OmniMatch: Overcoming the Cold-Start Problem in Cross-Domain Recommendations using Auxiliary Reviews

Yingjun Dai
yingjundai@cmail.carleton.ca
Carleton University, Canada

Ahmed El-Roby
ahmed.elroby@carleton.ca
Carleton University, Canada

Elmira Adeeb
elmiraadeeb@cmail.carleton.ca
Carleton University, Canada

Vivek Thaker
vivekthaker@cmail.carleton.ca
Carleton University, Canada

ABSTRACT

Data sparsity and the issue of cold-start users pose significant challenges in recommender systems. Review-based methods have been developed to mitigate data sparsity by incorporating information from user reviews. The cross-domain cold-start recommendation aims to leverage information from different domains (e.g., Books and Movies) whose overlapping users' data drastically improve the quality of recommendations provided to cold-start users in the target domain. In this paper, we present a novel, review-based, cross-domain recommendation framework, named OmniMatch. In contrast to traditional methods that employ a mapping function to transform the cold-start user's source domain features into target domain features, our approach focuses on generating auxiliary reviews for cold-start users in the target domain for mining and transferring domain-invariant information. We incorporate domain adversarial training and supervised contrastive learning to ensure that the learned features from the users' source and target feature extractors are domain-invariant. We conduct extensive benchmarking against other top cross-domain recommender systems on the widely-used Amazon Review dataset and Douban dataset. Our results demonstrate that OmniMatch has superior cross-domain performance for cold-start users, compared to state-of-the-art recommendation methods.

1 INTRODUCTION

The rise of e-commerce and online shopping has made it easier than ever before for consumers to write evaluative reviews about products that they purchase. These reviews commonly take the form of unstructured plain-text that encompasses a variety of perspectives and details relating to the users' experience with the products. The abundance of products in large e-commerce markets coupled with a steady influx of new items makes it difficult for users to sample all available offerings. This challenge promotes the emergence of "recommender systems". These systems utilize users' historical interactions, ratings, and reviews to generate new suggestions for items that users might not have previously seen.

Traditional recommender systems predominantly use Collaborative Filtering (CF) [18, 19, 23]. CF works by learning user preferences from users' aggregate interactions, such as their ratings of common items that they have bought. CF techniques are effective in cases where a sufficient quantity of ratings data is available. However, this may not be practical in real-world scenarios, as the number of available products may substantially exceed the number of users for emerging businesses. This results in two problems: (1) the data-sparsity problem and (2) the cold-start problem. The data-sparsity problem occurs when there is an insufficient quantity of user-item interactions (e.g., leaving a review for an item). Consequently, this affects the ability of recommender systems to learn users' preferences. The cold-start problem arises when there is a continual influx of new users, none of whom have any prior interactions with any of the products (e.g., users who never bought a product or written a review for a product). This affects the ability of traditional recommender systems to provide high-quality recommendations for these users [14].

Review-based approaches form a prominent strategy for addressing the aforementioned problems in recommender systems. The DeepCoNN [28] model introduces the idea of using a convolutional neural network to learn a user's behavior and an item's characteristics at the same time, based on the content of the user's review of said item. In combination with available ratings data, DeepCoNN can then perform neural collaborative filtering [12] utilizing the extracted features of both users and items. Applying the learned convolutional neural network to users and items with only a few interactions addresses the data sparsity problem, as well as a limited improvement to the cold-start problem.

Following this line of research, review-based methods [3, 25, 28] achieved satisfactory results in predicting ratings by extracting latent features from reviews written by a user and from reviews of a particular product, assuming that the data comes from the same domain (e.g., using reviews of books to help generate relevant recommendations of books to read). However, such a scenario is not always realistic. In scenarios where a user is new to a domain, they may not have any reviews for items within that domain. As a result, these methods fail to extract the latent features of the "cold-start" user. In heterogeneous settings – wherein data comes from more than one domain – review-based methods only perform well when the user has a sufficient quantity of reviews in the target domain. Given that an increasing number of users are interacting with more and more different domains, the potential for leveraging users' data that is common across different domains to alleviate the cold-start problem within a single domain has gained prominence. This has led Cross-Domain Recommendation (CDR) to become an area of increasing interest in recent years.

The core task of cross-domain recommendation is user preference mapping between the two relevant domains. Methods proposed in prior work [7, 17] involve encoding user and item representations separately for the source and target domain. And then a cross-domain mapping is learned from the overlapping users between the two domains, allowing for the transfer of knowledge between domains, and thus addressing the cold-start users problem. In data-scarce situations, these methods struggle to acquire robust representations of users and items for both the source and target domains. Consequently, an error propagation happens during the process of learning the mapping function that bridges the source domain representations to their corresponding representations in the target domain.

In this paper, we introduce OmniMatch, a review-based recommender system that addresses both the cold-start and data-sparsity problems in cross-domain recommendation scenarios. Unlike traditional methods that focus on learning a mapping function from the source to the target domain, our approach takes a novel path. By generating auxiliary reviews for cold-start users in the target domain, we can mine and transfer domain-invariant information more effectively. This approach offers a fresh and more effective perspective for addressing the cold-start problem in cross-domain recommendation systems. OmniMatch achieves state-of-art performance by making two assumptions, as illustrated in Figure 1: 1) Users share similar preferences across domains. For example, a user who loves sci-fi books will tend to love sci-fi movies as well. This assumption enables us to expect that we could extract two sets of features for each user in the source domain and target domain, such that the two sets of features lie close in the latent space - which we call the domain-invariant features. 2) Users share a certain degree of similar preferences if they give the same rating to the same item, which is a common assumption in collaborative filtering techniques in the literature [24, 28]. OmniMatch generates auxiliary review documents for cold-start users in the target domain (based on users who rate the same item with the same rating in the source domain) and combines them with the users' reviews in the source domain to extract the users' domain-invariant information. OmniMatch employs (i) supervised contrastive loss to ensure that each user's source domain features and target domain features are closer in the latent space, and (ii) domain adversarial techniques to learn the domain-invariant representations of users. Finally, a prediction is made for any user-item pair by concatenating the user's target features and the item's features through a multi-layer perceptron.

We summarize our contributions as follows:

- We introduce a novel approach to solve the cold-start problem in a cross-domain recommendation setting by generating auxiliary reviews for cold-start users in the target domain based on their like-minded users and extracting their preferences from those reviews to tackle the data sparsity problem.
- We employ both supervised contrastive learning and domain adversarial learning to align users' source and target domain distributions and make rating predictions for a user-item pair in an end-to-end fashion.
- We extensively evaluate OmniMatch by comparing it to state-of-the-art approaches on the Amazon Review dataset [10] and the Douban Dataset [29]. Our results confirm that our approach is superior across all the evaluated domains for cold-start users.

