Collaborative Reflection-Augmented Autoencoder Network for Recommender Systems

LIANGHAO XIA, South China University of Technology
CHAO HUANG, University of Hong Kong
YONG XU and HUANCE XU, South China University of Technology
XIANG LI, Australian National University
WEIGUO ZHANG, South China University of Technology

As the deep learning techniques have expanded to real-world recommendation tasks, many deep neural network based Collaborative Filtering (CF) models have been developed to project user-item interactions into latent feature space, based on various neural architectures, such as multi-layer perceptron, autoencoder, and graph neural networks. However, the majority of existing collaborative filtering systems are not well designed to handle missing data. Particularly, in order to inject the negative signals in the training phase, these solutions largely rely on negative sampling from unobserved user-item interactions and simply treating them as negative instances, which brings the recommendation performance degradation. To address the issues, we develop a Collaborative Reflection-Augmented Autoencoder Network (CRANet), that is capable of exploring transferable knowledge from observed and unobserved user-item interactions. The network architecture of CRANet is formed of an integrative structure with a reflective receptor network and an information fusion autoencoder module, which endows our recommendation framework with the ability of encoding implicit user's pairwise preference on both interacted and non-interacted items. Additionally, a parametric regularization-based tied-weight scheme is designed to perform robust joint training of the two-stage CRANet model. We finally experimentally validate CRANet on four diverse benchmark datasets corresponding to two recommendation tasks, to show that debiasing the negative signals of user-item interactions improves the performance as compared to various state-of-the-art recommendation techniques. Our source code is available at https://github.com/akaxlh/CRANet.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Recommender Systems, Collaborative Filtering, Autoencoder

1 INTRODUCTION

The past decades have witnessed the explosive growth of online services, e.g., social media, e-commerce, and online advertising, which has provided overwhelming choices for users [15, 29, 54]. To accurately characterize users' preferences and identify their interested items, personalized recommendation has become an important component in many customer-oriented applications to predict user behavior [13, 23]. Among various recommendation methods, **Collaborative Filtering** (**CF**) has become a key technique to infer the user's interests from not only his/her past interaction behavior, but also the behavior from other individuals [3, 7]. At the core of collaborative filtering methods is how to parameterize users and items with effective feature vectors based on their observed historical interactions [47].

With the advent of deep learning techniques, significant efforts have been made to develop neural collaborative filtering models and show performance superiority over conventional matrix factorization techniques [51]. For example, **Multi-layer perceptron** (**MLP**) has been utilized to capture non-linear user-item interactions [14]. Autoencoder architecture has served as an effective solution for dimensionality reduction by learning underlying patterns from the interaction data between users and items [6, 33, 34]. Furthermore, several graph neural network models aim at modelling relational information over the user-item interaction graph via different aggregation functions, such as graph convolution operation [38, 52], and attention-based message passing frameworks [32, 44].

To address the above limitation, this work proposes to learn complex user's preference from not only his/her positive feedback (e.g., click or visit), but also the knowledge of his/her negative and unobserved interactions with items in a explicit way. However, there exist several key technical challenges that remain to be solved for realizing the high potential predictability of user's preference with the collectively modeling of positive and unobserved user-item interactions. *First*, different types (i.e., positive, negative, and unobserved) of user-item interactions interweave with each other in complex ways, how to effectively encode cross-modal collaborative signals, remains a significant challenge. *Second*, due to the fact that the negative and unobserved instances dominate the user-item interactions, it is crucial to alleviate such overfitting phenomenon when encoding the different-dimensional relational structures between different users and items. *Third*, the sparsity degrees of interaction data vary by users, performing robust representation learning on different users and items is a key factor for effective recommendations.

With motivations and challenges above, this article develops a **Collaborative Reflection-Augmented Autoencoder Network** (**CRANet**) to improve the performance of CF for recommendations, with the exploration of the observed and unobserved user-item interactions. The proposed CRANet captures the implicit users' preference from their unobserved interactions over items with two main model designs: (i) we first design an integrative neural architecture with a reflective reception network and a tied-weight gated scheme, to encode the preference of users over their non-interacted items based on backward embedding propagation paradigm. With the reflection-based imputation scheme, we inject the collaborative signals from unobserved interactions into the user representation phase; (ii) additionally, our information fusion framework is built upon the autoencoder model to capture the latent dependency between observed positive user-item interactions and unobserved ones. With the reconstruction-guided learning objective, the interaction dependent structures are encoded through the stacked multiple projection layers in the modeling of collaborative effects. With the cooperation of the reflection receptor network and an autoencoder architecture, we promote the collaboration of cross-modal user-item interactions to characterize user's interests. In addition, a regularization-based tied-weight scheme is

introduced to yield a robust user-item interaction embedding learning under data imbalance, and offer efficient training for the autoencoder-based reflection network.

The main contributions of this work are listed as follows:

— We highlight the benefits of exploiting cross-modal user-item interactions, to allow the effectively joint modeling of both observed and unobserved interactive behavior of users.

— We propose a two-stage solution, CRANet, integrating a simple but effective reflective receptor network with an information fusion autoencoder architecture, which encodes the collaborative signal in the form of observed and unobserved interactions.

— Furthermore, another contribution is the effective design of a regularization-based tied weight scheme in our CRANet framework, which can alleviate the overfitting issue and offer robust embedding learning over user-item interactions by correcting the sparsity bias. A theoretical analysis is provided to show the rationality of our designed hierarchically structured gate-based reflective receptor framework in capturing user-item relations.

— We perform extensive experiments on several publicly accessible datasets corresponding to two recommendation tasks (i.e., rating prediction and item, recommendation), and compare our CRANet framework with various state-of-the-art baselines. The model ablation study verifies the efficacy of our developed modules, and the utility of reflection mechanism in encoding implicit user preferences. In addition, we also investigate the model scalability and compare CRANet with state-of-the-art recommender systems, to show its efficiency.

The rest of this article is organized as follows. We discuss the related work in Section 2. The technical details of our developed CRANet framework in Section 3. We also perform theoretical analysis of our model. Section 4 presents the evaluation results of our CRANet method as compared to state-of-the-art baselines. We finally conclude this work in Section 5.

2 RELATED WORK

2.1 Neural Network-Based Recommender Systems

Many efforts have focused on proposing recommendation models with different types of neural networks [2, 4, 11, 43], such as neural network-augmented recommendation via stacking several feed-forward layers [12, 14] and neural auto-regressive learning frameworks [6, 55]. More recently, several graph neural networks (e.g., graph convolutional network) have been applied to recommendation tasks to model the graph-structured relations between users and items [38, 41, 52]. Different from these state-of-the-art solutions, this work aims at promoting the cooperation among user-item relations and significantly improves the recommendation performance in terms of both effectiveness and efficiency, with the cross-modal reflective autoencoder network and efficient reflection mechanism.

2.2 Relation-Aware Recommendation Models

There exist many research work aiming to investigate various relationships among users and items for recommendation applications. For example, for user side, social relations between users have been considered as the side information to improve the recommendation performance [9, 16, 39]. For item, side, knowledge graph serve as another key source to model relationships across different items [10, 37]. In addition, multiple explicit item, relation modeling [46] and user behavior learning [45] have been integrated with the collaborative filtering with graph neural networks. Different from these work focusing on exploring explicit relations between users and items, this work
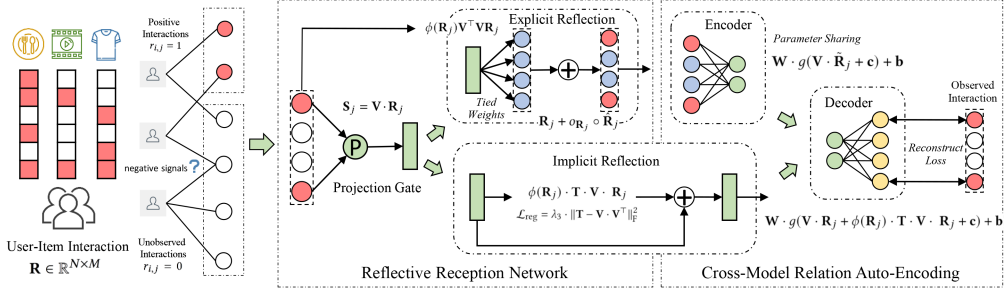
Fig. 1. Model architecture of the proposed CRANet. Explicit reflective reception network (i.e., utilizing $\mathbf{V}^{\top}$ to infer the missing interaction scores as defined by Equation (1) and Equation (3)), and implicit reflection (i.e., employing $\mathbf{T}$ to replace $\mathbf{V}^{\top}$ in the reflection and the encoder in the cross-model auto-encoding as defined by Equation (4) and Equation (5)).

explores the implicit user preference from unobserved user-item relations without any auxiliary information.

