A Deep Learning Model for Cross-Domain Serendipity Recommendations

ZHE FU, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, United States
XI NIU, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, United States
XIANGCHENG WU, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, United States
RUHANI RAHMAN, Computer Science, The University of North Carolina at Charlotte, Charlotte, United States

Serendipity means unexpected discoveries that are valuable, with positive outcomes ranging from personal benefits to scientific breakthroughs. This study proposes a cross-domain recommendation model, called *SerenCDR*, to model serendipity. *SerenCDR* leverages the knowledge beyond one domain as well as mitigates the inherent data sparsity problem in serendipity recommendations. The novelty of *SerenCDR* lies in the fact that it is the first deep learning-based cross-domain model for a serendipity task. More importantly, it does not rely on any overlapping users or overlapping items across different domains, which especially fits for the task of recommending serendipity, because serendipity in a single domain tends to be sparse; finding overlapping users or overlapping items in other domains are nearly impossible. To train and test *SerenCDR*, we have collected a two-domain ground truth dataset on serendipity, called *SerenCDRLens*. In addition, since we found that serendipity is sparse in *SerenCDRLens*, we designed an auxiliary loss function to supplement the main loss function to enhance serendipity learning. Through a series of experiments, we have harvested positive performance in recommending serendipity, empowering users with increased chances of bumping into unexpected but valuable discoveries.

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

Additional Key Words and Phrases: Cross-domain Recommendations, Serendipity, Deep Learning

1 Introduction

In the recent decade, the notion of serendipity has been advocated by many recommender researchers. Current deep learning-based recommendation models, like many machine learning models, tend to overly focus on recommendation accuracy [2, 9, 10, 46]. However, many people have expressed their hope that recommender systems could play a role in facilitating incidental exposure to serendipitous information, whereby individuals “stumble upon” unexpected but valuable items that they did not actively seek. As early as 1997, Gup [15] expressed his concern about the “end of serendipity" in the digital world, recalling with fondness his childhood experiences

Authors' Contact Information: Zhe Fu, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, North Carolina, United States; e-mail: zfu2@uncc.edu; Xi Niu, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, North Carolina, United States; e-mail: xniu2@uncc.edu; Xiangcheng Wu, Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, North Carolina, United States; e-mail: xwu20@uncc.edu; Ruhani Rahman, Computer Science, The University of North Carolina at Charlotte, Charlotte, North Carolina, United States; e-mail: rrahman3@uncc.edu.

ACM 2770-6699/2024/8-ART
https://doi.org/10.1145/3690654

coming across interesting tidbits of information while flipping encyclopedia pages. In his article, Gup stated that the "vastly more efficient" pursuit of information supported by computers would rob us of the "random epiphanies" and ’accidental discoveries’ that are limited in an information environment tailored to our needs and where "nothing will come unless summoned." Gup’s concerns are echoed by others in more recent studies (e.g., [39, 51, 54]). The sense that the online environment is increasingly determined promotes a widespread feeling that serendipity is threatened.

The word serendipity was created in 1754 to describe unexpected but valuable discoveries [49]. In the early 2000s, serendipity was first introduced to the context of recommender systems [21] in order to improve users’ engagement and satisfaction. Although currently there is no consensus on the definition of serendipity in the context of recommender systems, most researchers interpret and operationalize this notion with two facets: unexpectedness and relevance (e.g., [22, 31, 42, 43, 45, 61]). In this study, these two facets were used to guide the collection of the serendipity ground truth data *SerenCDRLens* as well as to guide the design of the auxiliary loss function.

Modeling the serendipity relationship is difficult due to the elusive nature of serendipity. The element of unexpectedness in serendipity means surprise and accident [44, 47], which are susceptible to modeling and prediction. In addition, collecting ground truth data is difficult due to the sparse nature of serendipity. Serendipity does not occur frequently in a natural environment. Serendipity by nature will have a data sparsity problem in the collected ground truth dataset. To address this challenge, we leveraged extra data from other domains as supplements to enhance the serendipity “signals”. In addition, multi-domains nourish richer associations, increasing the potential for the happening of serendipity.

Compared to the existing cross-domain deep learning recommendation models, our model does not rely on overlapping users or overlapping items as the “bridge” across different domains. Instead, we proposed an approach to extracting the “essence” of users and items in terms of their serendipity relationship. The extracted “essence” can be shared in different domains without the same users or the same items in these domains. This is especially important for serendipity recommenders, because serendipity in a single domain has a data sparsity problem; finding overlapping users or overlapping items in other domains are nearly impossible. In order to train and test *SerenCDR*, we collected a two-domain (books and movies) serendipity ground truth dataset called *SerenCDRLens* using the *URCW (User Reviews plus Crowd Wisdom)* approach in [12], a scalable ground truth collection approach by using both existing user generated reviews and a crowd sourcing method. In addition, in order to "strengthen" the sparse serendipity in each domain in *SerenCDRLens*, we designed an auxiliary loss function, leveraging the two facets of serendipity, to supplement the regular main loss function, Specifically, the auxiliary function pulls a user away from the expected items, but not too far away from relevant ones. The ground truth for both the expected items and the relevant items can be calculated or observed from the data itself, rather than relying on a human labeling process. The experiment results demonstrated better performance of *SerenCDR* on identifying serendipitous items than the state-of-the-art baseline models.

The main contributions of this paper are summarized as three-fold:

- The *SerenCDR* model: The first cross-domain deep learning model for serendipity recommendations, without relying on any overlapping users or items across domains.
- The auxiliary loss function: A supplement to the main loss function in order to strengthen the serendipity learning in each domain.
- The *SerenCDRLens* dataset: The two-domain serendipity ground truth data. The naming follows the naming convention of the well-known dataset for recommendation models, *MovieLens* [17].

2 Related Work

This project draws on research into serendipity, deep learning recommendation models for serendipity, and cross-domain recommendations. We will review the related work in these areas in the following sections.

2.1 The Concept of Serendipity and Its Distinction with Diversity and Novelty

First coined by Harold Walpole in 1754 [40], the word "serendipity" is used to describe the process of making discoveries by accident, but it received little attention until the mid-1900s when it was used as a descriptor of accidental or unplanned discovery in the scientific context [40].

Serendipity, diversity, and novelty are all "beyond-accuracy" goals proposed for recommender systems in recent years [1, 8, 11, 23]. It is worth distinguishing between them since there is significant overlap and potential for confusion. Diversity has been studied in the IR field since 1998 when Carbonell and Goldstein investigated the relationship between diversity and retrieval accuracy [4]. In the past decade, there is a growing consensus that user satisfaction and engagement have been improved with such diversification, even at the cost of some retrieval accuracy [52, 55]. Sometimes, the concept of diversity is also introduced to mitigate the frequency bias in observed user behavior and recommendations [35]. We believe diversity increases the chance of serendipity, but not every diversified result is serendipitous because they are not necessarily unexpected. As to novelty, it means how new, different, or unknown an item is to a user [14, 21], not necessarily unexpected as well. Therefore not all novel items are serendipitous. In contrast, serendipity means how unexpected but relevant an item is. Therefore, the element of unexpectedness is the main distinction between serendipity, diversity, and novelty.

2.2 Deep Learning Recommendation Models for Serendipity

Recently deep learning has been changing recommender systems research dramatically and bringing more opportunities to improve the recommendation accuracy (or relevance in this context). Since 2018, a few information retrieval (IR) researchers have attempted to build deep learning models for serendipity recommendations. We believe Pandey et al. [48] is the first effort to build a deep learning model to predict serendipity. The model, called SerRec, used a pre-training and fine-tuning mechanism to firstly train a deep neural network for relevance scores using a large *MovieLens* dataset and then fine-tune the model for serendipity scores using the smaller dataset *Serendipity 2018* [25]. Their pre-training and fine-tuning approach mitigated the issue of a small serendipity dataset and achieved reasonable NDCG (**N**ormalized **D**iscounted **C**umulative **G**ain) scores in predicting serendipity. However, the dataset *Serendipity 2018* [25] was collected using a small-scale survey with 481 participants. In addition, the controlled environment that relied on participants' instant recall is not ideal for collecting serendipity experiences. A period of "incubation" is sometimes necessary before serendipity is recognized [38].

With the lack of large-scale ground truth data for serendipity, other researchers are working to define serendipity in ways to leverage various existing relevance-oriented datasets. For example, Li et al. [33] defined serendipity as content difference and genre accuracy for a movie recommender, and then developed an algorithm called HAES (**H**ybrid **A**pproach for movie recommendations with **E**lastic **S**erendipity), to achieve the two aspects. In their follow-up study, Li et al. [34] adjusted the definition of serendipity as an item with a direction pointing from the short-term demand to the long-term preference as well as a suitable distance to the short-term demand. They developed a deep learning algorithm called DESR (**D**irectional and **E**xplainable **S**erendipity **R**ecommendation), to achieve the computational definition. Also, Xu et al. [58] defined serendipity as high satisfaction and low initial interest, and achieved both by using a neural network, called NSR (**N**eural **S**erendipity **R**ecommendation). Li et al. [31] presented a novel PURS (**P**ersonalized **U**nexpected **R**ecommender **S**ystem) model, the first deep learning model to incorporate the notion of unexpectedness into the recommendations. They defined the item's unexpectedness level as the item's average distance to each cluster center of a user's interests. The study pre-calculated the level of unexpectedness without putting the unexpectedness module into the model training process. In the most recent study, Zhang et al. [61] "engineered" the ground truth data on serendipity by defining a way to calculate the level of serendipity. The core part is how to calculate an item's level of unexpectedness. They calculated it as the sum of item difference and the category difference. With the engineered ground truth data available, they then trained and tested a model called SNPR (**S**erendipity-oriented **N**ext **P**OI **R**ecommendation model), which is based on the state-of-the-art Transformers [56].