The rest of the paper is organized as follows: Section 2 offers a comprehensive definition of the problem. Section 3 provides an overview of OmniMatch. Subsequently, Section 4 discusses the modules employed in OmniMatch. This is followed by Section 5 which presents the experimental evaluation of OmniMatch. Section 7 delves into a scholarly review of the work previously carried out in this domain, leading us towards the conclusion in Section 8.

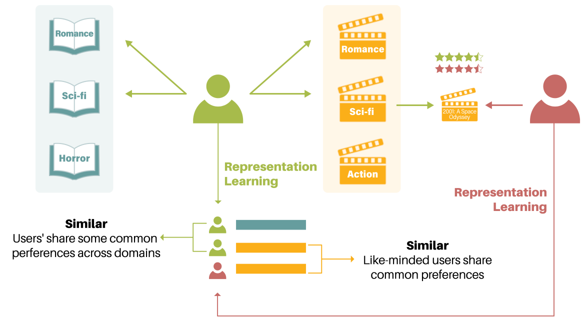


Figure 1: The two assumptions of OmniMatch. 1) Each user has some shared preferences across domains 2) Like-minded users have similar preferences.

Table 1: Notations used in the paper

Notations	
$\mathcal{D}^s$	the source domain data
$\mathcal{D}^t$	the target domain data
$\mathcal{U}^o$	the set of overlapping users
$\mathcal{U}^{cs}$	the set of cold-start users
$\mathcal{I}^s$	the set of items in the source domain
$\mathcal{I}^t$	the set of items in the target domain
D^u	the tokens document for user u
$\mathbf{d}_i$	the n-dimensional distributed vector of the i-th token
$\mathbf{D}^u_{1:k}$	the word embedding matrix of tokens 1 to k for user u
r_{j_source}	the j-th user's source domain features
r_{j_target}	the j-th user's target domain features
r_{j_item}	the j-th item's features

2 PROBLEM DEFINITION

We denote the respective source and target domain data as $\mathcal{D}^s$ and $\mathcal{D}^t$, each containing a set of their respective users ($\mathcal{U}^s$ and $\mathcal{U}^t$), items ($\mathcal{I}^s$ and $\mathcal{I}^t$), and a set $\{u, i, txt, r\}$ representing a user's ($u \in \mathcal{U}$) review text (txt) and rating (r) for a specific item ($i \in \mathcal{I}$). We use the superscripts s and t to represent the corresponding domain. An "overlapping user" is a user who has a review history in both the source and target domains and we denote the set of all "overlapping users" as $\mathcal{U}^o = \mathcal{U}^s \cap \mathcal{U}^t$. Recall that in a single-domain setting, a cold-start user is a user who has no prior reviews or ratings with any of the products in that sole domain. In a multi-domain setting, a cold-start user is a user who has no prior reviews or ratings for items only in the *target* domain. In other words, cold-start users only have review history in the source domain $\mathcal{D}^s$. We denote the set of all cold-start users as $\mathcal{U}^{cs} = \{u | u \in \mathcal{U}^s \wedge u \notin \mathcal{U}^t\}$. Table 1 summarizes the notations used in the paper. Formally, the problem is defined as follows:

Given a source domain $\mathcal{D}^s$ and a target domain $\mathcal{D}^t$, the *cross-domain recommendation for cold-start users* aims to predict the rating a user $u \in \mathcal{U}^{cs}$ will give to an item $i \in \mathcal{I}^t$ based on the preferences of $\{u : \forall\, u \in \mathcal{U}^s\}$ for $\{i : \forall\, i \in \mathcal{I}^s\}$; $\{u : \forall\, u \in \mathcal{U}^t\}$ for $\{i : \forall\, i \in \mathcal{I}^t\}$; and $\{u : \forall\, u \in \mathcal{U}^o\}$ for $\{i : \forall\, i \in \mathcal{I}^s\} \cup \{i : \forall\, i \in \mathcal{I}^t\}$.

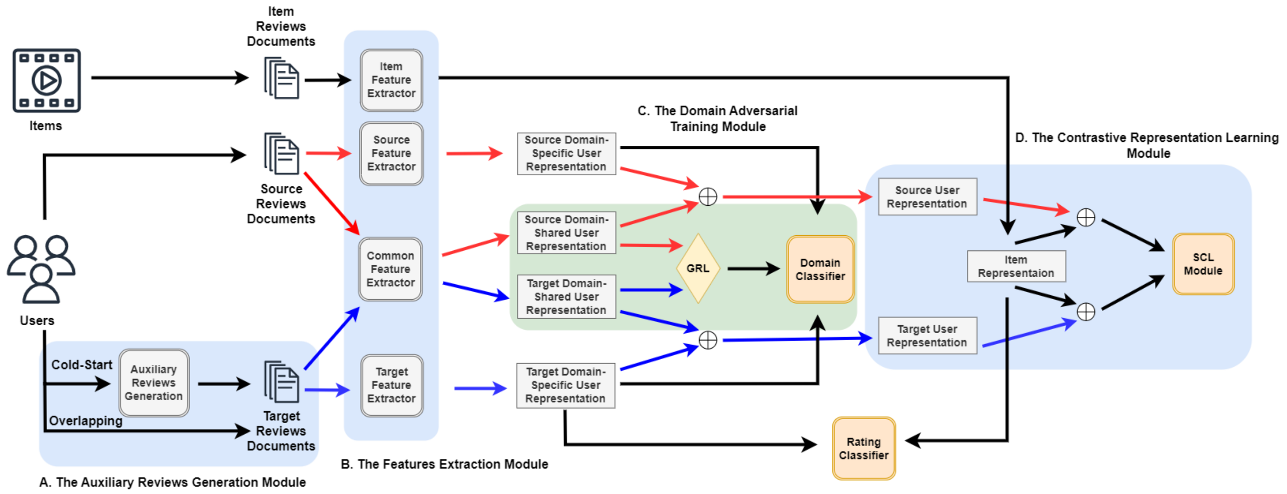


Figure 2: OmniMatch first generates auxiliary reviews in the target domain for cold-start users through *(A) The Auxiliary Reviews Generation Module*. It then generates source and target review documents for all users and an item review document for all items. The review documents are fed into their respective feature extractors in *(B) The Feature Extraction Module*, resulting in users' domain-invariant and domain-specific features and items' features. Users' domain-invariant features from both domains are then passed into *(C) The Domain Adversarial Training Module*, encouraging the features to become domain-invariant. The concatenated representation of users' source/target features and item features are then passed into *(D) The Contrastive Representation Learning Module*. Finally, the rating of a user for an item is predicted in the *Rating Classifier*, which takes the concatenated vector of item features and users' target domain features as input.