2.3 Autoencoder Collaborative Filtering Framework

Autoencoder has emerged as an important architecture to enable the collaborative filtering techniques to map the explicit user-item interactions into latent low-dimensional representations, with the objective of reconstructing the original inputs [28]. For instance, AutoRec [31] is the first AE-based collaborative filtering method that shows noticeably superior performance over the conventional recommendation models. Motivated by this work, many extensions have been developed to solve the collaborative filtering challenges (e.g., non-linear latent-variable modeling or data sparsity) with various autoencoder variants, such as variational AE [24, 40], contractive AE [53], and denoising AE [42]. However, most of these autoencoder modules require sufficient user-item interactions to learn accurate user/item embeddings. To fill this gap, our CRANet enables a practical learning scenario to capture implicit user-item relations with a designed cross-modal autoencoder network.

2.4 Imputation-Based Recommendation Methods

Another research line relevant to this work lies in the recommendation methods based on the data imputation techniques. For example, the early study S-Imp [26] proposes a regularized low-rank SVD method to replace the missing elements with derived values. With the advance of deep learning techniques, Lee et al. [22] develops a neural network-based approach to use the inferred imputed values for learning user preference. Wan et al. [35] designs an ensemble learning to perform the continuous imputation and optimize the neural matrix factorization framework. Furthermore, a zero imputation algorithm is introduced to handle missing data with sparsity normalization [50]. Along with this line and motivated by these work, we develop a reflection-augmented autoencoder network to capture implicit user preference with an effective imputation mechanism.

3 METHODOLOGY

In this section, we describe the technical details of our CRANet framework (the model architecture is shown in Figure 1). In particular, we first introduce key definitions and preliminaries. There are two key modules in the CRANet framework corresponding to our motivations and challenges: (i) the reflective reception network that captures users' implicit preference on their non-interacted items, with the backward embedding propagation paradigm; (ii) a collaborative dependency

Table 1. Summary of Key Notations

Notation	Description
u_i	the ith user ($1 \leq i \leq M$)
t_j	the jth item, ($1 \leq j \leq N$)
$\mathbf{R}_i \in \mathbb{R}^N$	the interaction vector of user u_i over different items
$\mathbf{R}_j \in \mathbb{R}^M$	the interaction vector of user t_j with different user
α	hyperparameter which controls the decay strength
$\phi(\cdot)$	decay function which is reflective of energy dissipation in our reflection gate
$o_{\mathbf{R}_j}, \circ$	zero indicator and element-wise product in our reflection gate
$\mathbf{V}, \mathbf{V}^\top$	the projection weight matrix and its corresponding transpose form for reflective gating mechanism
$\tilde{\mathbf{R}}$	the recalibrated relation-aware user-item interactions
$\mathbf{U} \in \mathbb{R}^{N \times d_p}$	learnable transformation matrix in our tied-weight scheme for training efficiency
$\mathbf{W}, \mathbf{V}, \mathbf{c}, \mathbf{b}$	transformatino matrices and bias terms of user-item relation auto-encoding
$\mathcal{L}_{\text{reg}}$	regularization term which simulates the reflection process in encoding the implicit user-item relations

auto-encoding component which aggregates the information from different user-item relational signals. Finally, we offer a theoretical analysis for our designed encoding function of complex relations between user and item, and discuss the model complexity.

3.1 Preliminaries

We consider a scenario with M items and N users. Their interaction matrix is denoted by $\mathbf{R} \in \mathbb{R}^{N \times M}$, where the corresponding interaction (e.g., explicit user rating) between the user u_i ($1 \leq i \leq N$) and item, t_j ($1 \leq j \leq M$) is denoted as $r_{i,j}$ ($r_{i,j}$ is set as 0 if there is no observed interaction). In the practical recommendation scenario, the data scarcity of user-item interactions poses challenges in learning accurate user preferences. However, the interaction value $r_{i,j} = 0$ does not necessarily indicate that user u_i does not like item, t_j, it is possible that the user is not aware of that item. With the observed interaction matrix $\mathbf{R}$, the objective of this work aims at learning user's interests over items, the result of our learning framework is expected to predict the unknown values in $\mathbf{R}$, with the exploration of observed and unobserved interactive relations across users and items to address the sparsity bias challenge.

3.2 The Reflective Reception Network

3.2.1 Projection Gate for Latent Embedding Learning. We first introduce the projection gate in our reflective reception network to map the user-item interaction data into the latent learning space. We denote the interaction vector of user u_i and t_j as $\mathbf{R}_i \in \mathbb{R}^N$ ($[r_{i,1}, \ldots, r_{i,N}]$) and $\mathbf{R}_j \in \mathbb{R}^M$ ($[r_{1,j}, \ldots, r_{M,j}]$), respectively. Without loss of generality, each column of $\mathbf{R}$ is denoted as the interaction vector $\mathbf{R}_j$ from the item, perspective as the concrete example of vector dimension setting. Our framework is flexible to handle user's interaction input $\mathbf{R}_i$ in a similar way. The projection gate operation is formally defined as: $\mathbf{S}_j = \mathbf{V} \cdot \mathbf{R}_j$, where $\mathbf{V} \in \mathbb{R}^{d_p \times N}$ is the learned transformation matrix and d_p denotes the hidden state dimensionality ($d_p << N$). $\mathbf{S}_j$ is the latent representation of sparse user-item interaction data $\mathbf{R}_j$.

3.2.2 Reflection Gate for Interaction Encoding. To explore the different user-item relations and alleviate the variable sparsity bias, we further design a reflection gate to propagate the learned latent embeddings back to the original data space, which explicitly learns the unknown user-item pairwise preference scores in an effective manner. To achieve this goal, we first define a decay function $\phi(\cdot)$ which is analogous to the energy dissipation in the mirror reflection phenomenon to prevent the imputed preference scores learned from the reflection gate dominating the user-item interaction data. Given the interaction vector $\mathbf{R}_j$, we formally present the decay function as: $\phi(\mathbf{R}_j) = \frac{\alpha}{\|\mathbf{R}_j\|_0}$, where α is a hyperparameter controlling the decay strength and $\|\mathbf{R}_j\|_0$ represents

the number of observed interactions. It is worth mentioning that the decay function normalizes the energy of the reflected inputs using ℓ_0 norm, as the sparsity degree varies by users/items causing the energy gaps in collaborative filtering architecture. The numerator can be substituted with other functions (e.g., $\log\left(\|\mathbf{R}_j\|_0 + 1\right)$ and $\sqrt{\|\mathbf{R}_j\|_0}$), to adjust how the reflection energy is normalized. In experiments, we will compare the performance of using the above decay functions. Based on the above definitions, the reflection gate operation can be formally given as follows:

$$\hat{\mathbf{R}}_j = \phi(\mathbf{R}_j) \cdot \mathbf{V}^\top \cdot \mathbf{S}_j = \phi(\mathbf{R}_j) \cdot \mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j, \tag{1}$$

where the projection weight matrix is denoted by $\mathbf{V}^\top$, and $\hat{\mathbf{R}}_j$ contains inferred quantitative preferences ($\hat{r}_{i,j}$ given $r_{i,j} = 0$). Then we combine the inferred scores with the observed interactions:

$$\tilde{\mathbf{R}}_j = h_1(\mathbf{R}_j; \{\mathbf{V}\}) = \mathbf{R}_j + o_{\mathbf{R}_j} \circ \hat{\mathbf{R}}_j, \tag{2}$$

where $o_{\mathbf{R}_j}$ is the zero indicator vector in which the element is set as 0 given the observed user-item interaction. $\circ$ denotes element-wise product operation. Note that the higher the learned preference score $\hat{r}_{i,j}$ is, the more likely that user u_i is interested in item, t_j. The value of $\hat{r}_{i,j}$ is inferred with a smaller scale. The self-augmented interaction matrix $\tilde{\mathbf{R}}$ not only preserves the original observed user-item interactions, but also encodes the relations between users and their non-interacted items. With the cooperation of reflection mechanism and decay factor, CRANet alleviates the data imbalance issue, i.e., dense interaction vectors often have higher energy than the sparse ones, by generating the reflected vectors with the uniform energy distribution regardless of the variable sparsity degrees.

3.2.3 Tied-Weight Scheme for Training Efficiency. In the reflective reception network, we integrate the projection and reflection gate with the tied-weight mechanism [18, 25], i.e., the transformation operations are designed as transpose matrices. With the dedicated parameter sharing schema, the incorporated parameter learning constrain could effectively address the variable sparsity issue [55]. Such a learning scheme could improve the efficiency of CRANet by reducing its training cost and serve as an effective way of regularization to alleviate overfitting. In Section 4, we perform ablation study by replacing $\mathbf{V}^\top$ with another learnable transformation matrix $\mathbf{U} \in \mathbb{R}^{N \times d_P}$, to show the effectiveness of our designed tied-weight scheme in CRANet framework.

