$ is the weight of assigning item l to cluster k. Hence, $\mathbf{A}$ acts as $\boldsymbol{\phi}^u$. Finally, user interests are represented by $\mathbf{V} \in \mathbb{R}^{K \times d}$, obtained by aggregating item representations weighted by assignment scores, i.e., $\mathbf{V} = \mathbf{A}^T \mathbf{H}$.

Self-attentive method has its own shortcomings. Firstly, cluster prototypes $\mathbf{W}_2$ is randomly initialized and updated solely based on supervision signals from user-item interactions, which is often sparse. This results in sub-optimal performance. Secondly, $\mathbf{W}_2$ is

[1]We consider dimension of $\mathbf{W}_2$, which is equal to that of $\mathbf{W}_1\mathbf{H}$, is d for simplicity

shared across users, which is less personalized. As such, one of our goals in this paper is to address these two drawbacks.

Dynamic Routing Method. Existing works mainly follow the idea from MIND [17], which repeats following equations R times

$$\mathbf{A}_{lk} = \frac{exp(\mathbf{b}_{lk})}{\sum_{k=1}^{K} exp(\mathbf{b}_{lk})} \quad \text{with} \quad \mathbf{b}_{lk} = (\mathbf{W}_{lk}\mathbf{H}_l)^T\mathbf{v}_k$$
$$\mathbf{v}_k = squash(\sum_{l=1}^{L} \mathbf{A}_{lk}\mathbf{W}_{lk}\mathbf{H}_l) \ (squash(\mathbf{x}) = \frac{||\mathbf{x}||_2}{1+||\mathbf{x}||_2}\frac{\mathbf{x}}{||\mathbf{x}||_2}) \tag{2}$$

$\mathbf{W}_{lk} \in \mathbb{R}^{d\times d}$ is the connection weight[2] between item l (low-level capsule) and user interest k (high-level capsule). $\mathbf{b}_{lk}$ is the routing logit between item l and cluster k. $\mathbf{A}_{lk}$ is the weight of assigning item l to interest k (cluster k). Therefore, $\mathbf{A} \in \mathbb{R}^{L\times K}$ plays the role of $\boldsymbol{\phi}^u$. $\mathbf{v}_k \in \mathbb{R}^d$ represents k-th interest (cluster) of user, obtaining by aggregating adopted items representations followed by *squash* non-linearity. Hence, $\mathbf{V} \in \mathbb{R}^{K\times d}$ plays the role of $\boldsymbol{\theta}^u$.

Equation 2 implies that dynamic routing has potential to address issues of self-attentive method. Firstly, k-th cluster prototype $\mathbf{v}_k$ is aggregated from user adopted items, which is personalized. Secondly, $\mathbf{v}_k$ is iteratively refined and further updated with supervision signals, which is more informative than randomly initialized ones. *However*, as there is no ground-truth label in each iteration, the iterative refinement of dynamic routing results in deep model, which is known to be difficult to train, affecting the model performance.

Overview of Proposed Approach. *First*, we introduce *iterative latent attention* for personalized item grouping into VAE framework, which iteratively updates cluster prototypes to alleviate non-personalized and uninformative prototypes problems of existing VAE-based self-attentive methods. *Second*, we employ *implicit differentiation* to alleviate the training difficulty caused by iterative refinement, which has not been done by existing dynamic routing multi-interest modeling methods. *Thirdly*, we study self-attention methods to explicitly model the interactions between cluster prototypes.

4 METHODOLOGY

Figure 1 presents the architecture of our proposed model VALID. The core idea is *iterative latent attention* for personalized item grouping. Concretely, prototypes in latent space iteratively attend to input items to group items into meaningful clusters. Prototypes are then updated to become more informative and personalized, meaning that each user has her own set of prototypes. Updated prototypes are used in subsequent cluster steps to better identify item groups.

Let $\mathbf{A} \in \mathbb{R}^{N\times K}$ be the assignment scores of N items to K interests (clusters). $\mathbf{m} \in \mathbb{R}^{K\times d}$ is the set of randomly initialized K cluster prototypes and $\mathbf{m}^u \in \mathbb{R}^{K\times d}$ be the set of updated cluster prototypes of user u. $\mathbf{H} \in \mathbb{R}^{N\times d}$ denote the item embedding matrix of N items, each of dimension d. Next, we will describe details of our proposed model, following the illustration of Figure 1 from left to right.

4.1 Iterative Latent Attention for Personalized Item Grouping

VALID first groups N items into K clusters via an iterative manner. A series of clustering blocks are employed, each has *cluster* and *update* steps, except the last block only includes *cluster* step. In the first clustering block, cluster centroids in $\mathbf{m}$ are shared across users and are refined in the subsequent blocks to make it personalized and informative. Using a set of independent prototypes $\mathbf{m}$, which *is not* a function of N, is favorable as it can scale up without burdening model's trainable parameters. We start by describing the first clustering block then present the proposed iterative approach.

4.1.1 ***Clustering Block.*** As depicted in Figure 1, each clustering block includes two steps, *Group items* and *Update prototypes*. The second step, *Update prototypes* includes updating prototypes and self-attention to model interactions between prototypes.

Group Items. As its name suggests, this step groups items into several clusters. The input includes prototypes $\mathbf{m}$ and item matrix $\mathbf{H}$ and the output is assignment scores $\mathbf{A}$ obtained as follows.

$$\mathbf{b}_{lk} = \mathbf{H}_l^T\mathbf{m}_k/(\tau \cdot ||\mathbf{H}_l||_2 \cdot ||\mathbf{m}_k||_2) \ \forall k = 1, 2, ..., K$$
$$\mathbf{A}_l \sim CATE(SOFTMAX([\mathbf{b}_{l1}, \mathbf{b}_{l2}, .., \mathbf{b}_{lK}])) \tag{3}$$

$\mathbf{A}_l \in \mathbb{R}^K$ is the score of assigning item l to interest (cluster) k. $\mathbf{A}_l$ is one-hot approximated vector estimated by Gumbel-Softmax [13] sampling (CATE). $\mathbf{b}_{lk}$ measures the cosine similarity between item representation $\mathbf{H}_l$ and cluster representation $\mathbf{m}_k$. Cosine similarity helps to prevent mode collapse where items are mostly grouped to a single prototype with highest magnitude [23]. A small temperature τ helps concentrate score to the most similar cluster.

Existing self-attentive methods stop at this step and move to user's interest aggregation. The inherent drawback is that the resulting clusters are based on randomly and non-personalized cluster prototypes $\mathbf{m}$, which negatively affects recommendation accuracy.

Update Prototypes. To alleviate this shortcoming, we propose to update and embed personalization into cluster representations, i.e., $\mathbf{m}$ becomes $\mathbf{m}^u$ (symbol ^{u} means association to a specific user u). To understand why personalization is needed for clustering, let us examine the case when $\mathbf{m}$ contains only one element, i.e., $\mathbf{m} \in \mathbb{R}^d$. In this case, $\mathbf{A}_l$ (without softmax normalization) measures the importance of each item l to user. In other words, $\mathbf{A}_l$ represents the degree to which user likes item l. Therefore, sharing $\mathbf{m}$ across users is equivalent to setting the same preference weight for each item across users, which is in contrast with the intention of personalization in recommender systems.