3 OVERVIEW OF OMNIMATCH

This section discusses the overview of OmniMatch, whose details will be discussed in Section 4. The architecture of our approach is illustrated in Figure 2. OmniMatch extracts user and item features, aligns users' features from the source to the target domain, and then predicts ratings based on the concatenated representation of the user and item features. It accomplishes these tasks using four components: an Auxiliary Reviews Generation Module, a Features Extraction Module, a Contrastive Representation Learning Module, and a Domain Adversarial Training Module.

Component A in Figure 2 depicts the Auxiliary Reviews Generation Module (Section 4.1), which generates reviews for cold-start users. Since cold-start users do not have reviews in the target domain, we propose an approach to generate auxiliary reviews for such users by leveraging reviews from like-minded users based on their reviews in the source domain. Initially, we identify all overlapping users who have provided identical ratings to an item as the cold-start user in the source domain (we called them the like-minded users). Then, the auxiliary reviews for this specific cold-start user are generated by amalgamating a randomly selected review in the target domain from a randomly selected like-minded user.

Component B in Figure 2 depicts the Features Extraction Module (Section 4.2). The module extracts the user's source domain and target domain features, as well as the item's features by a combination of a convolutional layer and a multi-layer perceptron. OmniMatch employs the shared-private paradigm [2] and uses two feature extractors in both the source and target domains for domain-invariant and domain-specific features. The advantages of the shared-private paradigm is discussed in Section 4.4. The extracted user and item features are then forwarded to the Contrastive Representation Learning Module and the Domain Adversarial Training Module for domain alignment. We essentially expect the same user's features (source and target) to be domain-invariant, as we assume that each user has similar preferences across domains.

Components C and D in Figure 2 depict the Domain Adversarial Training Module (Section 4.4) and the Contrastive Representation Learning Module (Section 4.3), respectively. These two components work in tandem to align the users' source domain features, r_{j_source}, to their corresponding target domain features, r_{j_target}, based on the reviews of overlapping users, $\mathcal{U}^o$, who have reviews in both domains. To ensure that the extracted features are domain-invariant, and to help the process of alignment, the Contrastive Representation Learning Module and the Domain Adversarial Training Module employ a Supervised Contrastive Loss function [13] and a Gradient Reversal Layer [8], respectively. Doing so aligns the distributions of the source and target user features, as will be explained in Sections 4.3 and 4.4.

4 OMNIMATCH MODULES

In this section, we present the proposed framework and the details of each component. Namely, the Auxiliary Reviews Generation Module, the Feature Extraction Module, the Contrastive Representation Learning Module, the Domain Adversarial Training Module, and the overall objective function of the model.

4.1 Auxiliary Reviews Generation Module

Recall that in a cross-domain recommendation setting, cold-start users lack reviews in the target domain, if we were to extract features directly from their source domain reviews, we would essentially be using source domain information to predict target domain ratings, which will lead to a suboptimal performance. To overcome this, we propose to generate auxiliary reviews for these cold-start users. This process effectively constructs a bridge between the user's known preferences in the source domain and their potential interests in the target domain. By generating these reviews, we can more accurately sketch the preferences of cold-start users as they would appear in the target domain.

Based on the assumption that like-minded users share some degrees of common characteristics, we can make use of reviews from those users who have similar ratings for the same items, thereby providing an effective solution to the mentioned problem.

Wu et al. [24] demonstrate the effectiveness of utilizing like-minded users' reviews to improve the quality of recommendations. We make modifications to their method of generating auxiliary review documents for cross-domain scenarios, and we generate documents only for cold-start users, as depicted in Algorithm 1. The auxiliary document of a cold-start user $u \in \mathcal{U}^{cs}$ is generated by first finding the set of all overlapping users who gave the same rating to an item u reviewed in the source domain. With the assumption that like-minded users share common characteristics, we randomly select one review from a randomly selected like-minded user and append it to u's auxiliary document. The auxiliary documents generated are utilized to construct target representations of cold-start users, which are then employed as input in the Contrastive Representation Learning Module, as discussed in Section 4.3. The Target Feature Extractor then captures the features of u from the collection of reviews of overlapping users.

Algorithm 1: Construction of User Auxiliary Documents for the Target Domain

Data: $\mathcal{D}^s$, $\mathcal{D}^t$, $\mathcal{U}^o$, $\mathcal{U}^{cs}$
Result: User Auxiliary Documents for the Target Domain
1 U_AUX_DOC = set( ) ;
2 **foreach** $u \in \mathcal{U}^{cs}$ **do**
3   u_aux_reviews = set( ) ;
  /* get u's purchase records (containing reviews & ratings) in the source domain */
4   records = get_purchase_records_in_source($\mathcal{D}^s, u$) ;
5   **foreach** *record* ∈ records **do**
    /* get u's rating of item */
6   (item, rating) = get_item_rating_info(*record*) ;
    /* get the set of all like-minded users who rate the same item with the same rating */
7   like_minded_s = get_like_minded_s(item, rating) ;
8   like_minded_t = set( ) ;
9   **foreach** *lm_u* ∈ like_minded_s **do**
10   **if** *lm_u* $\in \mathcal{U}^o$ **then**
11   like_minded_t.add(*lm_u*) ;
    /* randomly select one like-minded user */
12   aux_user = random(like_minded_t) ;
    /* get the random user's purchase records in the target domain */
13   aux_records = get_aux_records_target($\mathcal{D}^t$, aux_user) ;
    /* randomly select one record among the purchase records */
14   aux_record = random(aux_records) ;
    /* get the review from the selected record */
15   aux_review = get_review(aux_record) ;
16   u_aux_reviews.add(aux_review) ;
17   U_AUX_DOC.add(u_aux_reviews) ;
18 **return** U_AUX_DOC

In OmniMatch, we propose that similar ratings on a specific item may suggest a certain degree of like-mindedness within that specific context. Furthermore, the auxiliary reviews for a cold-start user in our framework are not based on a limited set of user reviews. These reviews are an aggregate, composed by selecting one review for each purchase record of the cold-start user in the source domain. This methodology is designed to provide a more nuanced and comprehensive view of a user's preferences and characteristics. By incorporating a broader spectrum of reviews, we aim to mitigate the potential bias that could arise from relying on a limited number of reviews. This approach helps in painting a more accurate picture of the cold-start user's preferences, based on the collective insights gleaned from like-minded users. Thus, while similar ratings are a starting point for understanding user preferences, the proposed algorithm also accounts for the diversity and complexity inherent in user behavior and preferences.

Here we provide the time complexity analysis of the algorithm. We first preprocess the entire dataset and generate the following two dictionaries:

(1) a dictionary where the key is the user_id and the value is a list of [item, rating, reviews], representing the ratings and reviews that the user provided for the items they reviewed.
(2) a dictionary where the key is the product_id and the value is a list storing users who rate this item with a specific rating. For example: dict[(item, rating)] = [user1, user2, ...].