3.3 Collaborative Dependency Auto-Encoding

To capture the inter-dependencies among different types of interactive relations (i.e., observed positive user-item interactions and learned unknown user-item interactions), we augment our CRANet framework with an information fusion autoencoder architecture. The developed autoencoder model endows the CRANet with the ability of fusing collaborative signals by learning the reconstruction of the recalibrated relation-aware user-item interactions $\tilde{\mathbf{R}}$. In the process of our collaborative signal fusion, we first design an encoder to map the interaction matrix $\tilde{\mathbf{R}}$ into another low-dimensional latent space which preserves the most informative signals across different relations. Following the encoder module, the decoder component aims at reconstructing $\tilde{\mathbf{R}}$ from the encoded representation to be as close to the original input. The dimension of the output layer in our autoencoder network is the same as that of the input layer. The interaction relation fusion network is trained by minimizing the reconstruction loss between the original input and the corresponding reconstructed interactions, and jointly optimized with the reflective reception network.

We configure our collaborative dependency auto-encoding framework with a three-layer autoencoder framework, in which the predictive function is defined as follows:

$$h_2\left(\tilde{\mathbf{R}}_j; \{\mathbf{W}, \mathbf{V}, \mathbf{c}, \mathbf{b}\}\right) = \mathbf{W} \cdot g\left(\mathbf{V} \cdot \tilde{\mathbf{R}}_j + \mathbf{c}\right) + \mathbf{b}, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{N \times d_p}$, $\mathbf{V} \in \mathbb{R}^{d_p \times N}$ are the transformation matrices, $\mathbf{c} \in \mathbb{R}^{d_p}$, $\mathbf{b} \in \mathbb{R}^N$ are the bias vectors, and $g(\cdot)$ is the sigmoid function such that $g(\mathbf{x})(i) = g(\mathbf{x}(i)), g(x) = \frac{1}{1+e^{-x}}$. Note that for the purpose of reducing models' parameters and achieving model efficiency, we conduct the parameter sharing mechanism by using the same transformation $\mathbf{V}$ as the projection matrix in the hidden layer of the autoencoder network.

Parametric Regularized Reflection Process. Furthermore, to correct the data imbalance bias with the flexibility in handling user's interaction vectors with various sparsity degrees, we further augment our CRANet by introducing a new parametric matrix $\mathbf{T} \in \mathbb{R}^{d_p \times d_p}$ and update the predictive function of CRANet as follows:

$$H(\mathbf{R}_j; \Theta) = \mathbf{W} \cdot g(\mathbf{V} \cdot \mathbf{R}_j + \phi(\mathbf{R}_j) \cdot \mathbf{T} \cdot \mathbf{V} \cdot \mathbf{R}_j + \mathbf{c}) + \mathbf{b}, \tag{4}$$

where $\mathbf{T}$ replaces $\mathbf{V}$ from the collaborative dependency autoencoder and $\mathbf{V}^\top$ from the reflective reception network. Then the following regularization term is incorporated in the loss function as follows:

$$\mathcal{L}_{\text{reg}} = \lambda_3 \cdot \|\mathbf{T} - \mathbf{V} \cdot \mathbf{V}^\top\|_{\text{F}}^2. \tag{5}$$

The above regularization term guides the learning process of CRANet by simulating the reflection process in encoding the implicit user-item relations. Moreover, this design has another advantage in further optimizing the model training efficiency, i.e., $\mathbf{T}$ is of the size $d_p \times d_p$ which is smaller than $d_p \times N$ and $N \times N$. Due to $d_p << N$, it reduces two times of large-matrix multiplications to one-time small-matrix multiplication, which is more computationally efficient.

3.4 Implicit Relation Learning With Reflective Autoencoder Framework

With the integration of the reflective reception network and collaborative dependency autoencoder module, CRANet draws on the latent factor modeling, by transferring the positive user-item interactive signals to encode the user's implicit preferences on all other unobserved items. By incorporating the decay factor into reflective autoencoder architecture, the imputed user's preference values has much smaller magnitude as compared to the original user-item interactions (e.g., , $\hat{r}_{i,j}$ might be around 0.1 in a five-star rating system). Such an energy decay mechanism could prevent the dominating phenomenon caused by the imputed values, which misleads the prediction.

Different from the stacked autoencoder models [17, 36], our CRANet is a hierarchical framework with the designed reflective reception mechanism, which enables the parameter sharing and greatly reduces the model size. This model refining may decrease the difficulty of training and get rid of the greedy pre-train strategy, which has been applied in stacked autoencoder network [31], to achieve the model efficiency. In addition, to further improve the model training performance and address the vanishing gradient problem, we configure CRANet with the residual connection strategy from the input to the output layer of CRANet.

3.5 Model Optimization

Algorithm 1 presents the learning process of CRANet. Given the predictive function defined in Equation (4), we perform the jointly training for both the reflective reception network and collaborative dependency autoencoder module with the following loss function:

$$\min_{\Theta} \sum_{\mathbf{R}_j \in \mathbb{T}} \|(\mathbf{1} - o_{\mathbf{R}_j}) \circ (H(\mathbf{R}_j; \Theta) - \mathbf{R}_j)\|_2^2 + \lambda_1 \cdot \|\mathbf{V}\|_{\text{F}}^2 + \lambda_2 \cdot \|\mathbf{W}\|_{\text{F}}^2 + \lambda_3 \cdot \|\mathbf{T} - \mathbf{V} \cdot \mathbf{V}^\top\|_{\text{F}}^2, \tag{6}$$

where $\mathbb{T}$ denotes the training set, and $\mathbf{1} = [1, \ldots, 1]^\top \in \mathbb{R}^N$. The first component of loss function ($\|(\mathbf{1} - o_{\mathbf{R}}) \circ (H(\mathbf{R}; \Theta) - \mathbf{R})\|_2^2$) is the reconstruction term which is based on the observed user-item interactions. Hence, there is no restriction on the originally unobserved ratings, even they can be

---

ALGORITHM 1: Learning Process of CRANet Framework

    Input: user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{N \times M}$, maximum epoch number E, regularization weights $\lambda_1, \lambda_2, \lambda_3$,
        learning rate η
    Output: trained parameters in Θ
1 Initialize all parameters in Θ
2 **for** $e = 1$ *to* E **do**
3   Draw a mini-batch $\mathcal{I}$ from all items $\{1, 2, \ldots, M\}$
4   Loss $= \lambda_1 \cdot \|\mathbf{V}\|_{\mathrm{F}}^2 + \lambda_2 \cdot \|\mathbf{W}\|_{\mathrm{F}}^2 + \lambda_3 \cdot \|\mathbf{T} - \mathbf{V} \cdot \mathbf{V}^\top\|_{\mathrm{F}}^2$
5   **for** *each* $t_j \in \mathcal{I}$ **do**
6   Compute $H(\mathbf{R}_j; \Theta)$ according to Equation (4)
7         $Loss+ = \|(1 - o_{\mathbf{R}_j}) \circ (H(\mathbf{R}_j; \Theta) - \mathbf{R}_j)\|_2^2$
8   **end**
9   **for** $V, W, T \in \Theta$ **do**
10        $\mathbf{V} = \mathbf{V} - \eta \cdot \partial\text{Loss}/\partial\mathbf{V}$
11        $\mathbf{W} = \mathbf{W} - \eta \cdot \partial\text{Loss}/\partial\mathbf{W}$
12        $\mathbf{T} = \mathbf{T} - \eta \cdot \partial\text{Loss}/\partial\mathbf{T}$
13   **end**
14 **end**
15 return all parameters Θ

---

filled with the implicit user-item preference scores learned by the reflective reception network as defined in Equation (1). The second component contains two weight decay regularization terms on $\mathbf{V}, \mathbf{W}$ and the reflective loss defined by Equation (5). Scalars $\lambda_1, \lambda_2, \lambda_3$ are defined to balance the three terms. We apply the Adam optimizer to infer the model parameters.

3.6 Theoretical Analysis

Our reflection receptor simulates the energy propagation. The benefit of our reflection mechanism can be interpreted from two perspectives: first, we compare the cosine similarity between the observed ratings and the imputed values in the latent space, and find that by adopting the transpose matrix $\mathbf{V}^\top$ for the reflection process, the similarity is guaranteed to be positive. In other words, we prove that the similarity between the original data and the imputed data given by our approach, has a lower bound. Second, by comparing the transpose $\mathbf{V}^\top$ with an independent variable $\mathbf{U}$, we find that the equivalent solutions of our approach are the set of orthogonal matrices, and that of the trivial method are the invertible matrices. This makes the solutions to our method much less ambiguous and more stable.