This step's input includes assignment scores $\mathbf{A}$, non-personalized prototypes $\mathbf{m}$ and the output is personalized prototypes $\mathbf{m}^u$

$$\mathbf{m}_k^u = \sum_{l=1}^{N} \mathbf{r}_l^u\mathbf{A}_{lk}\mathbf{H}_l \quad \forall k = 1, 2, ..., K \tag{4}$$

In Equation 4, each $\mathbf{m}_k^u$ is updated based on user rating vector $\mathbf{r}^u \in \{0, 1\}^N$. Therefore, the resulting $\mathbf{m}^u \in \mathbb{R}^{K\times d}$ are personalized and each user has their own set of clusters, depending on their rating vector $\mathbf{r}^u$. Additionally, users with similar rating vectors will have similar set of clusters because $\mathbf{m}^u$ is updated based on $\mathbf{r}^u$, i.e., $\mathbf{m}^u \approx \mathbf{m}^{u'}$ given $\mathbf{r}^u \approx \mathbf{r}^{u'}$ ($\approx$ is approximation symbol).

Self-Attention between Prototypes. Interactions between cluster prototypes have received less attention by existing works. Although each prototype represents a single interest, there may be the case that multiple interests of one user are related to one another. In this case, prototypes should exchange their information with others to refine themselves. Thus, we study self-attention [42] between

[2]The dimension of $\mathbf{W}_{lk}$ is $d \times d$ to ease the understanding.

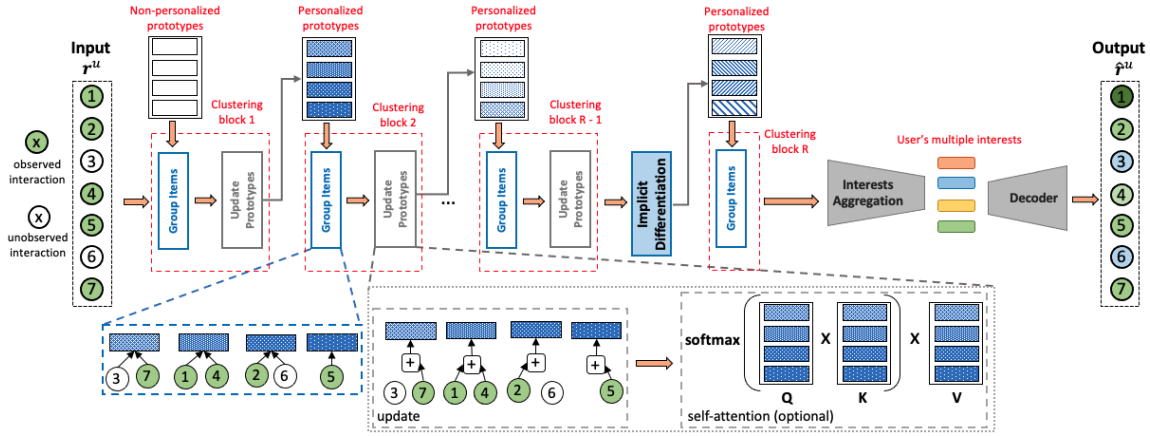


Figure 1: Architecture of VALID. **Input is the interaction vector $\mathbf{r}^u$ of user u. Then an array of clustering blocks is designed to group items into meaningful clusters. Each clustering block includes group and update steps, except the last one. In update step, only observed interacted items are considered, followed by an optional self-attention block. An *implicit differentiation* is added after $(R-1)^{th}$ block and before R^{th} block to improve training. In the last step, output of clustering is passed to interests aggregation layer, followed by decoder to predict rating $\hat{\mathbf{r}}^u$.**

prototypes, allowing them to read information from other prototypes of the same user. The input of self-attention is $\mathbf{m}^u$ and the output is its refined version. We study two self-attention variants.

Scaled Cosine Attention. Attention score is calculated based on cosine similarity. $\mathbf{m}^u$ is updated as $\mathbf{m}^u = \mathbf{B}\mathbf{m}^u + \mathbf{m}^u$, in which:

$$\mathbf{B}_{kj} = softmax(\frac{\mathbf{m}^u_k \cdot \mathbf{m}^u_j}{\tau_{sa} \cdot ||\mathbf{m}^u_k||_2 \cdot ||\mathbf{m}^u_j||_2}) \; \forall k, j = 1, 2, ..., K \quad (5)$$

We follow [8] to add residual connection to prevent vanishing or exploding gradients over multiple iterations. Equation 5 can be used in recursive manner for high order information exchange. τ_{sa} controls information exchange between prototypes. When $\tau_{sa} \rightarrow 0$, output of softmax is one-hot vector, i.e., $\mathbf{B}_{kj} = 1$ if $k = j$ and $\mathbf{B}_{kj} = 0$ if $k \neq j$. As such, self attention acts as identity function, i.e., no information exchanged between prototypes. Large τ_{sa} results in averaging information between prototypes, i.e., $\mathbf{m}^u_k$ and $\mathbf{m}^u_j$ are highly similar.

Compositional De-Attention. (CoDA for short) Unlike scaled cosine attention, CoDA [36] introduces negative attention weights, which allows *add*, *delete* and *forget* information.

$$\mathbf{B} = tanh(\mathbf{B}^t) \odot sigmoid(\mathbf{B}^s) \quad (\mathbf{B} \in \mathbb{R}^{K \times K})$$
$$\mathbf{B}^t = \mathbf{B}^t - mean(\mathbf{B}^t), \quad \mathbf{B}^t_{kj} = \frac{\mathbf{m}^u_k \cdot \mathbf{m}^u_j}{\tau_{sa} \cdot ||\mathbf{m}^u_k||_2 \cdot ||\mathbf{m}^u_j||_2} \; \forall k, j = 1, ..., K$$
$$\mathbf{B}^s = \mathbf{B}^s - mean(\mathbf{B}^s), \mathbf{B}^s_{kj} = -\frac{1}{\tau_{sa}} ||\mathbf{m}^u_k - \mathbf{m}^u_j||_1 \; \forall k, j = 1, ..., K \quad (6)$$

$\odot$ is element-wise multiplication. $||\cdot||_1$ is L_1 norm. $\mathbf{B}^t \in \mathbb{R}^{K \times K}$ measures similarity while $\mathbf{B}^s$ measures the negative dissimilarity between prototypes inside $\mathbf{m}^u$. $\mathbf{B}$ is towards 1 iff $tanh(\mathbf{B}^t)$ and $sigmoid(\mathbf{B}^s)$ are both close to 1. $\mathbf{B}$ is -1 iff $tanh(\mathbf{B}^t)$ is -1 and $sigmoid(\mathbf{B}^s)$ is 1. Furthermore, $sigmoid(\cdot)$ controls whether to forget information, e.g., *forget* when $sigmoid(\cdot) \approx 0$. $\frac{1}{\tau_{sa}}$ acts as hyperparameters α and β in original paper. We hypothesize CoDA has more representation capacity than scaled cosine attention to model the interactions between prototypes. Subtracting $mean(\cdot)$ from $\mathbf{B}^t$ and $\mathbf{B}^s$ is to ensure they have both negative and positive values. Similar to scaled cosine attention, we update $\mathbf{m}^u$ as $\mathbf{m}^u = \mathbf{B}\mathbf{m}^u + \mathbf{m}^u$.