In summary, these deep learning efforts mentioned above collectively demonstrate the effectiveness of deep learning in representing users' preferences. However, those serendipity definitions, model designs, and those self-defined evaluation metrics were designed to leverage existing data and avoid collecting the direct ground truth on serendipity, making both the models and the results not comparable or generalizable.

2.3 Cross-Domain Recommendations

Cross-domain recommender systems enhance recommendations in a target domain using the knowledge learned from one or multiple source domains. In literature, there are distinct definitions of the domain. The study [27] defined three types of domains: a system, a time period, and a type of data. The study [3] defined four types of domains: an item attribute, an item type, an item, and a system. We believe there is some confusion in this study [3] in the distinction between an item type and an item. We grouped both as an item category. We found the item category (such as movies, books, toys, electronics, etc) as the domain is widely used by other researchers (e.g., [26, 30, 59, 62]). Therefore, in this paper, we define a domain as an item category.

According to Cremonesi et al. [7], cross-domain recommendation research can be formulated as three different types of tasks: 1) linked-domain recommendations: items of the target domain are recommended to target users based on knowledge learned from the source domain; 2) cross-domain recommendations: items in the source domain are recommended to users of the target domain or vice versa; and 3) multi-domain recommendations: items of both domains are recommended to users of both or one domain. In this paper, technically, *SerenCDR* is able to accomplish all three types of tasks. Practically, our collected ground truth dataset *SerenCDRLens* does not have overlapping users or items in different domains. It is not possible to evaluate *SerenCDR*'s performance on the second type (cross-domain) and the third type (multi-domain) recommendation tasks. Therefore, we will focus on the first type (linked-domain) task in this paper.

Cremonesi et al. [7] also identified four user-item overlap scenarios in cross-domain recommendations as shown in Figure 1: 1) both users and items have some overlaps in the two domains (U-I); 2) only users have overlaps but not items (U-NI); 3) only items have overlaps but not users (NU-I); and 4) neither users nor items have overlaps (NU-NI). Most studies on cross-domain recommendations focus on Scenario 2 and 3, for example, the studies of [5, 16, 32] for the U-NI scenarios, the research of [13, 63] for the NU-I scenarios. However, very few studies have worked on the NU-NI scenario due to the challenge of no overlapping information across the domains. In the following paragraph, we will review the few state-of-the-art cross-domain studies for the NU-NI scenarios.

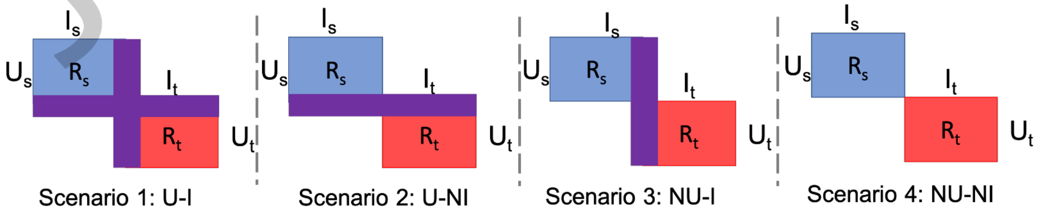


Fig. 1. User-item overlap scenarios

The most classical NU-NI study is the Codebook Transfer (CBT) [28]. It learned the "codebook" (i.e., cluster-level preference pattern) from a source domain. Then, the "codebook" was transferred to the target domain to learn the membership of its users and items to the corresponding clusters. Later on, there are some extensions of CBT, such as RMGM (**R**ating **M**atrix **G**enerative **M**odel, Li et al., [29]), TALMUD (**Tr**A**n**sfer **L**earning for **MU**ltiple **D**omains, Moreno et al. [41]), LKT-FM (**L**ow-rank **K**nowledge **T**ransfer via **F**actorization **M**achines, Zang & Hu [60]), MINDTL (**M**ultiple Rating Pattern **T**ransfer **L**earning, He et al. [18]), ACTL (**A**daptive **C**odebook **T**ransfer **L**earning, He et al. [19]), and MMMF (**M**aximum **M**argin **M**atrix **F**actorization, Veeramachaneni et al. [57]). All the methods mentioned above learned the shared cluster-level preference pattern from source domains without relying on overlapping users or items. However, these approaches transferred the shared pattern without considering any domain alignment.

Some other studies attempted to incorporate a domain factor to align the source domain(s) and the target domain before transferring any knowledge. For example, CIT (**C**onsistent **I**nformation **T**ransfer, Zhang et al. [62]) defined how two rating matrices in two domains were consistently tri-factorized and how the consistent knowledge could be extracted. MHF (**M**ixed **H**eterogeneous **F**actorization, Yu et al. [59]) introduced domain-shareable and domain-specific factors in a matrix factorization method. In addition, they defined a domain weighting coefficient to quantify a source domain's consistency with the target domain, and then used the coefficient as the source domain's importance weight in the model objective function.

Those methods demonstrate the feasibility of cross-domain recommendations without having any overlapping users or items in different domains. They have shown the existence of the "essence" knowledge for domains even if the users and items are completely different. However, all of those studies are for the recommendation accuracy task, and all are based on the classical matrix factorization (MF) techniques. With the rapid advances in deep learning techniques, we would like to propose a first deep learning approach for cross-domain recommendations under the NU-NI scenario. And importantly, our prediction task is beyond recommendation relevance and is a serendipity-oriented task.

3 The Proposed *SerenCDR* Model

We propose the *SerenCDR* model for two domains: one is the source domain and the other is the target domain. The model can be extended to multi-domain recommendations where the first $(n-1)$ domains are the source domains and the last n^{th} domain is the target domain. The mathematical notations used in *SerenCDR* are listed in Table 1. Details of *SerenCDR* will be introduced in the following subsections.

3.1 Problem Formulation

Serendipity recommendations, like any other information retrieval problems, are essentially a matching problem. Given a user u and an item i, the degree of matching is typically measured as a matching score produced by a matching function based on the representations of the user u and the item i:

$$match(u, i) = F(\mathbf{u}, \mathbf{i}), \tag{1}$$

where $\mathbf{u}$ and $\mathbf{i}$ are the representations of u and i respectively. F is the matching function based on the interactions between the two representations. The k items with the highest matching scores will make the final recommendation list. In cross-domain serendipity recommendations, there are typically two domains: D_s (the source domain) and D_t (the target domain). In a more practical setting for a serendipity recommendation problem, the two domains do not have any overlapping users or items. Our goal is to recommend for each user in the target domain the most likely serendipitous items in the same target domain with the help of the knowledge from the source domain. Computationally, our goal for the cross-domain serendipity recommendations can be written as learning the matching function in the target domain with the help of the learning process of the matching function in the source domain, equivalent to learning two matching functions simultaneously:

Table 1. Mathematical notations

Symbol	Description
D_t	the target domain
D_s	the source domain
e_{ut}, e_{us}	a user embedding for the target/source domain
e_{it}, e_{is}	an item embedding for the target/source domain
$\mathbf{U_t}$, $\mathbf{U_s}$	shareable user knowledge for the target/source domain
$\mathbf{I_t}$, $\mathbf{I_s}$	shareable item knowledge for the target/source domain
$\mathbf{T^U}$, $\mathbf{T^I}$	a domain alignment matrix for a user/item
$\mathbf{p'_t}$, $\mathbf{p'_s}$	a user's latent cluster membership in the target/source domain
$\mathbf{q'_t}$, $\mathbf{q'_s}$	an item's latent cluster membership in the target/source domain
$\mathbf{u'_t}$, $\mathbf{u'_s}$	a user domain-shareable vector for the target/source domain
$\mathbf{i'_t}$, $\mathbf{i'_s}$	an item domain-shareable vector for the target/source domain
$\mathbf{u''_t}$, $\mathbf{u''_s}$	a user domain-specific vector for the target/source domain
$\mathbf{i''_t}$, $\mathbf{i''_s}$	an item domain-specific vector for the target/source domain
$\hat{y}_{t\_ui}$, $\hat{y}_{s\_ui}$	the predicted serendipity score for the target/source domain
l_u, l_i	the number of latent user/item clusters
f	the dimension size of latent serendipity features
$\mathbf{i_t^{rel}}$, $\mathbf{i_s^{rel}}$	an item that is relevant to the user in the target/source domain
$\mathbf{i_t^{exp}}$, $\mathbf{i_s^{exp}}$	an item that is expected by the user in the target/source domain

$$\hat{y}_{t\_ui} = F_t(\mathbf{u_t}, \mathbf{i_t}), \tag{2}$$

$$\hat{y}_{s\_ui} = F_s(\mathbf{u_s}, \mathbf{i_s}), \tag{3}$$

where $\hat{y}_{t\_ui}$ and $\hat{y}_{s\_ui}$ denote the predicted serendipity score in the target domain D_t and the source domain D_s respectively. In order to learn the two matching functions at the same time, there should be a "bridge" element that connects and restricts the two matching functions. We will talk about the "bridge" element in the following paragraphs.

We believe each domain possesses some "essence" or latent clusters of its users and items. The "essence" could be shared among domains even though there are no overlapping users or items. In the meantime, each domain may have its own characteristics that belong to this domain only. For example, for the book domain and the movie domain, users in the two domains may have some common attributes that matter in their preferences, such as their age, gender, and cultural background. Users also have domain-specific attributes that are unique to that specific domain. Some book readers like annotating while reading. Some movie watchers prefer to watch movies with family and friends instead of on their own. On the item side, books and movies have common attributes such as genres and story plots. They also have their domain unique attributes. Books have publishers, authors, and writing styles. Movies have producers, directors, actors, etc.