One reasonable criterion on the implicit relation encoding of CRANet is that the imputed values $h_1(\mathbf{R}_j; \Theta_1)$ should be similar and consistent with the originally observed ratings $\mathbf{R}_j$ [48]. Particularly, the consistency between $h_1(\mathbf{R}_j; \Theta_1)$ and $\mathbf{R}_j$ can be measured by the similarity between their projected latent representations, i.e., $\mathbf{V} \cdot h_1(\mathbf{R}_j; \Theta_1)$ and $\mathbf{V} \cdot \mathbf{R}_j$. To justify the rationality of reflection mechanism in CRANet, we first replace the transformation $\mathbf{V}^\top$ with its counterpart $\mathbf{U}$. Then we formally estimate the correlation between $\mathbf{V} \cdot h_1(\mathbf{R}_j; \Theta_1)$ and $\mathbf{V} \cdot \mathbf{R}_j$ based on cosine similarity:

$$\cos(\mathbf{V} \cdot \mathbf{R}_j, \mathbf{V} \cdot h_1(\mathbf{R}_j; \Theta_1)) = \cos(\mathbf{V} \cdot \mathbf{R}_j, \phi(\mathbf{R}_j) \cdot \mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j) = \cos(\mathbf{V} \cdot \mathbf{R}_j, \mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j)$$

$$= \frac{(\mathbf{R}_j^\top \cdot \mathbf{V}^\top)(\mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j)}{\|\mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2} = \frac{(\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j)^\top (\mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j)}{\|\mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2}$$

$$= \frac{\|\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2}{\|\mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2} \cdot \cos(\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j, \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j).$$

$$(7)$$

Since $\frac{\|\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2}{\|\mathbf{V} \cdot \mathbf{R}_j\|_2 \cdot \|\mathbf{V} \cdot \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j\|_2}$ is non-negative, the sign in Equation (7) is dependent on the sign of $\cos(\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j, \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j)$. For a free-form $\mathbf{U}$, it is likely that $\cos(\mathbf{V} \cdot^\top \mathbf{V} \cdot \mathbf{R}_j, \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j)$ is negative which makes $\mathbf{V} \cdot \mathbf{R}_j$ and $\mathbf{V} \cdot h_1(\mathbf{R}_j; \Theta_1)$ opposite to each other. In contrast, by setting the transformation $\mathbf{U} = \mathbf{V}^\top$ with the tied-weight scheme, we have

$$\cos(\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j, \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{R}_j) = \cos(\mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j, \mathbf{V}^\top \cdot \mathbf{V} \cdot \mathbf{R}_j) = 1. \tag{8}$$

Then, the cosine similarity in Equation (7) is guaranteed to be positive and thus the reflected vectors do not contain the components which are opposite to the original user-item interaction matrix.

The benefit of using $\mathbf{V}^\top$ instead of $\mathbf{U}$ can also be interpreted from the perspective of solution's ambiguity. Suppose $(\mathbf{U}^*, \mathbf{V}^*)$ is one solution for CRANet, $(\mathbf{U}^*\mathbf{B}^{-1}, \mathbf{B} \cdot \mathbf{V}^*)$ is an equivalent solution based on the following formal operations:

$$h_1(\mathbf{R}_j; \{\mathbf{U}^*, \mathbf{V}^*\}) = \phi(\mathbf{R}_j) \cdot \mathbf{U}^*\mathbf{V}^*\mathbf{R}_j = \phi(\mathbf{R}_j) \cdot \mathbf{U}^*\mathbf{B}^{-1} \cdot \mathbf{B}\mathbf{V}^*\mathbf{R}_j = h_1(\mathbf{R}_j; \{\mathbf{U}^*\mathbf{B}^{-1}, \mathbf{B}\mathbf{V}^*\}). \tag{9}$$

Thus, we have

$$\mathbb{S}_1 = \{(\mathbf{U}^*\mathbf{B}^{-1}, \mathbf{B}\mathbf{V}^*) : \mathbf{B}^{-1} \text{ exists}\}; \mathbb{S}_2 = \{(\mathbf{V}^{*\top}\mathbf{B}^\top, \mathbf{B}\mathbf{V}^*) : \mathbf{B}\mathbf{B}^\top = \mathbf{B}^\top\mathbf{B} = \mathbf{I}\}. \tag{10}$$

$\mathbb{S}_1$ is a set of equivalent solutions for CRANet model. We consider $\mathbf{U} = \mathbf{V}^\top$ and suppose $(\mathbf{V}^{*\top}, \mathbf{V}^*)$ is one solution. Then the corresponding set of equivalent solutions $\mathbb{S}_2$ is also given above.

Note that $\|\mathbb{S}_1\|$ is much larger than $\|\mathbb{S}_2\|$, as orthogonal matrices are only a small subset of invertible matrices. In other words, replacing $\mathbf{V}^\top$ by $\mathbf{U}$ in CRANet would increase the ambiguity of solution of the model. In addition, an invertible matrix may have poor condition so that some solutions in $\mathbb{S}_1$ affect the stability of the module, while any orthogonal matrices are well-posed so that any solution from $\mathbb{S}_2$ is equivalently stable for the network.

3.7 Complexity Analysis of the CRANet Framework

As discussed in Section 3.4, although the proposed CRANet model seems to improve the performance by introducing more transformation terms, most of the parameters in the reflection mechanism are shared by the existing parameters of autoencoder. Specifically, the number of parameters of CRANet with tied weights is $2N \times d_p$ which is the same as that of a three-layer autoencoder, and the model size of CRANet with implicit reflection (i.e., utilizing $\mathbf{T}$ with $\mathcal{L}_{\text{reg}}$) is $(2N \times d_p + d_p \times d_p)$ which costs moderate extra space considering $d_p << N$. As for the computational cost, $2N \times d_p$ and $d_p \times d_p$ of extra computations are required, compared to the cost of $2N \times d_P$ with a three-layer autoencoder. In conclude, the reflection mechanism does not change the order of both the time and the space complexity when it is built into an autoencoder framework. Specifically, the time and the space complexity are all $O(N \times d_p)$, which is competitive as compared to many neural network-based approaches. We further investigate the computational cost of our developed CRANet in Section 4.8, to show the model efficiency of CRANet as compared to state-of-the-art collaborative filtering techniques.

4 EVALUATION

We perform extensive experiments to evaluate the performance of *CRANet* on several real-world datasets, and make comparison with different types of state-of-the-art techniques. Specifically, we aim at answering the following research questions:

— **RQ1:** How does CRANet perform as compared with state-of-the-art recommendation methods on different datasets?

Table 2. Statistics of Experimented Datasets

Dataset	Density	User #	Range of User Rating #	AVG User Rating #	*s.t.d.* of User Rating #	Item #	Range of Item Rating #	AVG Item Rating #	*s.t.d.* of Item Rating #
ML-1M	0.05	6,040	20−2,314	165.60	192.73	3,706	1−3,428	269.89	383.99
ML-10M	0.01	69,878	20−7,359	143.11	216.71	10,677	1−34,864	936.60	2,487.21
Netflix	0.01	480,189	1−17,653	209.25	302.34	17,770	3−232,944	5,654.50	16,909.20
Foursquare	0.005	24,748	5−244	40.22	21.03	7,763	5−7,741	128.22	359.84
Yelp	0.007	11,894	20−1,212	58.56	50.19	8,560	10−782	81.37	52.23

— **RQ2:** How is the performance of CRANet *w.r.t* different reflection mechanism in the reflective reception network?

— **RQ3:** How does the designed reflection reception network contribute to the recommendation performance of CRANet?

— **RQ4:** How is the impact of different decay functions for the model recommendation performance?

— **RQ5:** How is *CRANet*'s forecasting performance *w.r.t* different data sparsity degrees?

— **RQ6:** How is the model performance *w.r.t* user-based and item-based collaborative filtering architecture?

— **RQ7:** How is the model efficiency of our CRANet framework in handling different scales of data?

— **RQ8:** What are the hyperparameter influences for the performance of our CRANet framework?

4.1 Experiment Settings

4.1.1 Data Description. We evaluate the proposed model on four benchmark recommendation datasets: **MovieLens-1M** (**ML-1M**), **MovieLens-10M** (**ML-10M**), Netflix prize data, and Foursquare data, corresponding to two different recommendation tasks: *rating prediction* and *item recommendation*. Table 2 summarizes the data statistics.

Online Rating Data. These three rating datasets (i.e., ML-1M,[1] ML-10M,[2] and Netflix[3]), which vary in data sparsity degrees and user/item scales, have been widely used for evaluating various recommender systems. In particular, the user-item interaction data is formatted as (user id, movie id, and rating score). Each rating score of ML-1M and Netflix data ranges from 1 to 5 stars with one star as increments, and the ML-10M data come from the five star system using half-star as the increment. The preference matrices of ML-1M, ML-10M, and Netflix prize data, is on the scale of $10^3 \times 10^3$, $10^4 \times 10^4$, and $10^5 \times 10^4$, respectively.

Foursquare Check-in Data.[4] It collects the mobility traces of users by recording their check-ins over different venues. Following the similar settings in [49], we remove users and items whose check-in interactions are less than 20.

Yelp Data.[5] This dataset has been widely used for evaluating the performance of developed recommender systems. It is collected from the Yelp platform for business venue recommendation.