4.1.2 ***Iterative Clustering.*** Subsequent clustering blocks, indexed from 2 to $R-1$ in Figure 1, has the similar mathematical formulation as described in the first block in Section 4.1.1. The main difference is the input prototypes, which is personalized prototypes $\mathbf{m}^u$. The main idea here is that each block accepts updated and personalized prototypes from previous block as input and uses these prototypes to group items. This is followed by updating prototypes for the next block. As a result, in each clustering block, prototypes are refined to better capture user's preferences, leading to better grouping items in the next block. This process is presented in detail by function *Iterative Item Grouping* in Algorithm 1. Iterative prototype refinement creates a deeper network, which increases model complexity and is known to suffer from training instabilities. Next, we present an *implicit differentiation* approach to resolve this issue.

4.2 Implicit Differentiation

Updating cluster prototypes, i.e., from $\mathbf{m}$ to $\mathbf{m}^u$, to better reflect user's preferences is iterative refinement applied in representation learning. From attention perspective [12], each clustering block employs a cross-attention layer to group items. As noted by [6], iterative refinement is particularly useful for this case thanks to symmetric-breaking mechanism, which is brought by cross attention. However, iterative refinement is difficult to train due to its unsupervised learning nature, as we do not have access to ground-truth interests of users. Additionally, the clustering process has recurrent form, resulting in deep network, which causes training instabilities as well as increases the computational complexity. We propose to incorporate *implicit differentiation* [6] to improve training.

From optimization perspective, the process of grouping items into clusters for recommendation in VALID can be seen as:

$$\min_{\mathbf{m}^u, \boldsymbol{\phi}^u} \sum_{u=1}^{M} \mathcal{L}^{rec}(\mathcal{I}^u, \mathbf{m}^u, \boldsymbol{\phi}^u) \; s.t. \; \mathbf{m}^u = \underset{\mathbf{m}^u}{argmin} \; \mathcal{L}^{clus}(\mathcal{I}^u, \mathbf{m}^u, \boldsymbol{\phi}^u) \quad (7)$$

$\mathcal{I}^u$ is the set of user u's adopted items, $\mathbf{m}^u, \boldsymbol{\phi}^u$ represent personalized cluster representations and the assignment of items to clusters for user u. $\mathcal{L}^{rec}$ is the recommendation objective and $\mathcal{L}^{clus}$ is the clustering objective, guided by recommendation supervision signals. From this formulation, we further understand the connection

between item clustering and recommendation. Clustering serves as inner objective for outer objective recommendation. $\mathbf{m}^u$ is updated in each iteration, i.e., $\mathbf{m}^u_{r+1} = f_{cluster}(\mathbf{m}^u_r, \mathcal{I}^u)$ with $f_{cluster}$ is the clustering function at iteration r, while $\boldsymbol{\phi}^u$ is updated as intermediate step inside $f_{cluster}$.

Back-propagating gradients through clustering function which has recursive form causes training instabilities. To resolve this problem, we leverage *implicit differentiation* [6]. As illustrated in Lines 27-28 in Algorithm 1, *implicit differentiation* is implemented by differentiating computation graph of applying update $\mathbf{m}^u$ once to $\mathbf{m}^u_*$, which is the output of multiple clustering rounds.

By equipping *implicit differentiation*, the time complexity (as a function of the number of iterations R) of forward pass remains $O(R)$ while forward space, backward time and backward space complexities are all $\boldsymbol{O(1)}$. Because the gradient to $\mathbf{m}^u$ is detached before the last iteration, the complexity depends on the last iteration only. MacridVAE, a representative of VAE-based self-attentive multi-interest modeling method, has time complexity $O(1)$ of forward pass because there is no iterative refinement inside. Otherwise, forward space, backward time and backward space complexities of MacridVAE are the same as those of VALID. It is worth noting that forward pass is matrix multiplication, which is highly optimized on GPU. Therefore, forward pass is not the bottleneck of VALID compared to MacridVAE. Furthermore, VALID achieves significantly higher recommendation accuracy thanks to the iterative nature of forward pass, as presented in Table 2.

4.3 Interest Aggregation

After obtaining assignment scores $\mathbf{A}^u$, this step aggregates item representations assigned to clusters to produce multiple interest representations of user. Concretely, a context matrix $\mathbf{E} \in \mathbb{R}^{N \times d_{enc}}$ is employed to derive interest representations. Despite being similar with $\mathbf{H}$, i.e., storing N representations, $\mathbf{E}$ is different from $\mathbf{H}$ in the sense that from auto-encoder perspective, $\mathbf{E}$ is the weight of encoder while $\mathbf{H}$ is the weight of decoder.

$$\mathbf{x}^u_k = \frac{\sum_{l=1:\mathbf{r}^u_l=1}^{N} \mathbf{A}^u_{lk}\mathbf{E}_l}{\sqrt{\sum_{l:\mathbf{r}^u_l=1}^{N}(\mathbf{A}^u_{lk})^2}} \quad \forall k = 1, 2, ..., K \tag{8}$$

In Equation 8, each $\mathbf{x}^u_k$ of user u is derived by aggregating the representation of user u adopted item, i.e., item l such that $\mathbf{r}^u_l = 1$. Each item has a weight $\mathbf{A}^u_{lk}$, showing how item l fits interest k of user u. The denominator is for normalizing assignment score in $\mathbf{A}^u$. Finally, we follow MacridVAE [23] to derive final representation of interests. Each $\mathbf{x}^u_k, \forall k = 1, 2, ..., K$, is further processed by a function $f_{nn} : \mathbb{R}^{d_{enc}} \rightarrow \mathbb{R}^{2d}$ to estimate parameters of Gaussian distribution

$$\boldsymbol{\mu}^u_k = \frac{\mathbf{y}^u_k}{||\mathbf{y}^u_k||_2}; \ \boldsymbol{\sigma}^u_k = \sigma_0 \cdot exp(-\frac{1}{2}\mathbf{t}^u_k) \text{ with } (\mathbf{y}^u_k, \mathbf{t}^u_k) = f_{nn}(\mathbf{x}^u_k) \tag{9}$$

$\boldsymbol{\mu}^u_k$ is the estimated mean of distribution while $\boldsymbol{\sigma}^u_k$ is the estimated of log variance. σ_0 is a hyper-parameter, whose value is around 0.1 as noted by [23]. Then the k-th interest of user is sampled from Gaussian distribution, i.e., $\mathbf{z}^u_k \sim \mathcal{N}(\boldsymbol{\mu}^u_k, [diag(\boldsymbol{\sigma}^u_k)]^2)$.