The construction of the above dictionaries has a time complexity of O(N·M), where N is the number of users in the dataset and M is the average number of reviews per user.

With the above dictionaries, the Data Retrieval functions (line 4, 6, 7, 13) and the Random Selection (line 12, 14) in the algorithm have a O(1) time.

The dominant factors of the complexity of the algorithms are now the nested loops:

(1) The outer loop iterates over each user in $\mathcal{U}^{cs}$. Let's denote the number of cold-start users as L.
(2) The second loop iterates over the user's reviews to find their like-minded users. As mentioned above, the average number of reviews per user is M.
(3) The third loop iterates over the like-minded users to check if they are in the training data. Let's denote the average number of like-minded users as Q.

Thus, a rough approximation of the overall time complexity for the algorithm would be O(N·M + L·M·Q), where N is the number of users in the dataset, M is the average number of reviews per user, L is the number of cold-start users, and Q is the average number of like-minded users.

4.2 Features Extraction Module

We generate a document R^u, for the user u, which represents the concatenation of all the user's reviews in one domain:

$$R^u = (\text{Review}_1, \text{Review}_2, \ldots, \text{Review}_n) \tag{1}$$

We then tokenize R^u to produce a document D^u, for the user u, as follows:

$$D^u = (d_1, d_2, \ldots, d_k, \ldots, d_l) \tag{2}$$

where $d_1, d_2, \ldots d_l$, are the tokens of all the reviews, l is the length of the document D^u, and $D^u_{1:k}$ represents the first k tokens from D^u.

The user documents $D^u_{1:k}$ are passed into an embedding look-up layer, which maps each token d into an n-dimensional distributed vector $\mathbf{d} \in \mathbb{R}^{n \times 1}$. The word-embedding matrix of $D^u_{1:k}$ is constructed as follows:

$$\mathbf{D}^u_{1:k} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_k]^\top \tag{3}$$

where $\mathrm{d}_1, \mathrm{d}_2, \ldots \mathrm{d}_k$ each represent an n-dimensional vector of the corresponding token.

A convolutional layer is applied on top of $\mathbf{D}^u_{1:k}$ to extract contextual text features. We apply the convolutional operation using filter K_j to obtain each neuron z_j in this layer. The result of the convolutional operation is as follows:

$$z_j = \sigma(\mathbf{D}^u_{1:k} * K_j + b_j) \tag{4}$$

where the symbol $*$ denotes the Convolution operator, σ is the activation function, and b_j is the bias term. The activation function used in the framework is ReLU:

$$\mathrm{ReLU}(x) = \max\{0, x\} \tag{5}$$

A max-pooling layer is applied over the contextual text features to reserve the most valuable features and reduce the size of the vectors:

$$o_j = \max\{z_1, z_2, \cdots, z_t\} \tag{6}$$

$$\mathbf{o}_u = [o_1, o_2, \cdots, o_g] \tag{7}$$

where o_j is the maximum value among $z_1, ..., z_t$ and $\mathbf{o}_u$ is the resulting feature map after the max-pooling operation.

The output of the max-pooling layer is then fed into two fully-connected layers to obtain the domain-invariant and domain-specific representation of the user, respectively. It is important to note that the weights of the convolutional layers, and the weights of the domain-specific fully-connected layer are individual to each domain, but the weights of the domain-invariant fully-connected layer—which are used to extract a user's common features between the source and target domain—are shared, as the goal is to obtain domain-invariant features regardless of which domain the review comes from:

$$r_{j\_\mathrm{invariant}} = \sigma(\mathbf{W}_{u_1}\mathbf{o}_u + \mathbf{b}_{u_1}) \tag{8}$$

$$r_{j\_\mathrm{specific}} = \sigma(\mathbf{W}_{u_2}\mathbf{o}_u + \mathbf{b}_{u_2}) \tag{9}$$

where $r_{j\_\mathrm{invariant}}$ and r_{j_specific} are the final mathematical representation of the domain-invariant and domain-specific features of the input, $\mathbf{W}_{u_1}$ and $\mathbf{W}_{u_2}$ are the matrix representing the weights of the corresponding multi-layer perceptron, $\mathbf{b}_{u_1}$ and $\mathbf{b}_{u_2}$ are the bias terms, and σ is the activation function ($\mathrm{ReLU}(x)$).

The user's representation in one domain is the concatenation of the domain-invariant and domain-specific features of their reviews in that domain:

$$r_j = r_{j\_\mathrm{invariant}} \oplus r_{j\_\mathrm{specific}} \tag{10}$$

The same architecture is applied to obtain a user u's source domain representation, u's target domain representation, which are denoted as r_{j_source} and r_{j_target}, respectively. For an item's representation, we use only the shared feature, and the representation is denoted as r_{j_item}. We use $R_{\mathrm{source\_invariant}}$, $R_{\mathrm{target\_invariant}}$ and $R_{\mathrm{source\_specific}}$, $R_{\mathrm{target\_specific}}$ to denote the set of all users' domain-invariant and domain-specific representations in the source and target domains, respectively.

4.3 Contrastive Representation Learning Module

In recent years, contrastive learning has become the pinnacle of success in representation learning. It centers around learning an embedding space wherein positive pairs are brought closer together and negative pairs are kept at a distance in the latent space. In our framework, we utilize the supervised contrastive loss function. The set of inputs that are used for contrastive learning are referred to as user-item pairs. These user-item pairs are generated by concatenating the source and target representations of the user with the item's representation. As suggested by Chen et al. [4], employing a multi-layer perceptron (MLP) – which we denote as $\mathrm{Proj}(\cdot)$ – for reducing the dimension of the user-item pairs improves the efficiency of supervised contrastive learning. As a result, we construct $\hat{x}_{j\_source}$ and $\hat{x}_{j\_target}$, which represent the reduced vector representation of the user-item pairs as follows:

$$\begin{aligned} \hat{x}_{j\_source} &= \mathrm{Proj}(r_{j\_\mathrm{source}} \oplus r_{k\_\mathrm{item}}) \\ \hat{x}_{j\_target} &= \mathrm{Proj}(r_{j\_\mathrm{target}} \oplus r_{k\_\mathrm{item}}) \end{aligned} \tag{11}$$

where r_{j_source} is the representation of the j^{th} user of the source domain, r_{k_item} is the representation of the k^{th} item, $\hat{x}_{j\_source}$ is the projected vector representation of the concatenation of r_{j_source} and r_{k_item}, $\hat{x}_{j\_target}$ is the projected vector representation of the concatenation of r_{j_target} and r_{k_item}, and $\oplus$ is the vector concatenation operator.