---

[1]https://grouplens.org/datasets/movielens/1m/.
[2]https://grouplens.org/datasets/movielens/10m/.
[3]https://www.kaggle.com/netflix-inc/netflix-prize-data.
[4]https://sites.google.com/site/yangdingqi/home/foursquare-dataset?authuser=0.
[5]https://www.yelp.com/dataset/download.

4.1.2 Evaluation Protocols. In our experiments, we divide each experimented dataset into training and test set with the ratio of 9:1. Specifically, 5% of the training set is used as the validation set for hyperparameter tuning and the validation set determines the parameters which generate the best performance. All experiments were conducted for five runs with dataset re-division. The averaged performance is reported in the evaluation results.

To evaluate the model performance, we adopt two types of evaluation metrics for different recommendation tasks.

Rating Prediction. We employ the **Root Mean Squared Error** (**RMSE**) as the rating prediction evaluation metric which is formally defined as follows:

$$RMSE = \sqrt{\frac{\sum_{r \in \mathbb{S}} \|(1 - o_r) \circ (r - \hat{r})\|_2^2}{\sum_{r \in \mathbb{S}} \|r\|_0}}. \tag{11}$$

where r and $\hat{r}$ represent the actual and predicted rating score, respectively. The validation set is denoted as $\mathbb{S}$. During the test phase, the prediction ability of the model, instead of the reconstruction ability, is evaluated by using the vectors from the training set as inputs. The lower RMSE value indicates better performance. Through all the experiments, the standard deviation on each five RMSE scores is within 0.002 by our method which is rather small. Thus, we omit the standard deviation in the reported results.

Item Recommendation. We use two popular metrics for ranking-based performance measurement *Precision@K* and ***Normalized Discounted Cumulative Gain (NDCG@K)*** [8], to evaluate the accuracy of item, recommendations. Multiple *K* values are applied for evaluation. Note that a higher Precision and NDCG values reflect better performance.

4.1.3 Methods for Comparison. To comprehensively evaluate our CRANet, we compare it with different baselines from five research lines for comparison: (1) conventional matrix factorization recommender systems; (2) neural collaborative filtering models for recommendation; (3) learning to recommend with graph neural networks; (4) neural auto-regressive recommendation models; and (5) autoencoder collaborative filtering recommendation techniques.

Conventional Matrix Factorization Recommender Systems: We first consider several representative conventional baselines which are built based on the matrix factorization architecture.

— **PMF** [27]: It is a representative probabilistic recommendation algorithm that projects the user preference matrix into lower rank sub-matrices for users and items with the Gaussian noises.

— **SVD**++ [19]: This model combines the matrix factorization architecture with neighborhood-based methods, by integrating the latent factors and neighborhood signals into the framework.

— **BiasMF** [20]: It enhances the matrix factorization *w.r.t* the model flexibility by incorporating user and item, biases.

— **LORM** [21]: It relaxes the low-rank assumption on the interaction matrix with the local matrix factorization.

Neural Collaborative Filtering Models for Recommendation: We further compare *CRANet* with neural network-enhanced collaborative filtering methods through multilayer perceptron.

— **DMF** [48]: It is a neural matrix factorization model which learns latent vectors for both users and items, with the consideration of both explicit ratings and implicit feedback.

— **NCF** [14]: We consider three implementation versions of NCF method with different concatenation schemes: i.e., NCF-G: using the element-wise production; NCF-M: leveraging the

Multilayer perceptron for embedding projection; and NCF-N: applying the element-wise-product branch.

Neural Auto-Regressive Recommendation Models: *CRANet* is also compared against auto-regressive recommendation models with neural network architectures.

— **RBM** [30]: RBM is a representative pioneering neural network-based model which apply the classic restricted boltzmann machines to the collaborative filtering architecture.
— **NADE** [55]: It is a neural auto-regressive network for the collaborative filtering with the parameter sharing.
— **CF-UIcA** [6]: It simultaneously models users' and items' rating vectors based on the neural autoregressive framework.

Autoencoder Collaborative Filtering Architecture: One key research line of collaborative filtering models lies in the utilization of autoencoder for mapping user-item interactions into latent representations. *CRANet* also competes with the autoencoder CF paradigm.

— **ACF** [28]: The method is an autoencoder-based collaborative filtering model which regards each rating score as a unique class in the recommendation scenario.
— **AutoRec** [31]: The model learns compressed user/item representations by designing a three-layer autoencoder architecture in which rating scores treated as continuous values.
— **CDAE** [33]: It debiases the autoencoder with an adaptive loss function to learn robust user/item representations.

Recommendation with Graph Neural Networks: Another important relevant research line is to explore the user-item interactive relation structure in recommender systems. Three recent developed graph neural network models are considered in the performance comparison.

— **GC-MC** [1]: It is a graph-based autoencoder architecture that leverages the graph convolutions to jointly handle the user-item interactions and auxiliary information.
— **ST-GCN** [52]: It leverages the graph convolutional network as the aggregator in the encoder-decoder framework to learn the node embeddings for recommendation.
— **NGCF** [38]: This approach learns the vector representations of users and items, by utilizing the graph neural network to perform the propagation for collaborative signal encoding.

Imputation-based Recommender Systems: We further compare *CRANet* with the line of collaborative filtering which are on the basis of data imputation techniques:

— **S-Imp** [26]: This method imputes the missing elements with the computation of low-rank **Singular Value Decomposition (SVD)**. In S-Imp, grid value-based regularization is applied in the framework.
— **I-NMF** [35]: It is proposed to alleviate the data sparsity problem with the denoising autoencoder. The imputation-based neural matrix factorization model adopts the multiple-layer perceptron to optimize the collaborative filtering architecture.

4.1.4 Parameter Settings. We implement all models with TensorFlow and utilize Adam as the model optimizer. The dimension of the hidden layer is set to 500 which follows the settings in [31, 55]. When $\mathbf{U}$ is used to replace $\mathbf{V}^\top$ in CRANet, $\lambda_4\|\mathbf{U}\|_F^2$ is added to the loss function (6). In the hyperparameter tuning, the regularization weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are selected from {0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}, and the hyperparameter α in decay function is selected from {200, 20, 2, 0.2, 0.02}. To be consistent with the implementation of graph neural recommendation models (ST-GCN and NGCF), the depth of embedding propagation layers for those baselines is searched from the value range of {1, 2, 3, 4}.

Table 3. Performance Comparison of Rating Prediction

Category	Methods	ML-1M	ML-10M	Netflix
MF-based Model	PMF	0.883	0.791	0.902
	LLORMA	0.865	0.822	0.874
	SVD++	0.852	0.787	0.887
	BiasMF	0.845	0.803	0.844
Neural-CF	DMF	0.845	0.792	0.839
	NCF-MLP	0.866	0.799	0.825
	NCF-GMF	0.848	0.788	0.828
	NCF-NeuMF	0.840	0.780	0.818
GNN Method	GC-MC	0.832	0.777	-
	NGCF	0.830	0.778	-
	ST-GCN	0.832	0.770	-
Neural Auto-Regressive	RBM	0.854	0.823	0.845
	NADE	0.829	0.771	0.803
	CF-UIcA	0.823	0.769	**0.799**
Autoencoder CF	ACF	0.890	0.803	0.831
	AutoRec	0.831	0.782	0.823
	CDAE	0.833	0.778	0.819
Re-Imputation Method	S-Imp	0.838	0.783	0.831
	I-NMF	0.828	0.777	0.818
Our Model	CRANet	**0.822**	**0.768**	0.814

4.2 Performance Comparison (RQ1)

4.2.1 Rating Prediction Task. We first present overall evaluation results of our *CRANet* in rating prediction task with detailed discussions as follows.

Overall Performance of *CRANet*. We evaluate the performance of all compared methods and report the results in Table 3. From this table, we can observe that *CRANet* outperforms different types of baselines on various datasets in most cases, which demonstrates the effectiveness of *CRANet*. While two neural autoregressive methods (i.e., NADE and CF-UIcA) achieve slightly better performance on the Netflix dataset, we note that NADE is a more complicated model which involves more learnable parameters and is more time-consuming as compared to CRANet (the model scalability is investigated in later subsection). Due to the high computational and space cost of graph aggregation operation in GNN-based baselines (i.e., ST-GCN, GC-MC, and NGCF), it is difficult to scale them on large-scale data (i.e., Netflix), i.e., "-" indicates the out of memory issue.

Additionally, another interesting observation is that the performance gap between our CRANet and other baselines becomes larger as the the data volume decreases, i.e., the Movielens rating data contain much less ratings compared to the Netflix dataset as given in Table 3. This observation justifies that CRANet is capable of effectively capturing latent relation structures between users and items with the reflection mechanism, in order to alleviate the data scarcity issue in the recommendation scenario.