Therefore, we could separate a user or an item of each domain into two independent parts: the domain-shareable part and the domain-specific part. The domain-shareable part captures the essence of the user or the item. The essence is the "bridge" knowledge transferable between different domains even without overlapping users or items in these domains. In contrast, domain-specific knowledge represents a user or an item's characteristics unique to that domain. The separation of the domain-shareable and domain-specific knowledge is expected to help transfer only useful information to assist in serendipity learning. To reflect the idea of the separation, a user's vector representation $\mathbf{u}$ will contain two parts: the domain-shareable vector $\mathbf{u'}$ and the domain-specific vector $\mathbf{u}''$. Similarly, an item's vector representation $\mathbf{i}$ has two parts: $\mathbf{i}'$ and $\mathbf{i}''$. Then the goal of the cross-domain serendipity recommendations becomes learning these two matching functions simultaneously:

$$\hat{y}_{t\_ui} = F_t((\mathbf{u}_\mathrm{t}', \mathbf{u}_\mathrm{t}''), (\mathbf{i}_\mathrm{t}', \mathbf{i}_\mathrm{t}'')), \tag{4}$$

$$\hat{y}_{s\_ui} = F_s((\mathbf{u}_\mathrm{s}', \mathbf{u}_\mathrm{s}''), (\mathbf{i}_\mathrm{s}', \mathbf{i}_\mathrm{s}'')), \tag{5}$$

To represent the idea of "essence" in the domain-shareable part, for $\mathbf{u}_\mathrm{t}'$, $\mathbf{i}_\mathrm{t}'$, $\mathbf{u}_\mathrm{s}''$, and $\mathbf{i}_\mathrm{s}''$, they can be written as:

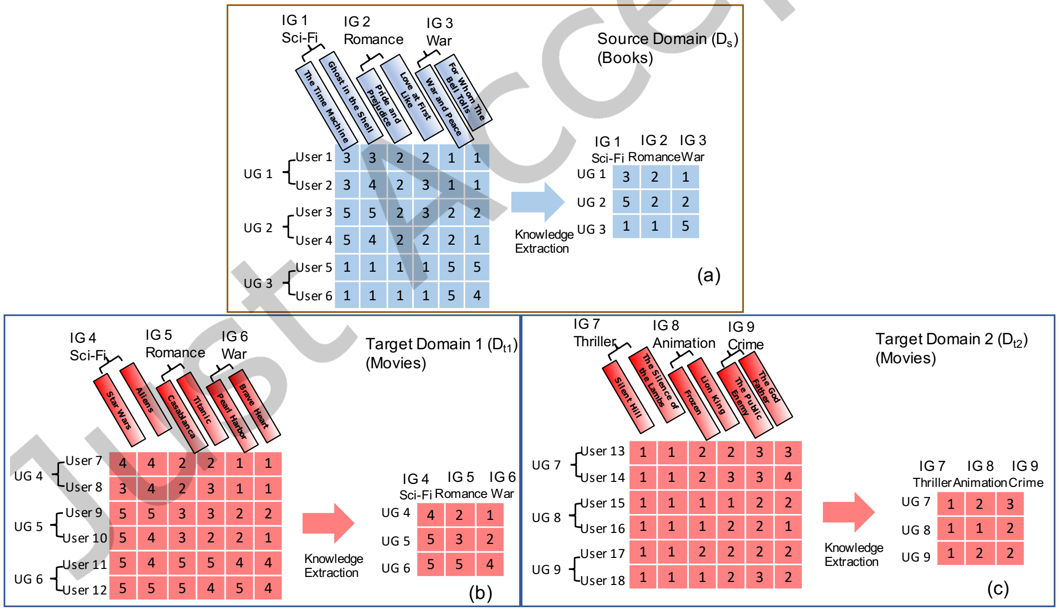
$$\mathbf{u}_\mathrm{t}' = \mathbf{p}_\mathrm{t}'\mathbf{U}_\mathrm{t}, \mathbf{i}_\mathrm{t}' = \mathbf{q}_\mathrm{t}'\mathbf{I}_\mathrm{t}, \tag{6}$$

$$\mathbf{u}_\mathrm{s}' = \mathbf{p}_\mathrm{s}'\mathbf{U}_\mathrm{s}, \mathbf{i}_\mathrm{s}' = \mathbf{q}_\mathrm{s}'\mathbf{I}_\mathrm{s}, \tag{7}$$

$$\mathbf{U}_\mathrm{t} = \mathbf{U}_\mathrm{s}\mathbf{T}^\mathrm{U}, \tag{8}$$

$$\mathbf{I}_\mathrm{t} = \mathbf{I}_\mathrm{s}\mathbf{T}^\mathrm{I}, \tag{9}$$

$\mathbf{U}_\mathrm{t}$ and $\mathbf{U}_\mathrm{s}$ represent the shareable user "essence" knowledge in the two domains, which could be computationally interpreted as a set of latent user clusters in each domain respectively. Each latent user cluster is a vector representing a series of latent features (aspects) of serendipity. Therefore both $\mathbf{U}_\mathrm{t}$ and $\mathbf{U}_\mathrm{s}$ are a two-dimensional matrix. Through a domain alignment matrix $\mathbf{T}^\mathrm{U}$, the latent clusters in the source domain could be mapped to those in the target domain. Similarly, $\mathbf{I}_\mathrm{t}$ and $\mathbf{I}_\mathrm{s}$ represent the shareable item "essence" knowledge in the two domains with a domain alignment matrix $\mathbf{T}^\mathrm{I}$. Therefore, $\mathbf{p}_\mathrm{t}'$ represents a specific user u_t's latent user cluster membership, and $\mathbf{q}_\mathrm{t}'$ represents a specific item i_t's latent item cluster membership in the target domain. Similarly, $\mathbf{p}_\mathrm{s}'$ and $\mathbf{q}_\mathrm{s}'$ represent their respective latent cluster membership in the source domain.



		The Time Machine	Ghost in the Shell	Pride and Prejudice	Love at First Like	War and Peace	For Whom The Bell Tolls
UG 1	User 1	3	3	2	2	1	1
	User 2	3	4	2	3	1	1
UG 2	User 3	5	5	2	3	2	2
	User 4	5	4	2	2	2	1
UG 3	User 5	1	1	1	1	5	5
	User 6	1	1	1	1	5	4

	IG 1 Sci-Fi	IG 2 Romance	IG 3 War
UG 1	3	2	1
UG 2	5	2	2
UG 3	1	1	5

		Star Wars	Aliens	Casablanca	Titanic	Pearl Harbor	Brave Heart
UG 4	User 7	4	4	2	2	1	1
	User 8	3	4	2	3	1	1
UG 5	User 9	5	5	3	3	2	2
	User 10	5	4	3	2	2	1
UG 6	User 11	5	4	5	5	4	4
	User 12	5	5	5	4	5	4

	IG 4 Sci-Fi	IG 5 Romance	IG 6 War
UG 4	4	2	1
UG 5	5	3	2
UG 6	5	5	4

		Silent Hill	The Silence of the Lambs	Frozen	Lion King	The Public Enemy	The God Father
UG 7	User 13	1	1	2	2	3	3
	User 14	1	1	2	3	3	4
UG 8	User 15	1	1	1	1	2	2
	User 16	1	1	1	2	2	1
UG 9	User 17	1	1	2	2	2	2
	User 18	1	1	1	2	3	2

	IG 7 Thriller	IG 8 Animation	IG 9 Crime
UG 7	1	2	3
UG 8	1	1	2
UG 9	1	2	2

Fig. 2. Examples to describe why we need domain alignment

The domain alignment matrices $\mathbf{T}^\mathrm{U}$ and $\mathbf{T}^\mathrm{I}$ are used to adjust the users and items "essence" in the source domain to be consistent with the target domain. Why do we need such domain alignment? Figure 2 serves as an example for describing the importance of domain alignment. Consider a book domain as the source domain D_s and a

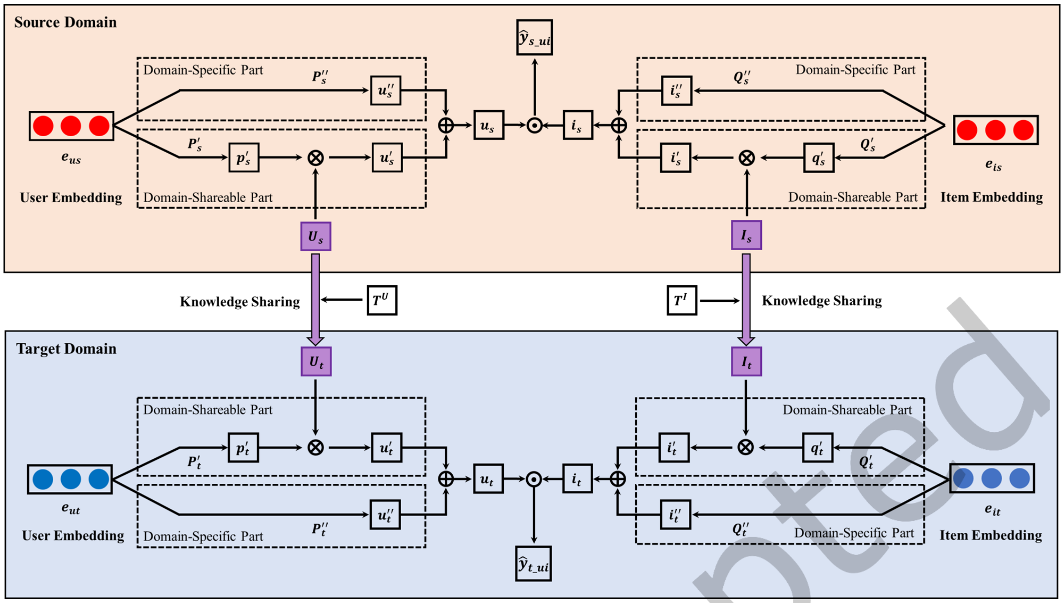


Fig. 3. The structure of the *SerenCDR* model

movie domain as a target domain D_{t1}. Figure 1(a) and Figure 1(b) illustrate the rating matrix for each of them. Users 1-4 in Figure 1(a) and Users 7-10 in Figure 1(b) have similar tastes for various genres, although the items are completely different (one set of items is books and the other is movies). Therefore, in this case, the latent user and item clusters (through knowledge extraction) contain consistent information in the D_s and D_{t1}. Transferring the latent cluster knowledge of D_s directly into D_{t1} is helpful without any need for domain alignment. However, Users 5-6 in Figure 2(a) and Users 11-12 in Figure 2(b) have opposite preferences in these genres. In this case, directly transferring the knowledge from D_s to D_{t1} will hurt the recommendation performance of D_{t1}.