4.4 Learning and Optimization

Decoder. Given K interests of user u, represented by $\mathbf{z}^u = \{\mathbf{z}^u_k\}^K_{k=1}$ and assignment scores $\mathbf{A}^u$, our model predicts the likelihood of interaction between user u and item l, i.e., $p(\mathbf{r}^u_l|\mathbf{z}^u, \mathbf{A}^u)$ as follows:

$$\begin{aligned} g(\mathbf{z}^u_k) &= exp(\frac{(\mathbf{z}^u_k)^T \cdot \mathbf{H}_l}{\tau_{dec} \cdot ||\mathbf{z}^u_k||_2 \cdot ||\mathbf{H}_l||_2}) \\ p(\mathbf{r}^u_l|\mathbf{z}^u, \mathbf{A}^u) &= \frac{\sum_{k=1}^{K} \mathbf{A}^u_{lk} \cdot g(\mathbf{z}^u_k)}{\sum_{l=1}^{N}\sum_{k=1}^{K} \mathbf{A}^u_{lk} \cdot g(\mathbf{z}^u_k)} \end{aligned} \tag{10}$$

Learning Objective. Our learning objective, following β-VAE [23], over a batch of user $\mathcal{B}$ is to maximize the following:

$$\mathcal{L} = \sum_{u \in \mathcal{B}} [\sum_{l=1}^{N} \mathbf{r}^u_l \ ln(p(\mathbf{r}^u_l|\mathbf{z}^u, \mathbf{A}^u)) - \beta \cdot D_{KL}(q(\mathbf{z}^u|\mathbf{r}^u, \mathbf{A}^u)||p(\mathbf{z}^u))] \tag{11}$$

The first term is the reconstruction objective, aiming at reconstructing observed user-item interactions. The second term is Kullback–Leibler (KL) divergence to match variational distribution $q(\mathbf{z}^u|\mathbf{r}^u, \mathbf{A}^u) = \Pi^K_{k=1}\mathcal{N}(\boldsymbol{\mu}^u_k, [diag(\boldsymbol{\sigma}^u_k)]^2)$ with prior distribution $p(\mathbf{z}^u) = \mathcal{N}(0, \sigma_0^2\mathbf{I})$, forcing the distribution of interests follows prior distribution. β controls the influence of KL divergence term, aiming at balancing the effects between recommendation objective and distribution regularization. The value of β is increased from 0 and 1 via annealing procedure as noted by [19].

5 EXPERIMENTS

Our experiments aim to answer the following research questions.

- **(RQ1)** How does VALID perform compared to existing multi-interest modeling collaborative filtering models?
- **(RQ2)** How do *iterative latent attention* and *implicit differentiation* affect model performance?
- **(RQ3)** Does self-attention between cluster prototypes bring information gain? Is self-attention suitable for this?
- **(RQ4)** How does iterative clustering affect user representations? Is VALID able to group items into meaningful clusters for user?

5.1 Experimental Settings

Datasets. Table 1 lists three datasets with implicit feedbacks.

- **Citeulike-a**[3] contains the interactions between users and articles, e.g., a user saves an article to their own collection.
- **Gowalla**[4] contains interactions between users and locations, e.g., a user shares her location by checking in.
- **Yelp**[5] contains reviews that users wrote for businesses. A review is considered one interaction.

For Citeulike-a, we keep the original data. For Gowalla and Yelp, we follow the common pre-processing practices [21, 24, 25, 48]. We retain users and items with at least 10 interactions for Gowalla. For Yelp, we consider interactions from 2016 onwards and keep users and items with at least 15 interactions. On all datasets, for each user, we randomly divide their interactions with ratio 0.8:0.1:0.1 for training, validation and test sets. Cold-start users and items in validation and test sets are discarded, as there have no trained parameters.

Competitors. We compare VALID against existing multi-interest models for collaborative filtering (**MacridVAE**, **DGCF**, **DPCML**) and VAE-based models (**MacridVAE**, **RecVAE**). We also include

[3] http://wanghao.in/CDL.htm
[4] https://github.com/RUCAIBox/RecSysDatasets
[5] https://www.yelp.com/dataset

Algorithm 1: Pseudo Code for VALID

Input: User rating vector $\mathbf{r}^u \in \{0,1\}^N$; Parameters Θ (item matrix $\mathbf{H} \in \mathbb{R}^{N \times d}$, context matrix $\mathbf{E} \in \mathbb{R}^{N \times d_{enc}}$, cluster representation $\mathbf{m} \in \mathbb{R}^{K \times d}$, parameters of neural network $f_{nn} : \mathbb{R}^{d_{enc}} \rightarrow \mathbb{R}^{2d}$); Number of clustering iterations R.
Output: updated Θ

$\mathbf{A}^u \leftarrow Iterative\ Item\ Grouping(\mathbf{H}, \mathbf{m})$
$\{\mathbf{z}_k^u\}_{k=1}^K \leftarrow Interest\ Aggregation(\mathbf{r}^u, \mathbf{A}^u)$
$\{p(\mathbf{r}_l^u | \mathbf{z}_u, \mathbf{A}^u)\}_{l=1}^N \leftarrow Decoder(\{\mathbf{z}_k^u\}_{k=1}^K, \mathbf{H}, \mathbf{A}^u)$
Calculate loss $\mathcal{L}$ as in Equation 11
Update Θ using gradients of $\mathcal{L}$

Function *Group items(**H**, **m**)*
  for *l = 1, 2, ..., N* **do**
    $\mathbf{b}_{lk} = \mathbf{H}_l^T \mathbf{m}_k / (\tau \cdot ||\mathbf{H}_l||_2 \cdot ||\mathbf{m}_k||_2), \forall k = 1, 2, ..., K$
    $\mathbf{A}_l \sim CATE(SOFTMAX([\mathbf{b}_{l1}, \mathbf{b}_{l2}, .., \mathbf{b}_{lK}]))$
  return ***A***
Function *Update prototypes(**A**, **m**, $\mathbf{r}^u$)*
  for *k = 1, 2, ..., K* **do**
    $\mathbf{m}_k^u = \sum_{l=1}^N \mathbf{r}_l^u \mathbf{A}_{lk} \mathbf{H}_l$
  $\mathbf{m}^u \leftarrow$ Self-Attention$(\mathbf{m}^u)$ `// Optional`
  return $\boldsymbol{m}^u$
Function *Iterative Item Grouping (**H**, **m**, $\mathbf{r}^u$)*
  for *r = 1, 2, ..., R* **do**
    if *r = 1* **then**
      $\mathbf{A} \leftarrow$ Group items$(\mathbf{H}, \mathbf{m})$
      $\mathbf{m}^u \leftarrow$ Update prototypes$(\mathbf{A}, \mathbf{m}, \mathbf{r}^u)$
    else
      $\mathbf{A}^u \leftarrow$ Group items$(\mathbf{H}, \mathbf{m}^u)$
      if *r = R* **then**
        stop
      $\mathbf{m}^u \leftarrow$ Update prototypes$(\mathbf{A}^u, \mathbf{m}^u, \mathbf{r}^u)$
    if *r = R - 1* **then**
      $\mathbf{m}^u = \mathbf{m}^u.detach()$ `// Implicit Differentiation`
  return $\boldsymbol{A}^u$ `// return assignment score`
Function *Interest Aggregation($\mathbf{r}^u$, $\boldsymbol{A}^u$)*
  for *k = 1 to K* **do**
    $\mathbf{x}_k^u = \frac{\sum_{l:\mathbf{r}_l^u=1} \mathbf{A}_{lk}^u \mathbf{E}_l}{\sqrt{\sum_{l:\mathbf{r}_l^u=1} (\mathbf{A}_{lk}^u)^2}}$
    $\mathbf{y}_k^u, \mathbf{t}_k^u = f_{nn}(\mathbf{x}_k^u)$
    $\boldsymbol{\mu}_k^u = \mathbf{y}_k^u / ||\mathbf{y}_k^u||_2 \quad \boldsymbol{\sigma}_k^u = \sigma_0 \cdot exp(-\frac{1}{2}\mathbf{t}_k^u)$
    $\mathbf{z}_k^u \sim \mathcal{N}(\boldsymbol{\mu}_k^u, [diag(\boldsymbol{\sigma}_u^k)]^2)$ `// $k^{th}$ interest`
  return $\{\boldsymbol{z}_k^u\}_{k=1}^K$
Function *Decoder($\{\mathbf{z}_k^u\}_{k=1}^K$, **H**, $\boldsymbol{A}^u$)*
  for *l = 1, 2, ..., N* **do**
    $g(\mathbf{z}_k^u) = exp((\mathbf{z}_k^u)^T \cdot \mathbf{H}_l \ / \ (\tau_{dec} \cdot ||\mathbf{z}_k^u||_2 \cdot ||\mathbf{H}_l||_2))$
    $p(\mathbf{r}_l^u | \mathbf{z}^u, \mathbf{A}^u) = \frac{\sum_{k=1}^K \mathbf{A}_{lk}^u \cdot g(\mathbf{z}_k^u)}{\sum_{l=1}^N \sum_{k=1}^K \mathbf{A}_{lk}^u \cdot g(\mathbf{z}_k^u)}$
  return $\{p(\boldsymbol{r}_l^u | \boldsymbol{z}^u, \boldsymbol{A}^u)\}_{l=1}^N$