Comparison among Baselines. The relative improvement between neural models and conventional matrix factorization approaches, suggests that the neural architecture is able to capture non-linearity among user-item interactions. Among all the baselines, neural probabilistic approaches and autoencoder-based collaborative filtering outperform the conventional CF techniques as well

Table 4. Performance Comparison for Item, Recommendation in Terms of *Precision@3* and *NDCG@3*

Model	Metrics	SVD++	BiasMF	DMF	NCF	GC-MC	NGCF	ST-GCN	NADE	UIcA	ACF	AutoRec	CDAE	S-Imp	I-NMF	*CRANet*
ML-1M	Prec@3	6.57%	6.58%	7.39%	8.62%	7.01%	9.18%	7.64%	13.81%	14.96%	7.92%	8.21%	8.65%	6.89%	8.44%	**15.71%**
	NDCG@3	2.49%	2.49%	3.05%	3.51%	2.91%	3.84%	3.27%	5.51%	5.96%	3.01%	3.38%	3.51%	2.74%	3.44%	**6.26%**
Fsq	Prec@3	4.32%	4.57%	4.40%	4.63%	4.30%	4.98%	4.64%	4.93%	4.88%	4.02%	4.68%	4.81%	4.66%	4.75%	**7.27%**
	NDCG@3	3.82%	4.03%	3.85%	4.08%	3.76%	4.60%	4.19%	4.44%	4.11%	3.26%	4.05%	4.15%	4.10%	4.10%	**6.91%**
Yelp	Prec@3	2.23%	2.42%	2.49%	2.50%	2.58%	2.73%	2.58%	2.94%	3.01%	2.72%	2.83%	2.79%	2.59%	2.93%	**3.21%**
	NDCG@3	1.50%	1.58%	1.73%	1.84%	1.77%	1.83%	1.80%	2.01%	2.03%	1.81%	2.01%	1.97%	1.82%	2.04%	**2.21%**
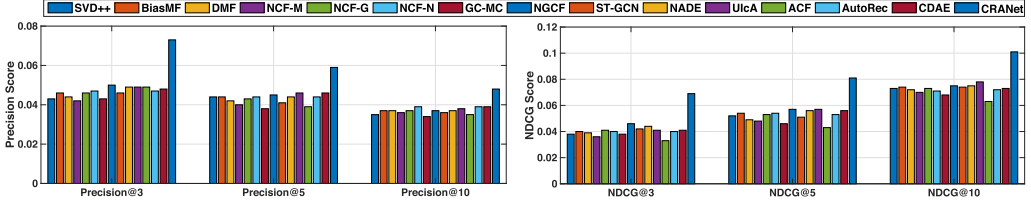
Fig. 2. Performance validation for location prediction in terms of *Precision@K* and *NDCG@K*.

as the neural CF models with the integration of MLP, suggesting that autoencoder and neural probabilistic models are more beneficial for modeling non-linear user-item interactions as compared to the simple MLP module. In the occasional cases that CRANet misses the best performance, it still generates very competitive results on the Netflix dataset when compared with two NADE-based models. However, we note that NADE-based methods are more complicated models which involve more learnable parameters and require larger computational cost during the training phase.

4.2.2 Item Recommendation Task. We further perform experiments to evaluate all methods on three datasets (i.e., ML-1M, Foursquare, and Yelp) for the item, recommendation task. As given in Table 4, we can observe that we can observe that our proposed method *CRANet* achieves the best performance as compared to various baselines, which demonstrates the effectiveness of our reflection-augmented autoencoder network in capturing the implicit users' preference from their unobserved interactions. From the presented overall performance comparison for item recommendation, we can observe that our CRANet achieves relatively 8.87%, 9.95% and 8.33% improvement over three best performed baselines CF-UIcA, NADE, I-NMF, in terms of NDCG@3 on Yelp dataset. This observation validates the superiority of our method over various state-of-the-art recommendation techniques. Additionally, Figure 2 shows the ranking performance of top-*K* recommended locations on Foursquare dataset where $K \in \{3, 5, 10\}$. We can see that *CRANet* achieves the best performance in both metrics Precision and NDCG in all cases, which points to the positive effect of modeling implicit users' preference over their non-interacted locations. Furthermore, while ST-GCN and NGCF utilize the high-order neighbor information to guide the representation learning of user-location interactions, they simply treat the unobserved interactions with zero imputation, which may lead to suboptimal performance. This further verifies the importance of capturing cross-modal collaborative signals in the embedding function of our reflective collaborative filtering framework.

4.3 Impact of Reflection Mechanism Configuration in CRANet (RQ2)

To study the impact of different reflective reception network design in *CRANet*, we perform experiments with different reflection mechanism configurations (an independent transformation operation $\mathbf{U}$, transpose-based reflection gate $\mathbf{V}^{\top}$, and regularization-based reflection gate $\mathbf{T}$ with $\mathcal{L}_{\text{reg}}$). We also evaluate the performance by incorporating $\mathbf{T}$ but without the regularization term defined in Equation (5). For fair comparison, when perform experiments on the same dataset, all models
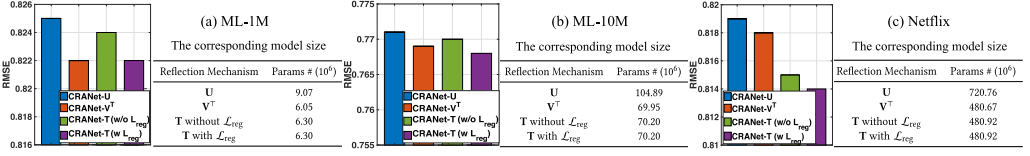
Fig. 3. Recommendation performance of *CRANet* with different reflection mechanism configurations.

Table 5. Effect of the Reflection Reception

Model	ML-1M	ML-10M	Netflix
CRANet-R	0.831	0.782	0.823
CRANet-N	0.838	0.779	0.821
CRANet	**0.822**	**0.768**	**0.814**

share the same hyperparameters λ and α. We also apply the regularization $\|\mathbf{U}\|_{\mathrm{F}}^2$ in the independent transformation.

Figure 3 shows the performance and corresponding model sizes. We observe that our regularization-based reflection mechanism $\mathbf{T}$ with $\mathcal{L}_{\mathrm{reg}}$ achieves the best performance when competing with other alternatives. In addition, the performance gap between $\mathbf{T}$ with $\mathcal{L}_{\mathrm{reg}}$ and $\mathbf{V}^\top$ increases, as the average item, density increases (see Table 2), which suggests that the transpose-based reflection gate $\mathbf{V}^\top$ might over-regularize the model on large datasets and the incorporation of a lightweight regularization operation ($\mathbf{T}$ with $\mathcal{L}_{\mathrm{reg}}$) could alleviate this overfitting issue. Finally, it is worth mentioning that both $\mathbf{V}^\top$ and $\mathbf{T}$ have very low memory cost, given $d_p << N$. Therefore, our *CRANet* can be easily scaled to the largest Netflix dataset with efficient model training, while the independent transformation $\mathbf{U}$ has high memory consumption.

4.4 Model Ablation Study (RQ3)

In this subsection, we perform model ablation study to investigate the effectiveness of our designed reflective reception network from the following two perspectives:

— *CRANet*-R: We remove the reflective reception network from the *CRANet* and directly utilize a three-layer autoencoder network to learn latent representations of user-item interaction $\mathbf{R}$.

— *CRANet*-N: We replace the reflective reception network with a neighborhood based data imputation approach based on a similarity estimation scheme across users and items [5].

The experimental results are reported in Table 5. With the help of our reflective reception network, our *CRANet* achieves the best performance as compared to *CRANet*-R and *CRANet*-N. The significant performance improvement suggests the effectiveness of our reflective reception module in capturing users' dynamic preferences over their non-interacted items, with an inner neural network structure to avoid the misleading of the user-item relation learning.

4.5 Effect of Decay Function (RQ4)

We perform experiments to investigate the impact of decay function $\phi(\mathbf{R}_j)$ selection in the proposed reflective reception network of CRANet. In particular, recall four choices of the numerator in decay function mentioned in Section 3.2. Different choices result in the following different decay
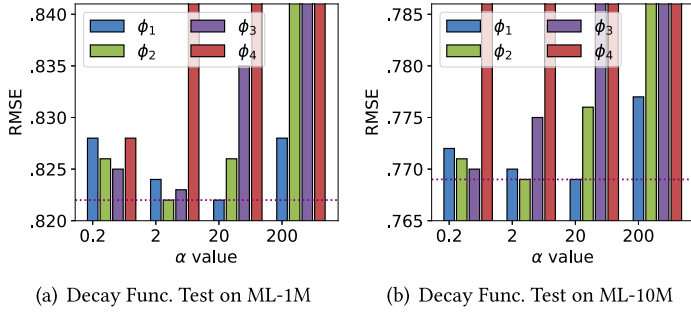
(a) Decay Func. Test on ML-1M (b) Decay Func. Test on ML-10M

Fig. 4. Model performance v.s. decay function settings on Movielens-1M and Movielens-10M Data.