Let us take a look at another rating matrix still in the movie domain as the target domain D_{t2}, as shown in Figure 2(c). The genres are completely different as D_s. On the user side, book readers in D_s and movie watchers in D_{t2} have different genre preferences. As a result, directly using knowledge from D_s for D_{t2} may hurt the recommendation performance even more. Therefore, we need a learnable domain alignment $\mathbf{T^U}$ to convert $\mathbf{U_s}$ to $\mathbf{U_t}$, and $\mathbf{T^I}$ to convert $\mathbf{I_s}$ to $\mathbf{I_t}$ in order to get consistent knowledge from the source domain to assist the target domain.

To sum up, the key challenge for the cross-domain recommendation problem becomes how to represent $\mathbf{u'}$, $\mathbf{u''}$, $\mathbf{i'}$, and $\mathbf{i''}$ in both source and target domains, how to identify the essence matrices $\mathbf{U_t}$, $\mathbf{U_s}$, $\mathbf{I_t}$ and $\mathbf{I_s}$, and the alignment matrices $\mathbf{T^U}$ and $\mathbf{T^I}$, as well as how to model the matching function F_t and F_s. The proposed *SerenCDR* model is to address all of these challenges.

3.2 The *SerenCDR* Model

As in Figure 3, the structure of the proposed *SerenCDR* model has two identical structures, one for the target domain and the other for the source domain. The two structures are connected through the shared knowledge on users (**U**) and items (**I**) with domain alignment mapping.

3.2.1 User and Item Representations for the Domain-Shareable and Domain-Specific Knowledge. *SerenCDR* converts a user u_t and an item i_t's pre-calculated representations $\mathbf{e_{ut}}$ and $\mathbf{e_{it}}$ in the target domain to the domain-shareable and domain-specific parts:

$$\mathbf{p'_t} = \mathbf{P'_t}\mathbf{e_{ut}}, \mathbf{q'_t} = \mathbf{Q'_t}\mathbf{e_{it}}, \tag{10}$$

$$\mathbf{u''_t} = \mathbf{P''_t}\mathbf{e_{ut}}, \mathbf{i''_t} = \mathbf{Q''_t}\mathbf{e_{it}}, \tag{11}$$

where $\mathbf{P'_t}$, $\mathbf{Q'_t}$, $\mathbf{P''_t}$, $\mathbf{Q''_t}$ are the learnable matrices. Similarly, we could obtain the user u_s and an item i_s's representations in the source domain: $\mathbf{p'_s}$, $\mathbf{q'_s}$, $\mathbf{u''_s}$, $\mathbf{i''_s}$. It is worth noting that the obtained representations for the domain shareable part of users or items are only the latent cluster membership representations. In order to recover the representation of the domain shareable users or items, we need to multiply the latent cluster information:

$$\mathbf{u'_t} = \mathbf{p'_t}\mathbf{U_t}, \mathbf{i'_t} = \mathbf{q'_t}\mathbf{I_t}, \tag{12}$$

$$\mathbf{u'_s} = \mathbf{p'_s}\mathbf{U_s}, \mathbf{i'_s} = \mathbf{q'_s}\mathbf{I_s}, \tag{13}$$

3.2.2 Fusion of Domain-Shareable and Domain-Specific Knowledge. There are many ways to fuse the domain-shareable and the domain-specific knowledge. One example is the method of making the concatenation of the two parts at the last hidden layer of the matching function F. That way, in the target domain, we have two sub-matching functions F'_t and F''_t for the domain-shareable and domain-specific parts respectively, and then combine them by concatenating their last hidden layers. The same method will be applied in the source domain. We tried different fusion approaches, and found the best approach is concatenating the two parts before the matching function F as shown in Figure 3. Specifically, in the target domain, we concatenated $\mathbf{u'_t}$ and $\mathbf{u''_t}$ as the final representation of the user $\mathbf{u_t}$. We also concatenated $\mathbf{i'_t}$ and $\mathbf{i''_t}$ as the final representation of the item $\mathbf{i_t}$. In the source domain, we made the same concatenations. The recommendation problem returns to learning the two matching functions simultaneously as in Equation 2 and 3.

For the matching function F_t or F_s, we could use any deep learning structure that calculates the interactions between $\mathbf{u_t}$ and $\mathbf{i_t}$, or $\mathbf{u_s}$ and $\mathbf{i_s}$. It could be a dot product operation, multiple MLP layers, or multiple CNN layers. We used dot product for its good performance and simplicity in this study.

4 The Main Loss Function and the Auxiliary Loss Function

To train *SerenCDR*, we used a main loss function applied on *SerenCDRLens*, the direct ground truth of serendipity. We also designed an auxiliary loss function to supplement the sparsity of serendipity in *SerenCDRLens* and to strengthen the serendipity learning process.

4.1 The Main Loss Function: Pairwise Learning from the Serendipity Ground Truth

Since our serendipity recommendation task is a personalized ranking task according to the predicted serendipity score, it is reasonable to assume that the observed serendipity should be ranked higher than the unobserved ones. To implement this idea, we adapted the well-established pairwise loss function: the Bayesian Personalized Ranking (BPR) [50] objective function, into our cross-domain recommendation tasks:

$$L_{main_t}(\Theta_t) = \sum_{(u_t,i_t,j_t)\in\mathcal{D}_t} -ln\sigma(\hat{y}_{t\_ui} - \hat{y}_{t\_uj}) + \lambda_t||\Theta_t||^2, \tag{14}$$

$$L_{main_s}(\Theta_s) = \sum_{(u_s,i_s,j_s)\in\mathcal{D}_s} -ln\sigma(\hat{y}_{s\_ui} - \hat{y}_{s\_uj}) + \lambda_s||\Theta_s||^2, \tag{15}$$

$$L_{main}(\Theta_t, \Theta_s) = L_{main_t}(\Theta_t) + L_{main_s}(\Theta_s), \tag{16}$$

where Θ_t and Θ_s are the sets of model parameters of the target domain and the source domain respectively. λ_t and λ_s are the parameter-specific regulation hyper-parameters to prevent overfitting. $\mathcal{D}_t$ denotes the set of training instances in the target domain: $\mathcal{D}_t := \{(u_t, i_t, j_t) | i_t \in \mathcal{Y}^+_{t\_u} \wedge j_t \in \mathcal{Y}^-_{t\_u}\}$, where $\mathcal{Y}^+_{t\_u}$ and $\mathcal{Y}^-_{t\_u}$ denotes the set of items that has been regarded by the user u_t as serendipitous (positive samples) or non-serendipitous (negative samples) respectively. Similarly, $\mathcal{D}_s$ denotes the set of training instances in the source domain.

By minimizing the BPR loss, we tailored the model for correctly predicting the relative orders between items rather than their absolute serendipity scores as optimized in pointwise loss. This can be more beneficial for addressing the serendipity recommendation task where serendipity positive cases are relatively rare.

4.2 The Auxiliary Loss Function: Strengthen Serendipity Learning

Since we observed that serendipity is sparse in *SerenCDRLens*, as expected due to the fact that serendipity rarely happens in real life. In order to strengthen the ability of *SerenCDR* model to learn the serendipity "signal", we further designed an auxiliary loss function to supplement the main loss function. As discussed in the Introduction section, the two facets of serendipity are unexpectedness and relevance. Therefore, the auxiliary loss function aims at providing additional serendipity "signals" from these two facets. Specifically, L_{aux} tries to pull a user away from the expected items and meanwhile makes sure he or she is not too far away from the relevant items. Computationally, L_{aux} is to learn a user representation that minimizes its similarity with the user's expected items, as well as maintains the similarity with the relevant items:

$$L_{aux_t}(\Delta_t) = \sum_{(\mathbf{u_t}, \mathbf{i_t^{rel}}, \mathbf{i_t^{exp}}) \in \mathcal{D}'_t} -ln \frac{\sigma(\mathbf{u_t}^T \mathbf{i_t^{rel}})}{\sigma(\mathbf{u_t}^T \mathbf{i_t^{exp}})}, \tag{17}$$

$$L_{aux_s}(\Delta_s) = \sum_{(\mathbf{u_s}, \mathbf{i_s^{rel}}, \mathbf{i_s^{exp}}) \in \mathcal{D}'_s} -ln \frac{\sigma(\mathbf{u_s}^T \mathbf{i_s^{rel}})}{\sigma(\mathbf{u_s}^T \mathbf{i_s^{exp}})}, \tag{18}$$

$$L_{aux}(\Delta_t, \Delta_s) = L_{aux_t}(\Delta_t) + L_{aux_s}(\Delta_s), \tag{19}$$

where Δ_t and Δ_s are the sets of model parameters of the target domain and the source domain respectively. $\mathcal{D}'_t$ denotes the set of training instances in the target domain: $\mathcal{D}'_t := \{(\mathbf{u_t}, \mathbf{i_t^{exp}}, \mathbf{i_t^{rel}}) | \mathbf{i_t^{exp}} \in \mathcal{E}_{t\_u} \wedge \mathbf{i_t^{rel}} \in \mathcal{R}_{t\_u}\}$, where $\mathcal{E}_{t\_u}$ denotes the items that are expected by the user $\mathbf{u_t}$ and $\mathcal{R}_{t\_u}$ is the set of items that are relevant to the user $\mathbf{u_t}$. Similarly, $\mathcal{D}'_s$ denotes the set of training instances in the source domain.