Table 1: Statistics of our chosen datasets after pre-processing.

Data	#users	#items	#interactions
Citeulike-a	5,551	16,945	204,929
Gowalla	29,858	40,988	1,027,464
Yelp	29,111	22,121	1,052,627

recently state-of-the-art CF models in the last two years as baselines (**SimpleX**, **UltraGCN**, **NCL**, **SimGCL**, **DirectAU**).

- **MacridVAE** [23] models macro- and micro-level of disentanglement using β-VAE for recommendation.
- **DGCF** [48] disentangles multiple factors representations of users and items through iterative refinement on interaction graph.
- **DPCML** [2] includes multiple representation vectors for users to improve Collaborative Metric Learning.
- **RecVAE** [29] introduces various techniques to improve training VAE for collaborative filtering.
- **SimpleX** [24] proposes cosine contrastive loss and large negative sampling ratio to improve collaborative filtering.
- **UltraGCN** [25] improves CF by approximating the limit of message passing layers and incorporating item-item relationships.
- **NCL** [21] incorporates structural and semantic neighbors via contrastive learning to improve graph-based collaborative filtering.
- **SimGCL** [52] adds random noise to representations for augmentation and regulate uniformity to enhance contrastive learning.
- **DirectAU** [44] directly optimizes uniformity and alignment of representations to improve recommendation accuracy.

We do not compare with MultiVAE [19] since MacridVAE and RecVAE have shown superior performance compared to MultiVAE. As we are working on collaborative filtering, which does not consider time dimension for recommendation, we do not compare VALID against existing multi-interest modeling methods for sequential recommendation, e.g., MIND[17] or ComiRec [4].

Hyper-parameter Settings. We set the embedding size to 64 for all datasets. For multi-interest models MacridVAE, DPCML and DGCF, the number of interests is 4. For baselines, we search other hyper-parameters following the range in original papers and choose those that achieve best results on validation set. For VALID, we set the hyper-parameters identically to those of MacridVAE for fair comparison: dropout rate is 0.5, σ_0 is chosen from $\{0.05, 0.075, 0.1\}$, β is chosen from $\{0.2, 0.5, 1\}$, the total number of annealing steps is from $\{5000, 10000, 20000\}$. For Gowalla dataset, $\tau_{dec} = 0.08$ and for Yelp and Citeulike-a, $\tau_{dec} = 0.1$. τ is set to 0.1 for all datasets. We use 1-layer hidden layer for MacridVAE and VALID with hidden size is searched in range $\{64, 128, 256, 512\}$. All models are trained with Adam optimizer and the learning rate is set to 0.001 for VALID and MacridVAE. For other models, we search learning rate in range $\{0.0001, 0.0003, 0.001\}$. The default value of R is 2. All models are trained on NVIDIA RTX 2080 Ti GPU machine. We run each model ten times with different random seeds and report the averaged numbers on test set. Training phase stops after 15 epochs without improving Recall@20 on validation set.

Evaluation Metrics. We use Recall at top P (Recall@P) and Normalized Discounted Cummalative Gain at top P (NDCG@P) [32] to evaluate recommendation performance. We follow the full ranking strategy in [55] and report numbers with $P = 20$ and $P = 50$.

5.2 Performance Comparison (RQ1)

Table 2 presents recommendation performance of all models. For MacridVAE and VALID, we report numbers generated by model with encoder hidden size 64. We study more values of encoder hidden size in Section 5.3. We also do not use self-attention between prototypes and study its effect in Section 5.4. Overall, our proposed model VALID achieves higher recommendation accuracy than all baselines w.r.t. all chosen metrics.

Among the baselines, MacridVAE stands out, achieving better accuracy than others in most cases (except lower NDCG@20 than SimpleX on Gowalla). VALID performance gain over MacridVAE is attributed to *iterative latent attention* and *implicit differentiation*, which will be extensively verified in Section 5.3. Despite including multiple representations for user, DPCML performance is close to or even lower than single-embedding representation (SimpleX, UltraGCN, NCL, DirectAU). This reveals that merely employing multiple vectors for user representation is not as effective as grouping items employed in VALID and MacridVAE. Regarding DGCF, VALID enjoys much better recommendation accuracy on chosen datasets. One explanation is the design of DGCF, i.e., dividing representation vector into K factors evenly, resulting in small number of dimensions for each factor, which may be insufficient to capture user's interests. In contrast, the employment of prototypes in VALID allows aggregating interest vectors with the same size as that of item.

Although employing β-VAE like VALID, RecVAE's performance is much lower. This stems from the fact that RecVAE is single-embedding model, highlighting the need of multi-interest modeling.

Regarding recently developed CF models, i.e., SimpleX, UltraGCN, NCL, SimGCL, DirectAU, despite varying in their approaches for collaborative filtering, they have in common in how they represent users and items by a single vector only. Despite of that, these models achieve comparable or even higher accuracy than multi-interest modeling counterparts DGCF and DPCML, showing that they are actually strong baselines for CF. VALID is much better than these methods, showing the evidence of VALID's strength.

5.3 Studies of Model Design (RQ2)

We conduct a series of ablative studies to verify our contributions that incorporates *iterative latent attention* and *implicit differentiation* into VAE framework. We contrast the performance of VALID with that of MacridVAE to highlight the effects of our proposed methods.

Personalized item grouping results in better accuracy. From Figure 2, VALID employing *iterative latent attention* for personalized item grouping achieves higher results than MacridVAE with non-personalized item grouping. The gap between two models is bigger in case of multi-interest modeling, i.e., K > 2, which shows that *iterative latent attention* is able to differentiate user's preferences on different topics. As such, personalized item grouping plays the key role to improve multi-interest modeling.

Multi-interest modeling achieves better performance than single vector interest representation. From Figure 2, we observe that representing users with $K > 1$ vectors results in higher accuracy than using single vector. While Citeulike-a prefers small K, i.e., $K = 2$, and overly large K hurts the performance, increasing number of user interests K is generally beneficial on Gowalla and Yelp.