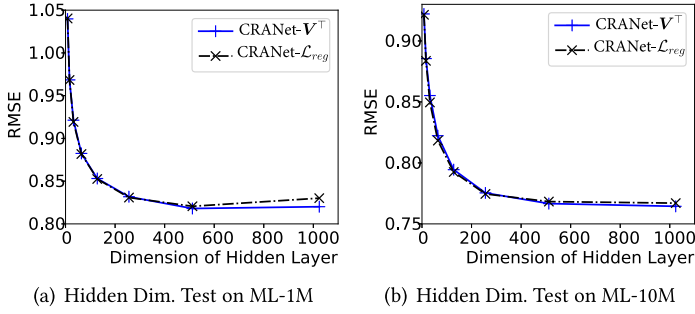(a) Hidden Dim. Test on ML-1M (b) Hidden Dim. Test on ML-10M

Fig. 5. Model Performance v.s. hidden layer dimensionality on Movielens-1M and Movielens-10M Data.

Table 6. Best Performance of *CRANet* for Different Decay Functions with the Corresponding α

Dataset	ϕ_1		ϕ_2		ϕ_3		ϕ_4	
	α	RMSE	α	RMSE	α	RMSE	α	RMSE
ML-1M	20	**0.822**	2	**0.822**	2	0.823	0.02	0.827
ML-10M	20	**0.768**	2	0.769	0.2	0.770	0.02	0.773

function candidates in formal presentations:

$$\phi_1(\mathbf{R}_j) = \frac{\alpha}{\|\mathbf{R}_j\|_0} \quad \phi_2(\mathbf{R}_j) = \frac{\alpha \log\left(\|\mathbf{R}_j\|_0 + 1\right)}{\|\mathbf{R}_j\|_0},$$

$$\phi_3(\mathbf{R}_j) = \frac{\alpha\sqrt{\|\mathbf{R}_j\|_0}}{\|\mathbf{R}_j\|_0} \quad \phi_4(\mathbf{R}_j) = \frac{\alpha\|\mathbf{R}_j\|_0}{\|\mathbf{R}_j\|_0} = \alpha.$$

In the experiments, the tied weight-based reflection mechanism is employed. Figure 4(a) and (b) shows the recommendation results with respect to the parameter α, and Table 6 summarizes the best performance of the four candidates with the corresponding α value. For all candidates except that using ϕ_4, a proper α (i.e., $\alpha \in \{2, 20\}$) is preferred to balance the original and the reflected representations, while ϕ_4 is likely to lead to the failure on larger α and cannot be adaptive to input $\mathbf{R}_j$. Such results indeed verify the effectiveness of our adaptive decay function in *CRANet*. The models using ϕ_1 and ϕ_2 are the best ones and they show very close prediction performance.
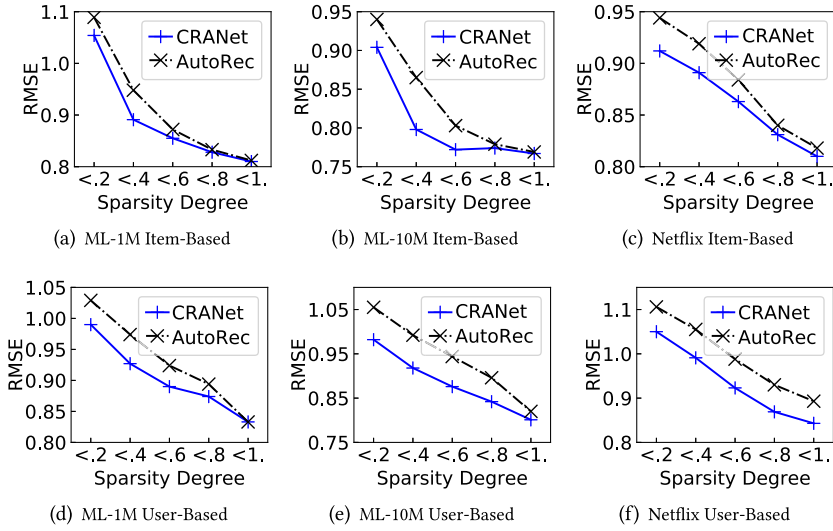
Fig. 6. Performance evaluation with varying sparsity degree. The *x*-axis denotes the interval of sparsity degree on the sub-dataset, e.g., the left-most two points are the test results on the sparsest 20% input data. The *y*-axis denotes the average RMSE scores.

Table 7. Performance of CRANet *w.r.t* Different Reflection Mechanisms

Item/User	Reflection Mechanism	ML-1M	ML-10M	Netflix
User-based	$\mathbf{V}^{\top}$	0.858	0.813	0.849
	$\mathbf{T}$ with $\mathcal{L}_{\text{reg}}$	0.860	0.831	0.868
Item-based	$\mathbf{V}^{\top}$	**0.822**	0.769	0.818
	$\mathbf{T}$ with $\mathcal{L}_{\text{reg}}$	**0.822**	**0.768**	**0.814**

4.6 Performance *v.s.* Data Sparsity (RQ5)

We perform experiments by varying sparsity degrees of input user-item interaction data, to investigate the capability of *CRANet* in dealing with sparse inputs. In specific, each dataset containing items' rating vectors is divided into five subsets equally, with the first sub-dataset containing the sparsest 20% input vectors and their corresponding test vectors, the second one containing the second sparsest 20% vectors, and so on. The model training and test process are separately performed for each dataset. The experimental results of the proposed *CRANet* and AutoRec are shown in Figure 6. We could notice that *CRANet* consistently outperforms AutoRec in most evaluation cases, which further demonstrates that the reflection mechanism is very helpful in modeling unobserved user-item relations from highly sparse user-item interaction data.

4.7 Item-Based *v.s.* User-Based *CRANet* Performance Investigation (RQ6)

To explore the impact of item-based and user-based recommendation framework, it is interesting to see how our *CRANet* performs when generating the user-item interaction matrix **R** from either user or item, side. Table 7 lists the evaluation results of *CRANet* with the designed different reflection mechanism (i.e., transpose-based reflection gate–$\mathbf{V}^{\top}$ and implicit reflection gate–$\mathbf{T}$ with $\mathcal{L}_{\text{reg}}$). We can observe that the item-based *CRANet* framework consistently outperforms the user-based *CRANet* framework with different network configurations, which is consistent with

Table 8. Model Scalability Study with Computational Cost
(Measured by Running Time Seconds) and the Corresponding Model Size
(# of Trainable Parameters in Millions)

Dataset	ML-1 M		ML-10 M		Foursquare	
Method	Time	Params #	Time	Params #	Time	Params #
BiasMF	1.80 s	4.89 M	19.03 s	40.44 M	20.75 s	16.29 M
DMF	5.18 s	4.87 M	57.24 s	40.28 M	49.63 s	16.26 M
NCF	1.96 s	5.37 M	27.29 s	40.78 M	21.43 s	16.76 M
ACF	1.53 s	30.23 M	27.45 s	699.48	25.68 s	27.19 M
AutoRec	1.00 s	6.05 M	19.15 s	69.95 M	18.94 s	7.77 M
CDAE	1.41 s	6.05 M	19.54 s	69.95 M	19.08 s	7.77 M
NADE	3.61 s	30.23 M	39.61 s	10.83 M	25.14 s	27.19 M
CF-UIcA	2.37 s	48.78 M	29.05 s	806.36 M	26.94 s	113.85 M
NGCF	4.52 s	24.87 M	41.90 s	403.28 M	44.08 s	81.78 M
ST-GCN	3.97 s	24.62 M	32.40 s	403.03 M	37.96 s	81.53 M
CRANet	1.09 s	6.30 M	18.54 s	70.72 M	19.01 s	8.02 M

the observations of state-of-the-art autoencoder-based recommendation techniques [28, 31, 33]. The performance gap may result from the lower density degrees of user's interaction vectors as compared to that of items, which leads to the overfitting.

4.8 Scalability Study of CRANet (RQ7)

In addition to evaluating the model effectiveness, we also investigate the efficiency of *CRANet* and state-of-the-art baselines as reported in Table 8 (experiments are performed on one Titan V card). We can observe that the computational cost (i.e., measured by running time and the corresponding model size) of *CRANet* is much less than that of NADE, which demonstrates the efficiency of *CRANet*, i.e., our reflective reception network with implicit reflection mechanism is a scalable learning module in terms of the space and computational cost.

4.9 Hyperparameter Study of *CRANet* (RQ8)

We also explore the influence of hidden state dimensionality for the recommendation performance. During the parameter sensitivity study, we vary the target hyperparameter within a specific value ranges and keep other hyperparameters as default settings. The results of the item-based *CRANet* framework with both the tied-weight and the implicit reflection schemes are shown in Figure 5(a) and (b). The experimental results of two versions are quite similar. Furthermore, with the increasing of the dimension of hidden layer, the prediction error first sharply decreases, then varies little and finally increase a bit due to overfitting. According to the experimental results, a hidden layer containing around 500 units causes no significant over-fitting or under-fitting. Models in other experiments use 500 as the dimensionality of hidden layers.