The key problem becomes how to obtain the ground truth for the expected items and the relevant items. Rather than relying on a human labeling process, we leveraged the calculated or observed labels from the data itself. For the expected items, we computationally define it as the high conditional likelihood of seeing an item given the user's history. Let $I = \{i_1, i_2, \ldots, i_{|I|}\}$ denotes the set of items in the data, and $T_u = \{i^u_1, i^u_2, \ldots, i^u_n\}$ denotes the user u's history, equivalent to a sequence of interacted items by the user u. The level of expectedness for an item is calculated as:

$$exp_{(u,i)} = log\, p(i|T_u), \tag{20}$$

The logarithm function is to smooth the larger values. Using the Law of Total Probability, we could rewrite the conditional probability in Equation 20 as:

$$exp_{(u,i)} = log\, p(i|T_u) = log \sum_{i^u_h \in T_u} p(i|i^u_h) p(i^u_h), \tag{21}$$

where i_h^u is a user's historically interacted item, and $p(i|i_h^u)$ could be calculated as:

$$p(i|i_h^u) = \frac{n(i, i_h^u)}{\sum_{i \in I} n(i, i_h^u)}, \tag{22}$$

$n(i, i_h^u)$ is the count of co-occurrences for an item i and i_h^u in the dataset. The denominator is the sum of the co-occurrence counts for each item in the item set with i_h^u.

Therefore, the level of expectedness $exp_{(u,i)}$ is calculated as:

$$exp_{(u,i)} = log\, p(i|T_u) = log \sum_{i_h^u \in T_u} \frac{n(i, i_h^u)}{\sum_{i \in I} n(i, i_h^u)} p(i_h^u), \tag{23}$$

All of the components on the right side of this Equation 23 could be pre-calculated from the dataset before the model training. After calculating all the items' $exp_{(u,i)}$ values for a user u, we selected the items with the top values as the user's expected items and used in the auxiliary loss functions.

For relevant items, we followed the common practice and defined them as the items interacted by the user in the past history.

At last, we jointly trained the proposed *SerenCDR* model with the two loss functions in each domain simultaneously.

$$L = L_{main}(\Theta_t, \Theta_s) + L_{aux}(\Delta_t, \Delta_s), \tag{24}$$

4.3 Convergence Analysis

In classical matrix factorization (MF) approach, several existing studies (e.g., Zhang et al. [62]; Yu et al. [59]) have proved that there exist non-negative matrices that simultaneously minimize two or multiple rating matrix reconstruction loss. In this study, we essentially propose a neural network-based MF approach. It remains unclear if the convergence will happen to the proposed model. Our hypothesis is that *SerenCDR* will experience similar convergence process guaranteed in the classical MF cases. We tested the convergence empirically. Specifically, we trained the model iteratively for n epochs until the change of loss function was less than a threshold. We plotted the training loss over the number of epochs. We observed whether *SerenCDR* was able to stabilize eventually and was able to outperform the baseline models after a certain number of epochs.

5 Experiments

5.1 *SerenCDRLens*: the Two-Domain Ground Truth Data on Serendipity

Currently, there are only two publicly available serendipity ground truth datasets. One is *Serendipity 2018* [25], which was collected using a small-scale survey with 481 participants in the movie domain. The other one is *SerenLens* [12], which provides a relatively reasonable scale of ground truth data in the book domain. However, currently there is not any multi-domain serendipity ground truth dataset. Therefore, in this study, we collected a two-domain serendipity ground truth dataset, using the *URCW (User Reviews plus Crowd Wisdom)* approach in [12], a scalable approach by using both user generated reviews and a crowd sourcing method. It is originally for collecting serendipity ground truth in a single domain. We applied *URCW* in two domains.

The two chosen domains are books and movies. Both book-reading behavior and movie-watching behavior are highly driven by a personal taste [46], and the experiences are highly subjective, laying a rich ground for serendipity occurrences. In addition, both book and movies reviews datasets are largely available. The scale and richness of the dataset warrant the high quality of the ground truth dataset. We used the Amazon Review Data [37] as the review corpus used in the first stage of *URCW*. In the second stage, we used Amazon Mechanical Turk (MTurk) to reach crowd workers to conduct our human intelligence tasks (HITs). The final two-domain serendipity ground truth data is called *SerenCDRLens*, as compared to the well-known dataset for recommendation models, *MovieLens* [17]. The *SerenCDRLens* dataset is publicaly available at https://github.com/zhefu2/SerenCDR. Table 2 shows the key statistics of the data collection process and the *SerenCDRLens* dataset. In the book domain, 2,557 reviews were labeled as serendipity experiences, which were written by 2,346 users (review writers) on 2,227 books. In the movie domain, 714 reviews were labeled as serendipity experiences, which were written by 619 users on 634 movies.

Table 2. Key statistics of *SerenCDRLens*

		Books	**Movies & TV**
HITs (Tasks)	Total HIT tasks assigned in MTurk	8,268	4,427
	Total HIT tasks accepted in MTurk	4,396	2,342
Worker Judgements	Total worker judgements collected	41,340	22,135
	Total worker judgements accepted	21,980	11,710
	Judgements with initial agreement	16,040	10,985
	Judgements need a third opinion	3,960	725
	Degree of initial agreement	72.98%	93.81%
	Reviews with judgements	10,000	5,000
	Reviews of serendipity	2,557	714
	Reviews of non-serendipity	7,443	4,286
SerenCDRLens **Dataset**	Users involved in the reviews of serendipity	2,346	619
	Total reviews involved	**265,037**	**74,967**
	Items involved in the reviews of serendipity	2,227	634
	Total items involved	**113,876**	**23,950**

5.2 Evaluation Metrics and Baseline Models

Since serendipity is sparse ($2,557/265,037 \approx 1.0\%$ in books and $634/74,967 \approx 0.8\%$ in movies) in *SerenCDRLens*, we adopted a recall-based metric, Hit Ratio (HR). $\mathrm{HR}_{\mathrm{seren}}$@k measures the proportion of times a serendipity item is retrieved in the top-k position (each time 1 for yes and 0 otherwise). In order to take the rank information into consideration and assign higher weights on higher ranks, we propose another metric called Serendipity-Based Normalized Discounted Cumulative Gain ($\mathrm{NDCG}_{\mathrm{seren}}$) based on the well-known metric Normalized Discounted Cumulative Gain (NDCG). $\mathrm{NDCG}_{\mathrm{seren}}$@k is calculated as:

$$NDCG_{seren}@k = \sum_{i=1}^{k} \frac{serendipity\ score(1\ or\ 0)}{log_2(i+1)} \tag{25}$$

In addition to evaluating serendipity, we conducted a second set of experiments to evaluate how much sacrifice on relevance *SerenCDR* was making (if there was) in order to accommodate serendipity. Therefore, we used the two standard metrics for relevance evaluation: HR@k (the "hit" here means hitting a relevant item in the top-k position) and NDCG. For all of the metrics above, a higher value indicates a better performance.

To evaluate the performance of the proposed models, we selected the following three groups of representative baseline recommendation models. The first group consists of the well-known deep learning models for single-domain recommendations: **NCF** [20], **BERT4Rec** [53]. We trained them on *SerenCDRLens* in the target domain only and compared their performances with **SerenCDR**. The second group is the recent single-domain serendipity deep learning recommendation models: **SerRec** [48], **PURS** [31], and **SNPR** [61]), both of which were also trained on *SerenCDRLens* in the target domain only and were compared with **SerenCDR**. The third group is the state-of-the-art cross-domain models that accommodate the scenarios of non-overlapping users and non-overlapping items: **CIT** [62], **MHF** [59], and **CFAA** [36]. All of the three groups of models are the state-of-the-art recommendation models published in top venues in recent years.

5.3 *SerenCDR* Settings

SerenCDR is a general framework and we may use many base recommendation models for the "encoding" purpose to represent a user and an item into vector representations, as the input of *SerenCDR*. We chose two classic recommendation models to generate user and item's pre-calculated representations e_u and e_i: matrix factorization (MF) [6] and BERT [24]. For MF, we pre-calculated the user and item representations by decomposing the rating matrix. For BERT, we pre-calculated the user and item representations by encoding the user comments and item comments (texts). Therefore, *SerenCDR* has two variants: *SerenCDR-BERT* and *SerenCDR-MF*.

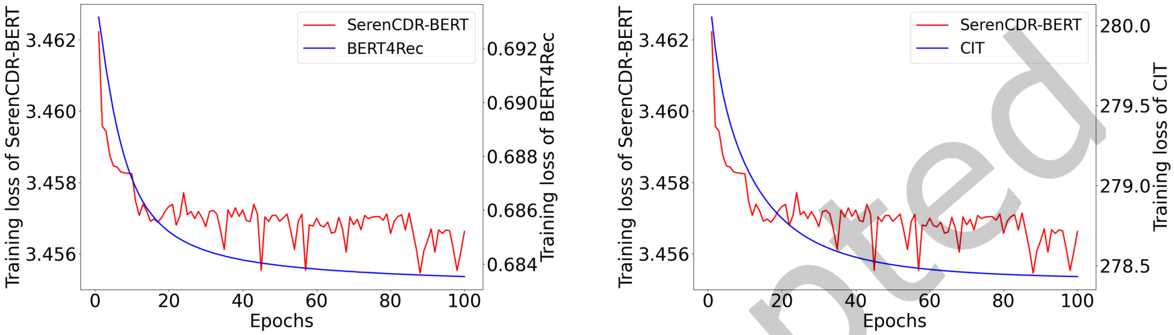
We trained the proposed model, both variants of *SerenCDR*, and the baseline models on the *SerenCDRLens* data. The codes of the models are available at https://github.com/zhefu2/SerenCDR. We compared the two variants of *SerenCDR* and the baseline models using the HR@k and NDCG$_{\text{seren}}$@k. Since the amount of reviews collected on the books is larger than that on movies, we define the movie domain as the target domain and the books as the source domain in this study. We keep our minds open to using books as the target domain and movies as the source domain in future studies. For *SerenCDR* and other cross-domain baseline models, 80% of the data in *SerenCDRLens-Movies* (the target domain) and all of the data in *SerenCDRLens-Books* (the source domain) were used for training, and the rest 20% of the *SerenCDRLens-Movies* was for testing. For those single-domain baseline models, we did not use the books data (the source domain). 80% of the data in *SerenCDRLens-Movies* (the target domain) was used for training and the rest 20% was for testing. For the second set of relevance-oriented experiments, we used the same training dataset to train the models, as the first set of serendipity-oriented experiments, but for the testing set, the ground truth was changed to relevance, available in the original Amazon Review Data [37].