Implicit differentiation helps to resolve training instabilities and improve model performance. We previously showed that personalized item grouping via iterative latent attention is the key to performance gain. However, iterative refinement causes training difficulties and *implicit differentiation* is one of the solutions. From Figure 3, we observe that VALID with *implicit differentiation (ID)* obtains better recommendation results than VALID without *ID*. This supports our claim that *implicit differentiation* alleviates the difficulties caused by back propagating gradient through recursive network. Further contrasting performance w.r.t. number of iterations R in Figure 3, we found that when increasing R, i.e., making deeper network, the performance of model variant without *implicit differentiation* reduces significantly. However, this reduction can be (partially) alleviated when *implicit differentiation* appears.

Excessive number of clustering steps results in clusters may not align with recommendation objective. From Figure 3, increasing number of clustering iterations generally degrades performance except NDCG@20 on Yelp. This observation suggests that we should care about the relation between clustering objective and recommendation objective. From Section 4.2, clustering objective acts as inner objective of outer recommendation objective. As such, with higher number of iterations R, item clusters are highly personalized but does not benefit recommendation. We conjecture that this comes from updating prototypes based solely on user's adopted item set, which does not contain sufficient number of semantically related items. Therefore, a future direction is to leverage semantically related items improve updating cluster prototypes.

VALID **works better than MacridVAE w.r.t.** d_{enc}**.** Finally, we verify the performance of VALID w.r.t. d^{enc}, i.e., the dimension of each element of $\mathbf{E}$ in Algorithm 1. Evidently, as shown by Figure 4, VALID is better than MacridVAE across various dimensionalities.

5.4 Self-Attention between Prototypes (RQ3)

Interactions between cluster prototypes has received less attention by prior works. For completeness, we investigate the benefits of refining cluster prototypes, i.e., allowing prototypes to read/share their captured information from/to other interests. We tune τ_{sa} (Section 5.4) controlling the information exchange between prototypes in range $\{0.1, 0.2, 0.5, 0.8, 1\}$. Results are reported in Table 5. As we do not introduce any parameters in self-attention, the change in performance comes from information exchange between prototypes.

Refining prototypes by self-attention has potential to improve performance, which is shown by the changes in performance, particularly observed on Citeulike-a and Gowalla. We intuit that on these datasets, multiple interests of a user are similar to a certain extent. As such, refining prototypes to capture such similarity is beneficial to recommendation performance.

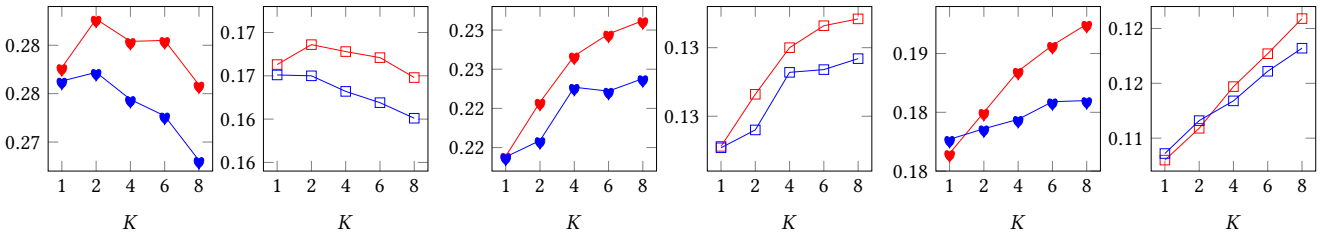
Compositional De-Attention is generally better for refining prototypes than scaled cosine attention, which supports our hypothesis that CoDA has more representation learning capacity for modeling interactions between prototypes. This observation suggests that besides designing better multi-interest extractor, developing a novel method that effectively refines cluster prototypes is a promising direction to improve multi-interest modeling.

5.5 Qualitative Analysis of Interests (RQ4)

Uniformity Measure of User Representations. As *iterative latent attention* embeds personalization into item clusters, it naturally results in a higher level of discrimination of user's representations, which are built upon item clusters. We report *uniformity* [46] to

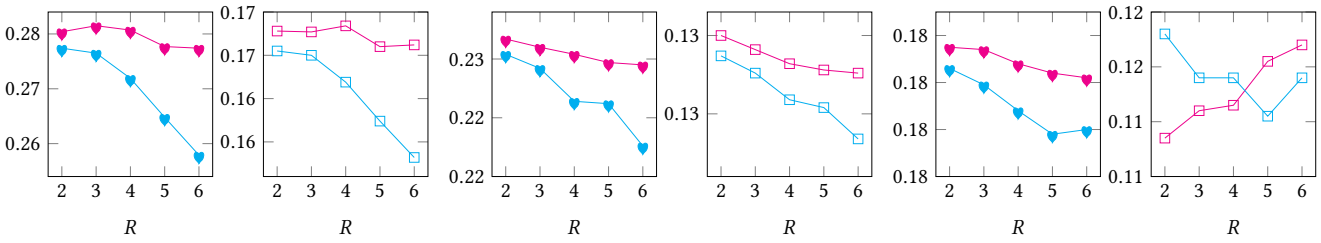
Table 2: Recommendation performance of all models. R@K and N@K stand for Recall at top K and NDCG at top K, respectively. We highlight the best result with bold face while the runner-up is underlined. ‡ denotes statistically significant number (p-value on paired t-test < 0.05).

Dataset	Metric	SimpleX	UltraGCN	RecVAE	NCL	SimGCL	DirectAU	DGCF	DPCML	MacridVAE	VALID
Citeulike-a	R@20	0.2607	0.2575	0.2398	0.2378	0.2444	0.2534	0.2066	0.2498	0.2744	**0.2804**‡
	N@20	0.1534	0.1497	0.1435	0.1353	0.1375	0.1412	0.1165	0.1429	0.1632	**0.1678**‡
	R@50	0.3857	0.3839	0.3481	0.3654	0.3706	0.3842	0.3348	0.3770	0.3974	**0.4098**‡
	N@50	0.1865	0.1835	0.1725	0.1688	0.1703	0.1751	0.1499	0.1759	0.1958	**0.2020**‡
Gowalla	R@20	0.2201	0.2186	0.2034	0.2145	0.2120	0.2005	0.1862	0.1915	0.2227	**0.2267**‡
	N@20	0.1289	0.1271	0.1188	0.1255	0.1243	0.1158	0.1085	0.1022	0.1282	**0.1300**‡
	R@50	0.3334	0.3341	0.3129	0.3292	0.3260	0.3129	0.2905	0.3191	0.3424	**0.3486**‡
	N@50	0.1566	0.1555	0.1456	0.1536	0.1523	0.1432	0.1339	0.1334	0.1577	**0.1599**‡
Yelp	R@20	0.1163	0.1114	0.1137	0.1551	0.1634	0.1677	0.1337	0.1125	0.1794	**0.1835**‡
	N@20	0.0611	0.0576	0.0596	0.0851	0.0929	0.0994	0.0713	0.0576	0.1134	**0.1147**
	R@50	0.2116	0.2065	0.2104	0.2624	0.2711	0.2763	0.2375	0.2094	0.2835	**0.2923**‡
	N@50	0.0857	0.0820	0.0844	0.1130	0.1207	0.1274	0.0981	0.0824	0.1405	**0.1430**‡