Equipped with the technique above, the supervised contrastive loss function can then be described as follows: let I denote the set of all user-item pairs in the training batch for the model:

$$I = (\hat{x}_1, \hat{x}_2, ...) \tag{12}$$

and let $i \in I$ be the index of an arbitrary sample. The supervised contrastive loss function can now be written in full as thus:

$$\mathcal{L}_{SCL} = \sum_{i \in I} \left(\frac{-1}{|P(i)|} \cdot \sum_{p \in P(i)} \log \left(\frac{\exp(\hat{x}_i \cdot \hat{x}_p / \tau)}{\sum\limits_{a \in A(i)} \exp(\hat{x}_i \cdot \hat{x}_a / \tau)} \right) \right) \tag{13}$$

where $P(i)$ is the set of indices of samples that form positive pairs (user-item pairs with the same label/rating) with the sample i, $A(i) = I - \{\hat{x}_i\}$, the $\cdot$ symbol is the dot product, and the τ symbol is a scalar hyperparameter called temperature.

The importance of the Contrastive Representation Learning Module is highlighted by two key aspects:

(1) the use of supervised contrastive loss encourages the source and target domain representations to be closer together in the latent space, thereby necessitating the model to learn domain-invariant features for the user from the source and target domains.
(2) it assists in learning the features of a particular rating group by drawing the concatenated user-item pairs with the same rating closer together.

Figure 3 provides a visual depiction of what the Contrastive Representation Learning Module achieves. The top two boxes illustrate the first aspect explained above, whilst the bottom two boxes illustrate the second aspect explained above.

4.4 Domain Adversarial Training Module

The main purpose of domain adversarial learning techniques is to regularize the training of the source and target representations (i.e., reduce the gap between the source and target distributions), thereby generating domain-invariant features. Recall that we have a domain classifier to classify which domain the user representation comes from and a rating classifier to predict the rating for the given user-item pair. The Domain Adversarial Training Module aims to minimize the rating classification loss while maximizing the domain classification loss for the domain-invariant features.

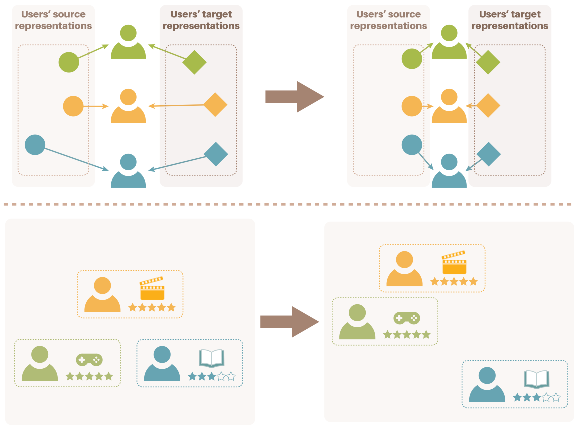


Figure 3: The supervised contrastive loss facilitates the alignment of user representation distributions in source and target domains, not only leading to increased proximity of each user's source and target representations, but also encouraging the convergence of user-item pairs by pulling pairs with the same ratings closer.

The domain classification loss is computed as follows:

$$p(y_d|u_i) = \text{MLP}(u_i) \quad (14)$$

$$\mathcal{L}_{Domain\_Invariant}(\theta_f, \theta_d) = \sum_{i=1} -\log p(y_d|u_i; \theta_f, \theta_d) \quad (15)$$

$$p(y_d|u_j) = \text{MLP}(u_j) \quad (16)$$

$$\mathcal{L}_{Domain\_Specific}(\theta_f, \theta_d) = \sum_{j=1} -\log p(y_d|u_j; \theta_f, \theta_d) \quad (17)$$

In Equation 14 and 16, $u_i \in (R_{\text{source\_invariant}} \cup R_{\text{target\_invariant}})$ is the user's domain-invarient representation, $u_j \in (R_{\text{source\_specific}} \cup R_{\text{target\_specific}})$ is the user's domain-specific representation and y_d is the domain label for u_i. In Equation 15 and 17, θ_f and θ_d are the parameters of the Feature Extractor Module and the Domain Classifier, respectively.

In this study, we incorporate the shared-private paradigm as proposed by Bousmalis et al. [2] within our framework. This paradigm enables the distinction and integration of domain-specific knowledge, which identifies the unique attributes of a given domain, with domain-invariant features, thereby augmenting their discriminative capacity. There might be an issue when only one singular feature extractor is employed. This issue will potentially lead to the model's predilection towards learning solely the user's rating intentions. Such a scenario may result in the representation learning being confined to predicting possible ratings a user might assign to a product, illustrating a bias where some users are predisposed to award higher ratings (e.g., five stars), while others may lean towards lower scores. This limitation similarly extends to item representation, where the model might predominantly forecast the likelihood of an item receiving a specific star rating, rather than encapsulating the item's inherent characteristics. Since those representations are not domain-specific, the domain classifier cannot accurately determine their originating domain, but they do not align with the desired feature set. The implementation of the shared-private paradigm offers a partial mitigation for this issue. By utilizing a concatenated representation that combines both the user's domain-specific and domain-invariant features as input for the rating classifier, it ensures that the domain-specific features convey information pertinent to the user's specific domain, rather than merely reflecting their rating tendencies.

The rating classification loss is computed as follows:

$$p(y_j|r_{j\_\text{target}}, r_{k\_\text{item}}) = \text{MLP}(r_{j\_\text{target}} \oplus r_{k\_\text{item}}) \quad (18)$$

$$\mathcal{L}_{Rating}(\theta_f, \theta_r) = \sum_{j=1} -\log p(y_j|r_{j\_\text{target}}, r_{k\_\text{item}}; \theta_f, \theta_r) \quad (19)$$

In Equation 18, y_j is the rating label of the given user-item pair. r_{j_target}, r_{k_item}, and $\oplus$ is the vector concatenation operator. In Equation 19, θ_r is the parameters of the rating classifier.

At learning time, the model learns the parameters θ_f, θ_r, and θ_d, of the domain-invariant feature extractor, the rating classifier, and the domain classifier, respectively. The parameters of θ_d are optimized to maximize the domain classification loss, while those of θ_r and θ_d are optimized to minimize their respective classification losses.

This module is crucial because our framework employs a common feature extractor designed to derive domain-invariant user features from both source and target domain reviews. Without the domain adversarial training, the effectiveness of this common feature extractor would be significantly compromised. As for our choice of algorithm, we opted for the *Gradient Reversal Layer* (GRL), which multiplies the gradients of the domain classifier by a negative value during back-propagation to achieve the min-max optimization, for domain adversarial training. GRL is widely favored in domain adaptation due to its straightforward implementation, seamless integration, and proven efficiency in fostering domain invariance. It is important to note that our framework is not inherently dependent on this technique; it is versatile enough to accommodate other domain adversarial training methods (GAN [21], ADDA [9]) as well.