For all models, we adopted 5-fold cross-validation approach to evaluate the performance and reported the average value across these 5 folds. We trained our models using the Adam optimizer. We set the learning rate 0.0001, the dimension 128 for the pre-trained user and item representations ($\mathbf{e_{ut}}$, $\mathbf{e_{it}}$, $\mathbf{e_{us}}$, and $\mathbf{e_{is}}$), the dropout rate 0.2, and the regularizer decay 0.001 for all the models. Other model-specific hyper-parameters were set either following their original studies or adjusting for the training performance in this study. For all of the baseline models, we reported the results using the optimal hyper-parameter settings. For the proposed *SerenCDR* model, the optimal values of the number of latent user clusters l_u (equivalent to the number of rows in $\mathbf{U_t}$ or $\mathbf{U_s}$), the number of latent item clusters l_i (equivalent to the number of rows in $\mathbf{I_t}$ or $\mathbf{I_s}$), and the dimension size of the latent serendipity features f (equivalent to the number of columns in $\mathbf{U_t}$ or $\mathbf{U_s}$ or $\mathbf{I_t}$ or $\mathbf{I_s}$) will be investigated and discussed in the experiments.

6 Results

6.1 Convergence Results

We first conducted a convergence analysis on our proposed model. Figure 4 presents three models' training loss over the number of epochs. The three models are **BERT4Rec**, **CIT**, and **SerenCDR-BERT**. **SerenCDR-MF** has a very similar curve with **SerenCDR-BERT**, and therefore is omitted here. Importantly, our proposed **SerenCDR-BERT** is able to converge after around 40 epochs of training, although **BERT4Rec** (a single-domain model) and **CIT** (a cross-domain model)'s convergence curves are smoother, suggesting better convergences due to their relatively less complex structures. **SerenCDR-BERT** has two objectives in the training process. One is the main loss for the direct serendipity knowledge learning and the other one is the auxiliary loss for the decomposed serendipity knowledge learning. Optimizing the two objectives simultaneously on two domains needs more effort and time to converge compared with single-domain or single-objective models.



(a) *SerenCDR-BERT* and BERT4Rec

(b) *SerenCDR-BERT* and CIT

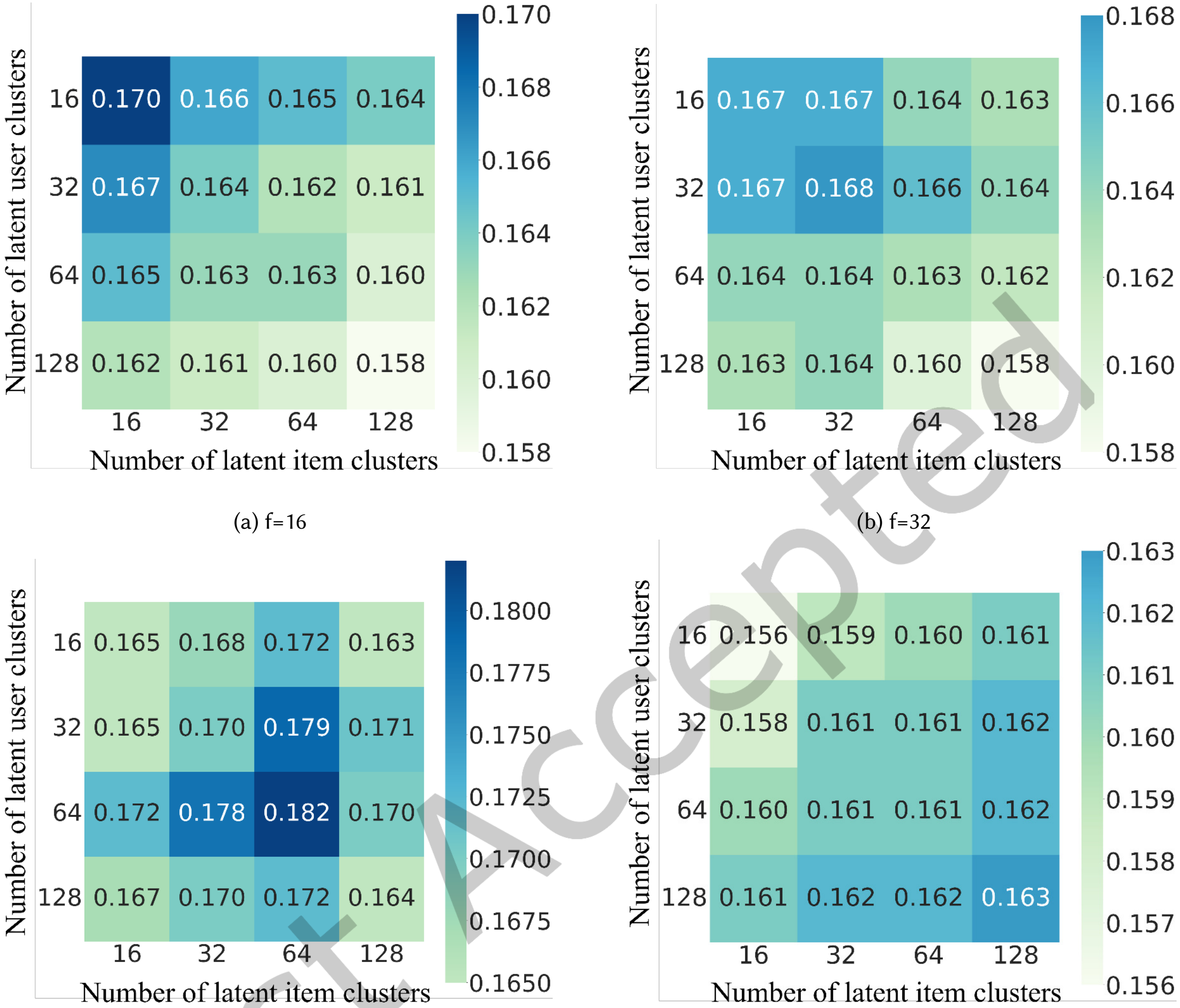
Fig. 4. The training epoch loss of *SerenCDR-BERT* and two baseline models

6.2 Effects of Hyper-Parameters

We investigated the impact of the number of latent user clusters (l_u), the number of latent item clusters (l_i), and the dimension size of the latent serendipity features f on the recommendation performance. Since **SerenCDR-BERT** and **SerenCDR-MF** have similar performance trends for these different hyper-parameters, we reported **SerenCDR-BERT** performance in terms of NDCG$_{\text{seren}}$@10 in Figure 5. If f is relatively small (e.g., 16), setting a large value for l_u and l_i (e.g. 128) dramatically decreases the model performance. On the other side, if f is relatively large (e.g., 128), setting a small value for l_u and l_i (e.g. 16) hurts the model performance too. From each of the four heat maps with a fixed f value, we know making the values of l_u and l_i the same with f has achieved the best performance. Those observations mean the granularity of latent user or item clusters should be aligned with the granularity of the serendipity features during the knowledge transferring process. The best performance is achieved when $\mathbf{U}_\text{t}$, $\mathbf{U}_\text{s}$, $\mathbf{I}_\text{t}$, and $\mathbf{I}_\text{s}$ are all square matrices with the dimension of 64. In the following sections, we only reported the experiment results of both **SerenCDR-MF** and **SerenCDR-BERT** with l_u=64, l_i=64, and f=64.

6.3 Overall Performance Comparison

The **SerenCDR-MF** and **SerenCDR-BERT** performances and the comparison with other baseline models on the serendipity recommendation task are presented in Table 3. **SerenCDR-BERT** achieves the best performance for both $HR_{seren}@k$ and NDCG$_{\text{seren}}$@k at varying k levels, comparing with the single-domain models, either the relevance-oriented models (**NCF** and **BERT4Rec**) or the serendipity-oriented models (**SerRec**, **PURS** and **SNPR**). The result suggests the effectiveness of the transferred knowledge from the source domain. Compared with the cross-domain models (**CIT**, **MHF**, and **CFAA**) that are based on the classical matrix factorization (MF) or content-based filtering method, the dramatic performance improvement of **SerenCDR-BERT** speaks to the



(a) f=16

(b) f=32

(c) f=64

(d) f=128

Fig. 5. Heat maps of NDCG$_{seren}$@10 for different numbers of latent user clusters (l_u), numbers of latent item clusters (l_i), and different dimension sizes of latent serendipity features (f) for *SerenCDR*

power of the deep learning techniques in transferring the knowledge across domains without any overlapping users or overlapping items.

It is worth pointing out that both **BERT4Rec** and **SNPR** are single-domain recommendation models but both perform better than the three cross-domain baseline models (**CIT**, **MHF**, and **CFAA**) in serendipity task, suggesting the deep learning models' stronger capacity to effectively identify serendipity in a single domain than the MF-based cross-domain methods.

Table 3. The performance comparison of different models on the serendipity recommendation task. The reported number is the average of 5 folds. The best results in each column are bolded and the second best results are underlined. $*$ denotes that our proposed model has statistically significant differences with all of the six baseline models under a two-tailed t-test with $p < 0.05$.