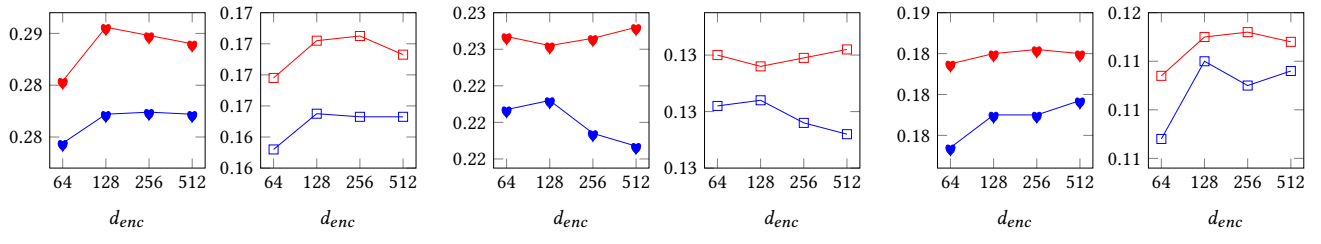
(a) Results on Citeulike-a (b) Results on Gowalla (c) Results on Yelp

Figure 2: Performance of personalized clustering VALID (red) w.r.t. number of interests K. We include performance of non-personalized clustering MacridVAE (blue) for contrasting. For VALID, we set $R = 2$. Heart symbols represent Recall@20 while square symbols represent NDCG@20.



(a) Results on Citeulike-a (b) Results on Gowalla (c) Results on Yelp

Figure 3: Effects of *Implicit Differentiation (ID)* in VALID. Magenta lines are VALID with ID while cyan lines are VALID without ID. We fix $K = 4$ and vary the number of clustering iterations R. Heart symbols represent Recall@20 while square symbols represent NDCG@20.



(a) Results on Citeulike-a (b) Results on Gowalla (c) Results on Yelp

Figure 4: VALID's performance (red lines) w.r.t. encoder hidden size d_{enc}. We include results of MacridVAE (blue lines) for contrasting. $K = 4$ is fixed in this experiments. Heart symbols represent Recall@20 while square symbols represent NDCG@20.

quantify the independence of user's representations w.r.t. number of clustering iterations R: $\mathcal{L}_{uniformity} = log \mathop{\mathbb{E}}_{i.i.d. x,y \sim p_{data}} e^{-2||x-y||_2^2}$. x, y are user representations produced by VALID, which is concatenation of K interest vectors then normalizing to unit length. As $\mathcal{L}_{uniformity}$ is negative, the lower it is, the higher level of independence between user's representations, i.e., they are uniformly distributed on unit hypersphere. Figure 5 clearly shows that increasing R results in lower uniformity, meaning that user's representations have higher level of discrimination, supporting our hypothesis.

Case Study on Multiple User's Interests. For an intuitive understanding of multiple interests of a user, we present a case study on Citeulike-a dataset. Table 4 presents the list of titles of articles with which a user has interacted. It is evident that this user is interested in multiple topics, e.g., *topic modeling*, *recommendation*,

Table 3: On Citeulike-a dataset, for each factor of VALID and MacridVAE, we present top 3 items with highest predicted score. Each item is an scientific article, described by its title.

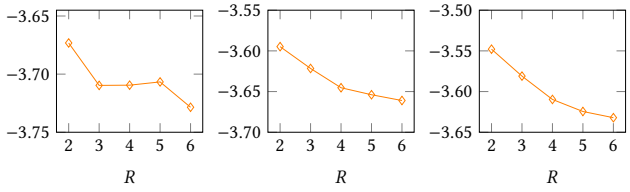
Model	Factor 1	Factor 2	Factor 3	Factor 4
VALID	**Information extraction** 1. Snowball: Extracting Relations from Large Plain-Text Collections 2. Automatic labeling of multinomial topic models 3. Open Information Extraction from the Web	**Topic Modeling** 1. Latent Dirichlet Allocation 2. Probabilistic Latent Semantic Analysis 3. Conditional random fields: Probabilistic models for segmenting and labeling sequence data	**Markov Model** 1. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms 2. Markov logic networks 3. An Introduction to Conditional Random Fields for Relational Learning	**Collaborative Filtering** 1. Amazon.com Recommendations: Item-to-Item Collaborative Filtering 2. Latent semantic models for collaborative filtering 3. Collaborative filtering with temporal dynamics
MacridVAE	**No specific topic** 1. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM 2. Snowball: Extracting Relations from Large Plain-Text Collections 3. Collaborative filtering with temporal dynamics	**Topic Modeling** 1. Probabilistic Latent Semantic Analysis 2. Latent Dirichlet Allocation 3. Automatic labeling of multinomial topic models	**Markov Model** 1. Markov logic networks 2. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms 3. Maximum Entropy Markov Models for Information Extraction and Segmentation	**Conditional Random Fields** 1. Dynamic conditional random fields 2. Conditional random fields: Probabilistic models for segmenting and labeling sequence data 3. An Introduction to ConditionalRandom Fields for Relational Learning

Table 4: Titles of interacted articles of a user on Citeulike-a.

List of interacted articles' titles of a user
1. Latent Dirichlet Allocation
2. Amazon.com Recommendations: Item-to-Item Collaborative Filtering
3. Markov logic networks
4. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms
5. Collaborative filtering with temporal dynamics
6. Snowball: Extracting Relations from Large Plain-Text Collections
7. Probabilistic Latent Semantic Analysis
8. Automatic labeling of multinomial topic models
9. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM
10. Dynamic conditional random fields

Table 5: Effect of refining cluster prototypes by self-attention. CoDA stands for Compositional De-Attention.

Dataset	Metric	No Self-Attention	Scaled Cosine Attention	CoDA
Citeulike-a	R@20	0.2804	**0.2821**	0.2816
	N@20	0.1678	**0.1687**	0.1683
	R@50	0.4098	0.4096	**0.4114**
	N@50	0.2020	0.2025	**0.2026**
Gowalla	R@20	**0.2267**	0.2248	0.2265
	N@20	0.1300	0.1290	**0.1302**
	R@50	0.3486	0.3461	**0.3488**
	N@50	0.1599	0.1587	**0.1601**
Yelp	R@20	0.1835	0.1824	**0.1836**
	N@20	0.1147	0.1124	**0.1148**
	R@50	0.2923	0.2920	**0.2923**
	N@50	0.1430	0.1408	**0.1431**



(a) Results on Citeulike-a (b) Results on Gowalla (c) Results on Yelp

Figure 5: Uniformity measures of user representations w.r.t. number of clustering iterations R. Lower is better.

information extraction and *Markov models*. In Table 3, for each factor produced by VALID and MacridVAE, we present the top 3 highest predicted scoring articles, each described by its title.

- VALID captures the diversity of user's preferences well. Each topic in Table 3 has correspondence to those of the user in Table 4.
- Although MacridVAE discovers two interests, e.g., *Markov models* and *topic modeling*, it does not highlight *collaborative filtering* as one interest. Instead, MacridVAE highlights *conditional random fields* as a topic, which is only a small aspect. Moreover, the first factor of MacridVAE is difficult to understand its meaning.

6 CONCLUSION

We analyze multi-interest recommendation models from clustering perspective to understand how they work and reveal their shortcomings. We then propose a novel VAE-based model called VALID with a couple of innovations. *Firstly*, it employs *iterative latent attention* to personalize item clustering, alleviating uninformative and non-personalized clustering of current works. *Secondly*, as iterative refinement method results in a deep network, to mitigate training difficulties, we propose to employ implicit differentiation. *Thirdly*, we study self-attention methods to refine multiple item clusters of a user, which opens a new research question for multi-interest modeling. We demonstrate the favorable performance of VALID on three real-world datasets from diverse sources. Qualitative analysis shows how VALID disentangles multiple interests of one user, more convincingly than those produced by the closest baseline MacridVAE.