4.5 Training Objective Function

The overall training objective function of OmniMatch is a weighted sum of the rating classification loss, the domain classification loss, and the supervised contrastive loss are shown in Equation 21:

$$\mathcal{L}_{Domain} = \mathcal{L}_{Domain\_Specific} + \mathcal{L}_{Domain\_Invariant} \quad (20)$$

$$\mathcal{L}_{Total} = \mathcal{L}_{Rating} + \alpha\mathcal{L}_{SCL} + \beta\mathcal{L}_{Domain} \quad (21)$$

In the total loss $\mathcal{L}_{Total}$, the supervised contrastive loss $\mathcal{L}_{SCL}$ measures the progress of the feature alignment between domain representations of users and user-item pairs with the same rating. The rating classification loss $\mathcal{L}_{Rating}$ represents how well the model predicts a user-item representation of its actual rating. Likewise, the domain classification loss $\mathcal{L}_{Domain}$ regulates the gap between the source and target domain. Minimizing the weighted sum of the above losses enables OmniMatch to perform well in cross-domain recommendations.

5 EXPERIMENTS & DISCUSSION

In this section, we discuss the experimental setup, present our experimental results on six cross-domain recommendation scenarios, perform an ablation study on each component of our framework, explore the effects of varying the values of the hyper-parameters of OmniMatch, a complexity analysis of different modules, and finally we present a case study for the Auxiliary Reviews Generation Module.

5.1 Datasets

We empirically assess the performance of our model against state-of-the-art approaches using the Amazon Review dataset [10] and the Douban Dataset [29]. Specifically, among the largest categories, we select three relevant domains: Books, Movies, and Music, as our evaluation settings.

We conducted experiments to evaluate the rating prediction performance of OmniMatch in six distinct scenarios, namely Books → Movies, Movies → Books, Movies → Music, Music → Movies, Books → Music, and Music → Books.

5.2 Experimental Setup

For each domain, we removed the records that do not include reviews and for each cross-domain scenario, we only keep users who have records in both domains. Among all the overlapping users, 80% of them were randomly selected as training users. The remaining 20% of overlapping users were treated as the cold-start users. These cold-start users' reviews in the target domain were not seen by the model, but were used for validation and testing purposes. Of these cold-start users, half of them were allocated as validation users, and the other half were used as test users.

Users might focus on different qualities of the same item, despite providing similar ratings. Thus, we use the "review summary" field of each record instead of the full review, and we convert the text to lowercase and eliminate all punctuation. Based on our experimental results, it is more effective than analyzing complete review texts. It enables the model to process a larger volume of data within input constraints, thus offering a broader and more accurate portrayal of user characteristics.

We use RMSE (root mean squared error) and MAE (mean absolute error) as the evaluation metrics:

$$RMSE = \sqrt{\frac{1}{|O_{\text{test}}|} \sum_{(u,i) \in O_{test}} (y_{u,i} - \hat{y}_{u,i})^2} \tag{22}$$

$$MAE = \frac{1}{|O_{\text{test}}|} \sum_{(u,i) \in O_{test}} |y_{u,i} - \hat{y}_{u,i}| \tag{23}$$

where O_{test} is the cold-start test set, $y_{u,i}$ is the user u's gold-standard rating given to item i, and $\hat{y}_{u,i}$ is the user u's predicted rating given to item i.

5.3 Baselines

Our method utilizes the rating information, so we compare our approach against recent methods which also make use of the ratings as a source of information. We compared our methods with 2 single domain recommendation systems(NGCF and LIGHTGCN) and 4 cross-domain recommendation systems(CMF, EMCDR, PTUPCDR and HeroGraph):

- **CMF** [20] shares the user factors, which represent the user in the latent space, in the source and target domain, and factorize multiple rating matrices simultaneously.
- **NGCF** [22] integrates user-item interactions into the embedding process with a bipartite graph structure.
- **LIGHTGCN** [11] simplifies Graph Convolution Network for collaborative filtering by removing feature transformation and nonlinear activation, and keeps only neighborhood aggregation for nodes' representations.
- **EMCDR** [17] first utilizes matrix factorization to learn latent factors in both domains, which are then used to map user latent factors from the source domain to the target domain through the implementation of a Multi-Layer Perceptron (MLP).
- **PTUPCDR** [31] uses a meta-network to learn personalized bridge function for each user for transferring user preferences.
- **HeroGraph** [6] obtains cross-domain information by a shared graph, which is constructed by collecting users' and items' information from multiple domains.

5.4 Implementation Details

The convolutional kernel size for the user and item feature extractors is set to (3, 4, 5) and the number of kernels is 200. The input to the convolutional layer is the pre-trained, 300-dimensional, fastText [1] word embedding. The output of the projection network used for contrastive learning is a 128-dimensional vector.

ReLU is used as the activation function. The dropout rate is set to 0.4 and is applied after each linear layer. The batch size used for training is 64. The Adadelta optimizer is employed with a learning rate of 0.02, $\rho = 0.95$, and the model is trained for 15 epochs. The temperature of contrastive learning τ is set to 0.07. We conducted 5 random trials for each experiment and reported the average. The model is trained on one Nvidia A100 40GB GPU.

5.5 Experimental Results

In this study, we present a comprehensive evaluation of OmniMatch[1] against state-of-the-art methods across six cross-domain scenarios, specifically focusing on the domains of Books, Movies, and Music. The results are illustrated in Table 2 and Table 3. The results obtained indicate that our approach outperforms all previous methods across all scenarios, with the best and second-best results indicated in boldface and underlined, respectively.

Across the six evaluated scenarios in each dataset, our method not only consistently achieved the best results but also showed significant improvements in the rating prediction performance. For instance, in the scenario for Books to Movies within the Douban dataset, our model achieved a remarkable 25.9% improvement in RMSE and 32.6% in MAE compared to the second-best performing methods. This trend of notable improvement was similarly observed in the Amazon dataset, with improvements such as a 14.6% increase in RMSE and 13.0% in MAE in the Movies to Music scenario.

Overall, these results highlight the robustness and adaptability of our model across different datasets and scenarios. Our approach significantly outperformed the nearest competitors, achieving an average improvement of 23.1% in RMSE and 26.6% in MAE across the Douban dataset, and an average improvement of 7.4% in RMSE and 9.1% in MAE across the Amazon dataset. This dual-dataset analysis not only affirms the generalizability of our method but also demonstrates its capability to leverage domain-invariant features effectively, thereby enhancing recommendation performance in diverse settings.