4.10 Case Study

We visualize the latent representations learned by *CRANet* to investigate the interpretation ability of our reflective reception network. We show the representations of the corresponding hidden layer from several typical data instances and make comparison with the best performed autoencoder-based model variant *CRANet*-R as shown in Figure 7. For data instances with few observations like Figure 7(a) and (b), *CRANet*-R generally gives representations which contain a lot of neutral values (close to 0.5, colored in green), due to the trade-off between the sparse and dense vectors.
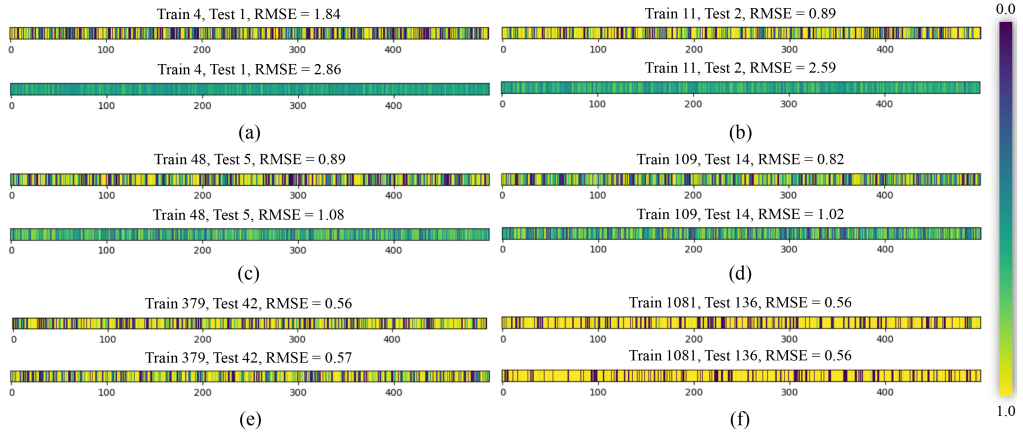
Fig. 7. Typical samples of activated latent representations given by CRANet on ML-1M dataset. The samples are visualized by heat maps. The upper and bottom heatmaps in (a)–(f) show the representations for one data instance given by CRANet and AutoRec, respectively, and each strip contains 500 elements whose values are represented by colors. The relationship between color and value is shown by the color-bar in the legend: the brighter the color is, the larger the value will be. The numbers of observed ratings in training vectors, the numbers of corresponding tested ratings, and the prediction errors are also shown.

Such representations can hardly show good interpretation ability in encoding interactive relations between users and items.

In contrast, owing to the reflected representations normalized by data sparsity in *CRANet*, we give such data instances almost full of activated (bright yellow) and suppressed (dark purple) elements, which is more distinguishable. Such higher discrimination brings the better performance of *CRANet* in dealing with highly sparse vectors. We also found that both methods sometimes assign non-distinguishable representations to those with fairly enough observations (e.g., Figure 7(c) and (d)). This may be caused by the inherent characteristics of the data instances. In comparison, *CRANet* can still give much more distinguishable representations, with more neutral values than those in Figure 7(a) and (b).

5 CONCLUSION

This article proposes a new reflective collaborative filtering framework (namely CRANet) to model implicit multi-modal user-item relations for better recommendation. The CRANet model integrates the designed reflection mechanism with a cross-modal autoencoder network for latent representation mapping. The newly developed reflection mechanism endows the collaborative filtering paradigm with the ability of encoding more informative user-item relations in correcting variable sparsity bias. Our proposed frameworks achieve significantly better performances as compared to state-of-the-art techniques on both rating prediction and item, recommendation applications across multiple datasets. Since CRANet is a generic reflective collaborative filtering, our future work plans to extend CRANet with other recommendation architecture, such as factorization machines.

REFERENCES

[1] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph convolutional matrix completion. arXiv:1706.02263. Retrieved from https://arxiv.org/abs/1706.02263.

[2]   Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[3]   Wanyu Chen, Fei Cai, Honghui Chen, and Maarten De Rijke. 2019. Joint neural collaborative filtering for recommender systems. *Transactions on Information Systems* 37, 4 (2019), 1–30.

[4]   Yifan Chen, Yang Wang, Xiang Zhao, Jie Zou, and Maarten De Rijke. 2020. Block-aware item similarity models for Top-N recommendation. *Transactions on Information Systems* 38, 4 (2020), 1–26.

[5]   Christian Desrosiers and George Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Proceedings of the Recommender System Handbook*. 107–144.

[6]   Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. 2018. Collaborative filtering with user-item co-autoregressive models. In *International Conferences on Artificial Intelligence*.

[7]   Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[8]   Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep adversarial social recommendation. In *International Conference on Recommender Systems*. ACM.

[9]   Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. ACM, 417–426.

[10]   Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, and Gerard de Melo. 2020. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[11]   Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong 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*. ACM.

[12]   Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2227–2233.

[13]   Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural attentive item similarity model for recommendation. *Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[14]   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*. ACM, 173–182.

[15]   Chao Huang, Xian Wu, Xuchao Zhang, Chuxu Zhang, Jiashu Zhao, Dawei Yin, and Nitesh V. Chawla. 2019. Online purchase prediction via multi-scale modeling of behavior dynamics. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2613–2622.

[16]   Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *International Conferences on Artificial Intelligence*.

[17]   Wenhao Jiang , Hongchang Gao , Wei Lu, Wei Liu, Fu-Lai Chung, and Heng Huang. 2018. Stacked robust adaptively regularized auto-regressions for domain adaptation. *Transactions on Knowledge and Data Engineering* 31, 3 (2018), 561–574.

[18]   Hanna Kamyshanska et al. 2013. On autoencoder scoring. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*. 720–728.

[19]   Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 426–434.

[20]   Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[21]   Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local low-rank matrix approximation. In *Proceedings of the 30th International Conference on Machine Learning*. 82–90.

[22]   Youngnam Lee, Sang-Wook Kim, Sunju Park, and Xing Xie. 2018. How to impute missing ratings? Claims, solution, and its application to collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 783–792.

[23]   Defu Lian, Xing Xie, Enhong Chen, and Hui Xiong. 2020. Product quantized collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*.

[24]   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*. ACM, 689–698.

[25]   Wei Luo, Jun Li, Jian Yang, Wei Xu, and Jian Zhang. 2017. Convolutional sparse autoencoders for image classification. *Transactions on Neural Networks and Learning Systems* 29, 7 (2017), 3289–3294.

[26] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research* 11 (2010), 2287–2322.

[27] Andriy Mnih and Ruslan Salakhutdinov. 2008. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*. 1257–1264.

[28] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong. 2014. Autoencoder-based collaborative filtering. In *Proceedings of the International Conference on Neural Information Processing*. 284–291.

[29] Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, and Yong Yu. 2019. Deep landscape forecasting for real-time bidding advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 363–372.

[30] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.

[31] 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*. ACM, 111–112.

[32] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 555–563.

[33] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 11–16.

[34] Florian Strub and Jeremie Mary. 2015. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *Proceedings of the Conference on Neural Information Processing Systems Workshop*.

[35] Xinyue Wan, Bofeng Zhang, Guobing Zou, and Furong Chang. 2019. Sparse data recommendation by fusing continuous imputation denoising autoencoder and neural matrix factorization. *Applied Sciences* 9, 1 (2019), 54.

[36] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational stacked denoising autoencoder for tag recommendation. In *Proceedings of the International Conferences on Artificial Intelligence*. 3052–3058.

[37] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 968–977.

[38] 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*. (2019).

[39] Xin Wang, Wenwu Zhu, and Chenghao Liu. 2019. Social recommendation with optimal limited attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & and Data Mining*. ACM, 1518–1527.

[40] Ga Wu, Mohamed Reda Bouadjenek, and Scott Sanner. 2019. One-Class collaborative filtering with the queryable variational autoencoder. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 921–924.

[41] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 346–353.

[42] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. ACM, 153–162.

[43] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2397–2406.

[44] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the International Conference on Artificial Intelligence*. 4486–4493.

[45] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[46] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[47] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep item-based collaborative filtering for top-n recommendation. *Transactions on Information Systems* 37, 3 (2019), 1–25.

[48] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, et al. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3203–3209.

[49] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In *Proceedings of the World Wide Web Conference*. ACM, 2147–2157.

[50] Joonyoung Yi, Juhyuk Lee, Kwang Joon Kim, Sung Ju Hwang, and Eunho Yang. 2020. Why not to use zero imputation? correcting sparsity bias in training neural networks. In *Proceedings of the 8th International Conference on Learning Representations*.

[51] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. 2016. Discrete collaborative filtering. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 325–334.

[52] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

[53] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders. In *IProceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[54] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-Domain recommendation via preference propagation GraphNet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2165–2168.

[55] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. 764–773.