ACKNOWLEDGMENTS

This research/project is supported by the Ministry of Education, Singapore under its Tertiary Education Research Fund (MOE Reference Number: MOE2021-TRF-013). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

REFERENCES

[1] Seyun Bae, Hoyoon Byun, Changdae Oh, Yoon-Sik Cho, and Kyungwoo Song. 2022. Graph Perceiver IO: A General Architecture for Graph Structured Data. *CoRR* abs/2209.06418 (2022).

[2] Shilong Bao, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. 2022. The Minority Matters: A Diversity-Promoting Collaborative Metric Learning Algorithm. In *NeurIPS*.

[3] Christian Bauckhage. 2015. Lecture Notes on Data Science: Soft k-Means Clustering.

[4] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *KDD*. 2942–2951.

[5] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. 2022. User-Aware Multi-Interest Learning for Candidate Matching in Recommenders. In *SIGIR*. 1326–1335.

[6] Michael Chang, Thomas L. Griffiths, and Sergey Levine. 2022. Object Representations as Fixed Points: Training Iterative Refinement Algorithms with Implicit Differentiation. In *NeurIPS*.

[7] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten De Rijke. 2021. Multi-Interest Diversification for End-to-End Sequential Recommendation. *ACM Trans. Inf. Syst.* 40, 1 (2021).

[8] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. 2021. Recurrent Independent Mechanisms. In *International Conference on Learning Representations*.

[9] Zhiqiang Guo, Guohui Li, Jianjun Li, and Huaicong Chen. 2022. TopicVAE: Topic-Aware Disentanglement Representation Learning for Enhanced Recommendation. In *ACM MM*. 511–520.

[10] Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, Hannah Sheahan, Neil Zeghidour, Jean-Baptiste Alayrac, Joao Carreira, and Jesse Engel. 2022. General-purpose, long-context autoregressive modeling with Perceiver AR. In *ICML*. 8535–8558.

[11] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. 2022. Perceiver IO: A General Architecture for Structured Inputs & Outputs. In *ICLR*.

[12] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General Perception with Iterative Attention. In *Proceedings of the 38th International Conference on Machine Learning*. 4651–4664.

[13] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations*.

[14] Daeryong Kim and Bongwon Suh. 2019. Enhancing VAEs for Collaborative Filtering: Flexible Priors & Gating Mechanisms. In *RecSys*. 403–407.

[15] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. 2022. Conditional Object-Centric Learning from Video. In *ICLR*.

[16] Beibei Li, Beihong Jin, Jiageng Song, Yisong Yu, Yiyuan Zheng, and Wei Zhou. 2022. Improving Micro-Video Recommendation via Contrastive Multiple Interests. In *SIGIR*. 2377–2381.

[17] 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 *CIKM*. 2615–2623.

[18] Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. MINER: Multi-Interest Matching Network for News Recommendation. In *Findings of ACL*. 343–352.

[19] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. 689–698.

[20] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *International Conference on Learning Representations*.

[21] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-Enriched Contrastive Learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.

[22] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-Centric Learning with Slot Attention. In *NeurIPS*, Vol. 33.

[23] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*.

[24] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM*. 1243–1252.

[25] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM*. 1253–1262.

[26] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with β-VAE. In *CIKM*. 1356–1365.

[27] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing between Capsules. In *NeurIPS*. 3859–3869.

[28] Aghiles Salah, Thanh Binh Tran, and Hady Lauw. 2021. Towards Source-Aligned Variational Models for Cross-Domain Recommendation. In *RecSys*. 176–186.

[29] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *WSDM*. 528–536.

[30] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *CoRL*.

[31] Caiqi Sun, Penghao Lu, Lei Cheng, Zhenfu Cao, Xiaolei Dong, Yili Tang, Jun Zhou, and Linjian Mo. 2022. Multi-interest Sequence Modeling for Recommendation with Causal Embedding. In *SDM*. 406–414.

[32] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. 2021. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?. In *Fifteenth ACM Conference on Recommender Systems*. 708–713.

[33] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-Interest Network for Sequential Recommendation. In *WSDM*. 598–606.

[34] Y. Tan, C. Yang, X. Wei, Y. Ma, and X. Zheng. 2021. Multi-Facet Recommender Networks with Spherical Optimization. In *ICDE*. 1524–1535.

[35] Zineng Tang, Jaemin Cho, Jie Lei, and Mohit Bansal. 2023. Perceiver-VL: Efficient Vision-and-Language Modeling with Iterative Latent Attention. In *WACV*.

[36] Yi Tay, Anh Tuan Luu, Aston Zhang, Shuohang Wang, and Siu Cheung Hui. 2019. Compositional De-Attention Networks. In *NeurIPS*, Vol. 32.

[37] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. 2022. When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation. In *SIGIR*. 1632–1641.

[38] Nhu-Thuat Tran and Hady W. Lauw. 2022. Aligning Dual Disentangled User Representations from Ratings and Textual Content. In *KDD*. 1798–1806.

[39] Quoc-Tuan Truong, Aghiles Salah, and Hady W. Lauw. 2021. Bilateral Variational Autoencoder for Collaborative Filtering. In *WSDM*. 292–300.

[40] Yao-Hung Hubert Tsai, Nitish Srivastava, Hanlin Goh, and Ruslan Salakhutdinov. 2020. Capsules with Inverted Dot-Product Attention Routing. In *ICLR*.

[41] Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussá. 2022. Efficient Speech Translation with Dynamic Latent Perceivers. https://arxiv.org/abs/2210.16264

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[43] Chenyang Wang, Zhefan Wang, Yankai Liu, Yang Ge, Weizhi Ma, Min Zhang, Yiqun Liu, Junlan Feng, Chao Deng, and Shaoping Ma. 2022. Target Interest Distillation for Multi-Interest Recommendation. In *CIKM*. 2007–2016.

[44] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *KDD*. 1816–1825.

[45] Shicheng Wang, Shu Guo, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2022. Multi-Interest Extraction Joint with Contrastive Learning for News Recommendation. In *ECML-PKDD*.

[46] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*.

[47] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2023. Disentangled Representation Learning for Recommendation. *IEEE TPAMI* 45, 1 (2023), 408–424.

[48] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. 1001–1010.

[49] Zhikai Wang and Yanyan Shen. 2022. Time-aware Multi-interest Capsule Network for Sequential Recommendation. In *SDM*. 558–566.

[50] Jason Weston, Ron J. Weiss, and Hector Yee. 2013. Nonlinear Latent Factorization by Embedding Multiple User Interests. In *ACM RecSys*. 65–68.

[51] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and Contrastive Variational Autoencoder for Sequential Recommendation. In *Proceedings of the Web Conference 2021*. 449–459.

[52] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *SIGIR*. 1294–1303.

[53] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, and Fei Wu. 2022. Re4: Learning to Re-Contrast, Re-Attend, Re-Construct for Multi-Interest Recommendation. In *The Web Conference*. 2216–2226.

[54] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR*. 367–377.

[55] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM*. 2329–2332.