5.6 Experiments with different proportions of overlapping users

In this section, we investigate the impact of training with varying proportions (100%, 80%, 50%, and 20%) of training users. We compared OmniMatch agasint two baselines (EMCDR and PTUPCDR). The outcomes are presented in Table 4. It is noteworthy that the RMSE results exhibit minimal variation as the

[1]Our implementation of the framework is available at the following link: https://github.com/pjxxxd/EDBT25-OmniMatch

Table 2: Performance comparisons for different cross-domain scenarios of the Amazon Dataset in terms of RMSE and MAE. The best and second-best results are in boldface and underlined, respectively.

Domains \ Methods		NGCF	LIGHTGCN	CMF	EMCDR	PTUPCDR	HeroGraph	Ours	△%
Books -> Movies	RMSE	1.150	1.124	1.558	1.166	1.049	1.118	**1.031**	1.7%
	MAE	0.893	0.870	1.188	0.903	0.906	0.861	**0.758**	12.0%
Movies -> Books	RMSE	1.180	1.174	1.747	1.222	1.215	1.133	**1.035**	8.6%
	MAE	0.958	0.901	1.319	0.953	0.946	0.867	**0.787**	9.2%
Books -> Music	RMSE	1.104	1.102	2.510	1.167	1.175	1.026	**0.962**	6.2%
	MAE	0.906	0.828	1.967	0.920	0.894	0.815	**0.725**	11.0%
Music -> Books	RMSE	1.180	1.174	1.641	1.337	1.300	1.121	**1.038**	7.4%
	MAE	0.958	0.901	1.266	1.054	1.015	0.886	**0.821**	7.3%
Movies -> Music	RMSE	1.104	1.102	1.972	1.095	1.118	1.101	**0.940**	14.6%
	MAE	0.906	0.828	1.468	0.829	0.843	0.798	**0.694**	13.0%
Music -> Movies	RMSE	1.150	1.124	1.972	1.109	1.118	1.088	**1.026**	5.7%
	MAE	0.893	0.870	1.068	0.935	0.908	0.802	**0.785**	2.1%

Table 3: Performance comparisons for different cross-domain scenarios of the Douban Dataset in terms of RMSE and MAE. The best and second-best results are in boldface and underlined, respectively.

Domains \ Methods		NGCF	LIGHTGCN	CMF	EMCDR	PTUPCDR	HeroGraph	Ours	△%
Books -> Movies	RMSE	1.312	1.296	1.598	1.416	1.142	1.131	**0.838**	25.9%
	MAE	1.091	1.055	1.131	1.008	0.951	0.894	**0.603**	32.6%
Movies -> Books	RMSE	1.412	1.212	2.602	2.732	2.820	1.201	**0.919**	23.5%
	MAE	1.121	1.055	1.900	2.173	2.732	0.987	**0.727**	26.3%
Books -> Music	RMSE	1.284	1.237	2.917	2.908	3.008	1.212	**0.904**	25.4%
	MAE	1.101	1.002	2.273	2.351	2.329	0.979	**0.801**	18.2%
Music -> Books	RMSE	1.412	1.212	3.034	2.826	3.036	1.268	**0.914**	25.4%
	MAE	1.121	1.055	2.341	2.232	2.284	1.049	**0.780**	25.6%
Movies -> Music	RMSE	1.284	1.237	2.863	2.802	2.851	1.226	**0.958**	21.9%
	MAE	1.101	1.002	2.138	2.210	2.158	0.988	**0.657**	33.5%
Music -> Movies	RMSE	1.312	1.296	1.869	1.414	1.377	1.158	**0.873**	24.6%
	MAE	1.091	1.055	1.289	0.989	0.941	0.895	**0.687**	23.2%

Table 4: Experiments results with different proportions of overlapping users across domains

Methods	Metrics	Books → Movies				Movies → Music				Books → Music			
		100%	80%	50%	20%	100%	80%	50%	20%	100%	80%	50%	20%
EMCDR	RMSE	1.166	1.184	1.197	1.221	1.095	1.128	1.154	1.183	1.167	1.189	1.192	1.199
	MAE	0.903	0.906	0.921	0.944	0.829	0.859	0.871	0.885	0.920	0.945	0.947	0.954
PTUPCDR	RMSE	1.049	1.066	1.143	1.225	1.118	1.150	1.173	1.209	1.175	1.183	1.201	1.254
	MAE	0.906	0.910	0.924	0.946	0.843	0.874	0.884	0.906	0.894	0.926	0.941	0.972
Ours	RMSE	1.031	1.036	1.041	1.071	0.940	0.953	0.973	1.006	0.962	0.976	0.991	1.014
	MAE	0.758	0.791	0.787	0.812	0.694	0.706	0.733	0.756	0.725	0.822	0.864	0.876

percentage of training users decreases. Remarkably, even when trained with merely 20% of overlapping users, OmniMatch consistently delivers the best RMSE outcomes.

Traditional methods typically follow a three-step optimization process: learning user representations in the source domain, then in the target domain, and finally learning a mapping function between these domains. This conventional approach is highly sensitive to the volume of training data available. In scenarios where this ratio is small, the mapping function suffers due to insufficient training data, leading to suboptimal results.

Table 5: Ablation study on each component of OmniMatch.

Scenario		Books → Movies	Books → Music	Movies → Music
w/o SCL	RMSE	1.073	1.029	1.013
	MAE	0.909	0.902	0.769
w/o DA	RMSE	1.075	1.025	1.011
	MAE	0.905	0.894	0.764
w/o Aux_Reviews	RMSE	1.173	1.034	1.061
	MAE	0.928	0.896	0.854
OmniMatch	RMSE	**1.068**	**1.021**	**1.006**
	MAE	**0.901**	**0.830**	**0.756**
OmniMatch-ReviewText	RMSE	1.088	1.080	1.031
	MAE	0.848	0.856	0.781
OmniMatch-BERT	RMSE	1.174	1.038	1.077
	MAE	0.917	0.810	0.836

In contrast, our method innovatively incorporates auxiliary reviews from like-minded users and employs Domain Adaptation (DA) and Supervised Contrastive Learning (SCL) techniques. This approach allows for the development of more robust user representations. Learning user representations from reviews is less reliant on the quantity of training data, enabling them to maintain high performance even with a smaller dataset. As a result, our method demonstrates superior performance by effectively overcoming the limitations of traditional approaches in data-constrained environments.

Our innovative approach highlights the effectiveness of extracting user features from reviews and combining with advanced techniques like DA and SCL in scenarios with limited training data, leading to the significant performance improvement observed in our results.