		Serendipity-based metrics				
		HR_{seren}@1	HR_{seren}@5	HR_{seren}@10	$NDCG_{seren}$@5	$NDCG_{seren}$@10
Single-domain	NCF	1.46%	5.04%	9.59%	0.032	0.046
	BERT4Rec	5.37%	15.28%	20.65%	0.106	0.129
	SerRec	3.41%	10.57%	18.21%	0.085	0.105
	PURS	2.44%	6.67%	11.54%	0.044	0.057
	SNPR	4.27%	16.59%	23.90%	0.096	0.112
Cross-domain	CIT	2.28%	11.22%	19.67%	0.067	0.094
	MHF	2.42%	10.18%	13.09%	0.062	0.071
	CFAA	3.25%	12.89%	17.64%	0.095	0.111
	SerenCDR - MF	7.32%*	17.07%*	31.71%*	0.148*	0.162*
	SerenCDR - BERT	**8.07%***	**18.82%***	**35.89%***	**0.157***	**0.182***

Table 4. The performance comparison of different models on the relevance recommendation task. The reported number is the average of 5 folds. The best results in each column are bolded and the second best results are underlined. $*$ denotes that our proposed model has statistically significant differences with all of the six baseline models under a two-tailed t-test with $p < 0.05$.

		Relevance-based metrics				
		HR@1	HR@5	HR@10	NDCG@5	NDCG@10
Single-domain	NCF	2.12%	6.23%	11.49%	0.034	0.053
	BERT4Rec	2.47%	6.52%	12.75%	0.036	0.055
	SerRec	1.38%	5.56%	10.42%	0.027	0.043
	PURS	1.68%	5.23%	9.70%	0.024	0.043
	SNPR	1.92%	5.82%	10.51%	0.029	0.051
Cross-domain	CIT	2.54%	6.64%	11.32%	0.032	0.051
	MHF	3.29%	6.25%	12.25%	0.035	0.056
	CFAA	1.63%	5.45%	12.83%	0.022	0.052
	SerenCDR - MF	3.25%	8.13%*	**13.01%***	0.046*	0.070*
	SerenCDR - BERT	**4.07%***	**9.76%***	12.20%	**0.051***	**0.071***

As to relevance, the comparison results with other baseline models are presented in Table 4. Surprisingly, either **SerenCDR-BERT** or **SerenCDR-MF** performs better than the baseline models, including those relevance-oriented models (i.e. **NCF**, **BERT4Rec**, **CIT**, **MHF**, and **CFAA**). This indicates that the process of predicting serendipity, with the cross-domain knowledge transfer and the auxiliary loss functions, can in fact boost the process of predicting relevance, a traditional task for recommender systems.

6.4 An Ablation Study

To evaluate the effectiveness of the key components of our proposed models, we further conducted a series of ablation analyses. The key components to be evaluated are the *Domain-Specific Part*, the *Domain-Shareable Part*, the *Domain Alignment Matrices* ($\mathbf{T^U}$ and $\mathbf{T^I}$), the L_{main} loss function, and the L_{aux} loss function. Please refer to Figure 3 and Section 4 for these components. The evaluations were conducted by removing each component at one time to test the model performance change. Please note that removing the *Domain-Specific Part* makes the cross-domain mechanism share the entire knowledge from one domain to another without preserving the domains' unique knowledge. Removing the *Domain-Shareable Part* downgrades the model to a single-domain recommendation model, only relying on the domain-specific knowledge in the target domain. Removing the *Domain Alignment Matrices* ($\mathbf{T^U}$ and $\mathbf{T^I}$) ends up with transferring domain-shareable knowledge directly from the source domain to the target domain without considering any domain alignment. Removing the L_{main} loss function results in *SerenCDR* model merely being trained on the relevant items and expected items without using the direct serendipity ground truth data in *SerenCDRLens*. Removing the L_{aux} loss function makes the model only trained on *SerenCDRLens*. The results of the ablation analyses for **SerenCDR-BERT**, the better performed variant, are presented in Table 5.

As in Table 5, without either the *Domain-Shareable Part* or the L_{main} loss function, the model performance drops dramatically (more than 30%) for both HR$_{\text{seren}}$@k and NDCG$_{\text{seren}}$@k. The result demonstrates the effectiveness of sharing knowledge across different domains and directly learning knowledge from serendipity ground truth data in *SerenCDRLens*. In addition, it can be observed that without the *Domain-Specific Part* or the alignment matrices ($\mathbf{T^U}$ and $\mathbf{T^I}$), the performance also drops around 25% for both HR$_{\text{seren}}$@k and NDCG$_{\text{seren}}$@k metrics, suggesting the positive role of domain-specific attributes and domain alignment. Furthermore, without the L_{aux} objective function, the performance also drops around 20% for both HR$_{\text{seren}}$@k and NDCG$_{\text{seren}}$@k metrics, indicating our proposed auxiliary loss function helps enhance the serendipity learning process.

Table 5. The ablation study results on different model variants. The reported number is the average of 5 folds. The best results in each column are bolded. "↓" indicates a performance drop more than 30% relative to that of the original **SerenCDR-BERT** model.

	HR$_{\text{seren}}$@1	HR$_{\text{seren}}$@5	HR$_{\text{seren}}$@10	NDCG$_{\text{seren}}$@5	NDCG$_{\text{seren}}$@10
SerenCDR-BERT	**8.07%**	**18.82%**	**35.89%**	**0.157**	**0.182**
W/O *Domain-Specific Part*	4.88%↓	14.63%	26.02%	0.103↓	0.124↓
W/O *Domain-Shareable Part*	3.85%↓	12.38%↓	23.58%↓	0.089↓	0.113↓
W/O $\mathbf{T^U}$ & $\mathbf{T^I}$	6.07%↓	15.31%	27.34%	0.112	0.130
W/O L_{main}	2.43%↓	8.94%↓	13.82%↓	0.041↓	0.062↓
W/O L_{aux}	6.67%	17.72%	31.38%	0.136	0.157

6.5 A Case Study

To have an intuitive understanding of the model results, we selected an example user to showcase the **SerenCDR-BERT** recommendation results compared to those generated by **SNPR**, one of the best baseline models. As shown in Figure 6, a user in the movie domain is interested in comedy, children's, romance, and action movies as shown in his or her profile record. We also find that this user had a serendipity experience on finding a movie titled *The Dresden Files*, a fantasy movie, as in his or her written review:

"*I came across this by accident - I had never heard of the show nor the books… After viewing the movie, I am interested in reading the books.*"

As shown in Figure 6, the top-5 recommended movies by **SNPR** are of the genres of horror, action, romance, and history. **SNPR** failed to hit the serendipity movie, *The Dresden Files*. These recommendations generally followed the user's profile record but also deviated a bit with horror and history movies, making attempts for serendipitous recommendations.

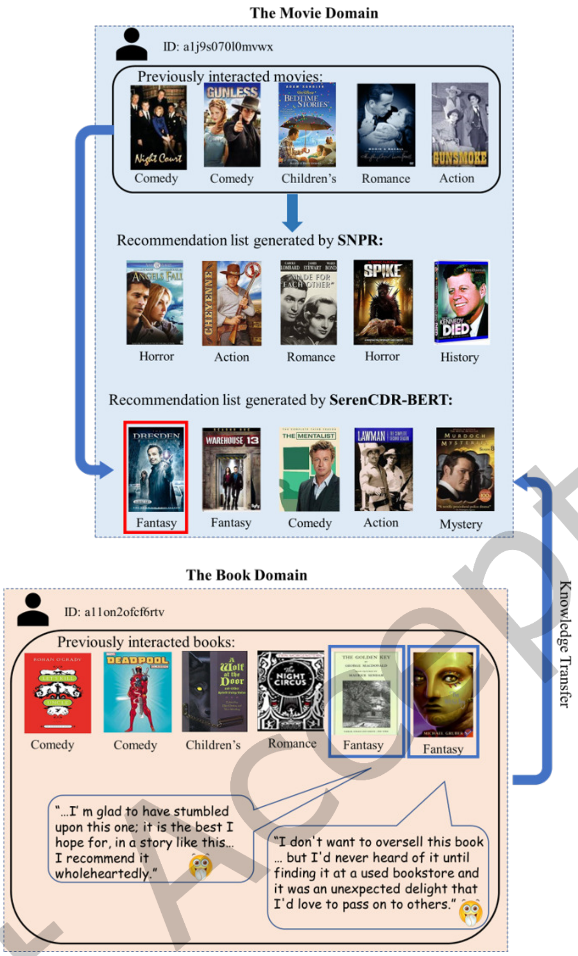

The Movie Domain
ID: a1j9s070l0mvwx
Previously interacted movies:
Comedy
Comedy
Children's
Romance
Action
Recommendation list generated by SNPR:
Horror
Action
Romance
Horror
History
Recommendation list generated by SerenCDR-BERT:
Fantasy
Fantasy
Comedy
Action
Mystery
Knowledge Transfer
The Book Domain
ID: a1lon2ofcf6rtv
Previously interacted books:
Comedy
Comedy
Children's
Romance
Fantasy
Fantasy
"...I' m glad to have stumbled upon this one; it is the best I hope for, in a story like this... I recommend it wholeheartedly."
"I don't want to oversell this book ... but I'd never heard of it until finding it at a used bookstore and it was an unexpected delight that I'd love to pass on to others."

Fig. 6. Top-5 recommendations generated by **SNPR** (a single-domain model) and **SerenCDR-BERT** (our proposed cross-domain model)

In contrast, our proposed **SerenCDR-BERT** successfully hit *The Dresden Files* as the top recommendation. Looking into the data beyond the movie domain, in the book domain, we found another user who had a similar taste with the user in the movie domain. They both like comedy, children's, and romance genres. We also found this book user's two pieces of reviews, each describing his or her serendipity experience in finding a fantasy book. This book domain knowledge may have inspired **SerenCDR-BERT** to look for serendipity in the fantasy movies for the other user. This way, the proposed **SerenCDR-BERT**, mitigates the imitations of one single domain and extends the knowledge using another domain without relying on the same user or the same item.

In addition, the ending part of the movie user's review, "*After viewing the movie, I am interested in reading the books.*" confirms that in real life, people consciously or unconsciously transfer their likes and dislikes between books and movies, or other medium types. Our **SerenCDR-BERT** was modeling that aspect of reality.

7 Conclusion

This paper contributes a novel cross-domain recommendation model for serendipity called *SerenCDR*, with main and auxiliary loss functions for serendipity learning. In addition, we also collected a new ground truth dataset for serendipity in two domains, called *SerenCDRLens*. Both serendipity-based and relevance-based experiments were conducted. Three groups of the state-of-the-art baseline models were implemented to compare with *SerenCDR*. Extensive experimental results show that *SerenCDR*, without relying on overlapping users or items in different domains, outperforms the state-of-the-art baseline models in predicting serendipity. We analyzed the effectiveness of different model components and also used a case study to demonstrate how our proposed *SerenCDR* works for serendipity recommendations.

For future works, we are interested in extending this study in three ways. First, this study collected serendipity data from books and movies. In the rich Amazon Review Data [37], many other domains are available, such as music, electronics, clothing, etc. It would be interesting to explore how *SerenCDR* is extended to other two domain pairs (e.g., music and electronics) or a multi-domain (more than two domains) scenario. Second, with the recent development of AI and advent of large language models (LLMs), we are planning to make use of LLMs to address the challenges for both serendipity ground truth generation and recommendation model development.

Third, this study did not consider the impact of item temporal order in serendipity recommendations. In the future, we will further investigate the potential order effect on the performance of cross-domain serendipity recommendations.

References

[1] Fakhri Abbas and Xi Niu. 2019. Computational serendipitous recommender system frameworks: A literature survey. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 1–8.

[2] Fakhri Abbas and Xi Niu. 2019. One size does not fit all: Modeling users' personal curiosity in recommender systems. *ArXivorg* (2019).

[3] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. *Recommender systems handbook* (2015), 919–959.

[4] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[5] Xu Chen, Ya Zhang, Ivor W Tsang, Yuangang Pan, and Jingchao Su. 2023. Toward Equivalent Transformation of User Preferences in Cross Domain Recommendation. *ACM Transactions on Information Systems* 41, 1 (2023), 1–31.

[6] Andrzej Cichocki and Anh-Huy Phan. 2009. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences* 92, 3 (2009), 708–721.

[7] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *2011 IEEE 11th International Conference on Data Mining Workshops*. Ieee, 496–503.

[8] Xiangyu Fan and Xi Niu. 2018. Implementing and evaluating serendipity in delivering personalized health information. *ACM Transactions on Management Information Systems (TMIS)* 9, 2 (2018), 1–19.

[9] Zhe Fu and Xi Niu. 2023. Modeling Users' Curiosity in Recommender Systems. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), 1–23.

[10] Zhe Fu and Xi Niu. 2024. The Art of Asking: Prompting Large Language Models for Serendipity Recommendations. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*. 157–166.

[11] Zhe Fu, Xi Niu, and Mary Lou Maher. 2023. Deep learning models for serendipity recommendations: a survey and new perspectives. *Comput. Surveys* 56, 1 (2023), 1–26.

[12] Zhe Fu, Xi Niu, and Li Yu. 2023. Wisdom of Crowds and Fine-Grained Learning for Serendipity Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 739–748.

[13] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain recommendation without sharing user-relevant data. In *The world wide web conference*. 491–502.

[14] Kazjon Grace, M Maher, Douglas Fisher, and Katherine Brady. 2014. A data-intensive approach to predicting creative designs based on novelty, value and surprise. *International Journal of Design, Creativity, and Innovation* (2014).

[15] Ted Gup. 1998. Technology and the End of Serendipity. *The Education Digest* 63, 7 (1998), 48.

[16] Zhongxuan Han, Xiaolin Zheng, Chaochao Chen, Wenjie Cheng, and Yang Yao. 2023. Intra and Inter Domain HyperGraph Convolutional Network for Cross-Domain Recommendation. In *Proceedings of the ACM Web Conference 2023*. 449–459.

[17] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[18] Ming He, Jiuling Zhang, Peng Yang, and Kaisheng Yao. 2018. Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 225–233.

[19] Ming He, Jiuling Zhang, and Shaozong Zhang. 2019. ACTL: Adaptive codebook transfer learning for cross-domain recommendation. *IEEE Access* 7 (2019), 19539–19549.

[20] 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*. 173–182.

[21] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[22] Jizhou Huang, Shiqiang Ding, Haifeng Wang, and Ting Liu. 2018. Learning to recommend related entities with serendipity for web search users. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17, 3 (2018), 1–22.

[23] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016), 1–42.

[24] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. 2.

[25] Denis Kotkov, Joseph A Konstan, Qian Zhao, and Jari Veijalainen. 2018. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd annual acm symposium on applied computing*. 1341–1350.

[26] Youfang Leng, Li Yu, and Xi Niu. 2022. Dynamically aggregating individuals' social influence and interest evolution for group recommendations. *Information Sciences* 614 (2022), 223–239.

[27] Bin Li. 2011. Cross-domain collaborative filtering: A brief survey. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*. IEEE, 1085–1086.

[28] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Twenty-First international joint conference on artificial intelligence*.

[29] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*. 617–624.

[30] Huiyuan Li, Li Yu, Xi Niu, Youfang Leng, and Qihan Du. 2023. Sequential and graphical cross-domain recommendations with a multi-view hierarchical transfer gate. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), 1–28.

[31] Pan Li, Maofei Que, Zhichao Jiang, Yao Hu, and Alexander Tuzhilin. 2020. PURS: personalized unexpected recommender system for improving user satisfaction. In *Fourteenth ACM Conference on Recommender Systems*. 279–288.

[32] Pan Li and Alexander Tuzhilin. 2020. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.

[33] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, and Guojun Wang. 2019. Haes: A new hybrid approach for movie recommendation with elastic serendipity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1503–1512.

[34] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, Guojun Wang, and Kenli Li. 2020. Directional and explainable serendipity recommendation. In *Proceedings of The Web Conference 2020*. 122–132.

[35] Xiaohan Li, Zheng Liu, Luyi Ma, Kaushiki Nag, Stephen Guo, S Yu Philip, and Kannan Achan. 2022. Mitigating frequency bias in next-basket recommendation via deconfounders. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 616–625.

[36] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022. Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM Web Conference 2022*. 1181–1190.

[37] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[38] Lori McCay-Peet and Elaine G Toms. 2010. The process of serendipity in knowledge work. In *Proceedings of the third symposium on Information interaction in context*. 377–382.

[39] William McKeen. 2006. The Endangered Joy of Serendipity. *St. Petersburg Times* (2006).

[40] Robert K Merton and Elinor Barber. 2011. The travels and adventures of serendipity. In *The Travels and Adventures of Serendipity*. Princeton University Press.

[41] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. 2012. Talmud: transfer learning for multiple domains. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 425–434.

[42] Xi Niu. 2018. An adaptive recommender system for computational serendipity. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 215–218.

[43] Xi Niu and Fakhri Abbas. 2017. A framework for computational serendipity. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. 360–363.

[44] Xi Niu and Fakhri Abbas. 2019. Computational surprise, perceptual surprise, and personal background in text understanding. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. 343–347.

[45] Xi Niu, Fakhri Abbas, Mary Lou Maher, and Kazjon Grace. 2018. Surprise me if you can: Serendipity in health information. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[46] Xi Niu, Chuqin Li, and Xing Yu. 2017. Predictive analytics of E-commerce search behavior for conversion. (2017).

[47] Xi Niu, Wlodek Zadrozny, Kazjon Grace, and Weimao Ke. 2018. Computational surprise in information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1427–1429.

[48] Gaurav Pandey, Denis Kotkov, and Alexander Semenov. 2018. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1771–1774.

[49] Theodore G Remer. 1965. *Serendipity and the three princes: From the Peregrinaggio of 1557*. University of Oklahoma Press.

[50] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[51] Victoria L Rubin, Jacquelyn Burkell, and Anabel Quan-Haase. 2011. Facets of Serendipity in Everyday Chance Encounters: a Grounded Theory Approach to Blog Analysis. *Information Research* 16, 3 (2011).

[52] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic. 2012. Adaptive diversification of recommendation results via latent factor portfolio. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 175–184.

[53] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[54] Jennifer Thom-Santelli. 2007. Mobile Social Software: Facilitating Serendipity or Encouraging Homogeneity? *IEEE Pervasive Computing* 6, 3 (2007), 46.

[55] Saúl Vargas and Pablo Castells. 2014. Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender systems*. 145–152.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[57] Sowmini Devi Veeramachaneni, Arun K Pujari, Vineet Padmanabhan, and Vikas Kumar. 2019. A maximum margin matrix factorization based transfer learning approach for cross-domain recommendation. *Applied Soft Computing* 85 (2019), 105751.

[58] Yuanbo Xu, Yongjian Yang, En Wang, Jiayu Han, Fuzhen Zhuang, Zhiwen Yu, and Hui Xiong. 2020. Neural serendipity recommendation: Exploring the balance between accuracy and novelty with sparse explicit feedback. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 4 (2020), 1–25.

[59] Ting Yu, Junpeng Guo, Wenhua Li, and Meng Lu. 2021. A mixed heterogeneous factorization model for non-overlapping cross-domain recommendation. *Decision Support Systems* 151 (2021), 113625.

[60] Yizhou Zang and Xiaohua Hu. 2017. LKT-FM: A novel rating pattern transfer model for improving non-overlapping cross-domain collaborative filtering. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*. Springer, 641–656.

[61] Mingwei Zhang, Yang Yang, Rizwan Abbas, Ke Deng, Jianxin Li, and Bin Zhang. 2021. SNPR: A Serendipity-Oriented Next POI Recommendation Model. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2568–2577.

[62] Qian Zhang, Dianshuang Wu, Jie Lu, Feng Liu, and Guangquan Zhang. 2017. A cross-domain recommender system with consistent information transfer. *Decision Support Systems* 104 (2017), 49–63.

[63] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. A graphical and attentional framework for dual-target cross-domain recommendation.. In *IJCAI*. 3001–3008.

Received 17 August 2023; revised 17 May 2024; accepted 1 August 2024