5.7 Ablation Study

The ablation study detailed in Table 5 systematically evaluates the contribution of distinct components within OmniMatch, a comprehensive model designed for cross-domain recommendations. We conducted experiments in a data-scarce scenario by using 20% of overlapping users as training data to best anaylize the effects of the individual and collective impact of various modules embedded within OmniMatch, namely the (i) Contrastive Representation Learning Module, the (ii) Domain Adversarial Training Module, and the (iii) Auxiliary Reviews Generation Module. By selectively disabling these components, we aim to discern their respective roles in enhancing the model's performance across different domain transitions, specifically from Books to Movies, Books to Music, and Movies to Music.

The most significant impact is observed when the Auxiliary Reviews Generation Module is omitted. This module synthesizes auxiliary reviews to enrich the model's understanding of user preferences and item characteristics, acting as a pivotal factor in improving recommendation quality. The considerable increase in both RMSE and MAE across transitions upon removing this module underscores its importance in bridging the information gap between domains, particularly in scenarios with sparse data.

The ablation study conclusively demonstrates the integral role of each component within OmniMatch. The Contrastive Representation Learning and Domain Adversarial Training modules incrementally refine the model's feature representations, making them more effective for cross-domain recommendations. In contrast, the Auxiliary Reviews Generation Module significantly enhances the model's capability to understand and predict user preferences across domains, highlighting its critical role in OmniMatch's architecture. This comprehensive evaluation validates the framework's design choices, where each module is essential for achieving superior performance in cross-domain recommendation scenarios.

We also conducted two extra experiments: the first involved the utilization of the "reviewText" field of the record, while the second employed BERT as the feature extractors. The results show that employing the review summary yielded superior performance compared to utilizing the full review. This improvement is possibly due to the succinct nature of review summaries, which allows for a more comprehensive portrayal of the cold-start user's characteristics through the concatenation of multiple review summaries from like-minded users. Additionally, the results also indicated that utilizing BERT as feature extractors yielded suboptimal performance in comparison to employing CNNs.

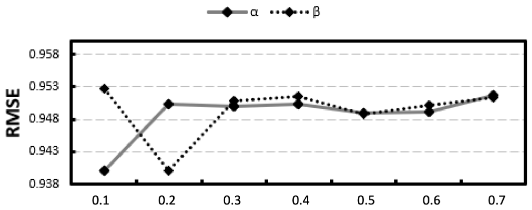
CNNs can be particularly effective when the relevant features for prediction are local and can be captured by patterns or keywords. Since we are using user reviews summaries, the users' preferences may be strongly indicated by keywords or phrases, and CNNs are able to capture these features more efficiently than BERT. When training with BERT as feature extractors, we also observed a consistent decline in the training loss which did not parallel improvements in the validation loss—indicative of overfitting.

5.8 Hyperparameters Analysis

In this section, we explore the impact of the hyperparameters (α and β) on OmniMatch. These two hyperparameters are used to obtain an optimal balance between the three different loss functions. We conduct hyperparameters analysis for Movies → Music and the results are shown in Figure 4. In our experiments, we did not notice any significant change in the results for the other scenarios. When evaluating one hyperparameter, the other one is always set to a fixed value. That is, when evaluating α, β is fixed to 0.1, and when evaluating β, α is fixed to 0.2.

It is worth-noting that we determine the optimal values for hyperparameters α and β (0.1/0.2) through a comprehensive grid search, finding the model's performance to be robust across a range of values. This indicates our method's flexibility and effectiveness, independent of precise hyperparameter tuning.

(a) The RMSE of different values of α (Movies → Music)



(b) The MAE of different values of α (Movies → Music)

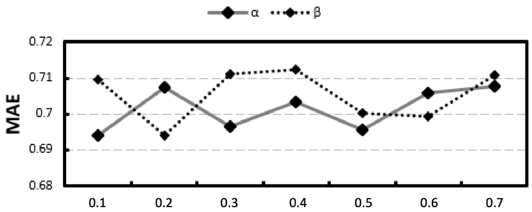


Figure 4: Hyperparameters Analysis

5.9 Performance Analysis of Modules

In this section, we conduct experiments to assess the impact of specific modules on the training time of our model across different domain adaptation scenarios. The components analyzed are (1) the Supervised Contrastive Learning (SCL) Module, and (2) the Domain Adaptation (DA) Module. We compare the full-fledged OmniMatch to the cases where each of these modules is individually removed. The results are summarized in Table 6.

Table 6: Training Time for removing different Modules

Scenario	Full Model	w/o DA	w/o SCL
Books → Music	20 mins	16 mins	17 mins
Movies → Music	24 mins	19 mins	20 mins

5.10 Case Study

In this section, we provide one case study for the auxiliary reviews generation process for the user with the ID:AKOHBSPLTYBYZ, under the scenario Books -> Movies.

User: AKOHBSPLTYBYZ has the following rating histories in the source domain (Books):

(1)
- **Item in source domain**: B00640YZ1U
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Vampire Romance"
- **Like-minded User**: A3U7ELIED4WP4R (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "Fang-tastic, Fun and Freaky"

The cold-start user rated the book(B00640YZ1U) 5.0 stars, and among all overlapping users, the auxiliary review generation module randomly chooses one user A3U7ELIED4WP4R, since they both gave the same book the same rating (5.0 stars). Among the reviews the like-minded users wrote in the target domain, the module picks "Fang-tastic, Fun and Freaky". We expect that this review will reveal some of the preferences of the cold-start user within the target domain. Specifically, the auxiliary review assists the model in understanding that the cold-start user has a penchant for freaky movies featuring 'Fangs'.

(2)
- **Item in source domain**: 0988624524
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Shape shifters"
- **Like-minded User**: A29FFT26RF63YX (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "Are you afraid of the Boogeyman?"

This process is repeated for each book purchased by the cold-start user in the source domain. For instance, the cold-start user wrote a review "Shape shifters" for one book, and the module generates the review "Are you afraid of the Boogeyman?". This generated review, focusing on horror/demon themes, potentially reflects aspects of the cold-start user's preferences. Given that the user assigned a 5-star rating and mentioned 'Shape shifters', it suggests an affinity for narratives involving supernatural entities like 'vampires, werewolves, or demons'.

(3)
- **Item in source domain**: 1455546941
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Adventure"
- **Like-minded User**: A3JF001WP479T7 (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "They did a fantastic job with the picture and sound"

While this particular iteration does not establish a direct connection between the 'Adventure' and the appreciation for 'fantastic job with the picture and sound', it may be inferred that users fond of adventure movies tend to favor films with superior visual and sound quality. As the module processes and generates reviews for an increasing number of cold-start users, it will progressively develop a more comprehensive and nuanced understanding of how preferences transfer across different cold-start users in terms of specific features.

(4)
- **Item in source domain**: 1479257389
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Vampire Romance"
- **Like-minded User**: A3FOL8CN5A1TFR (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "Crouching Tiger, Hidden Dragon"

In this iteration, despite the lack of a direct link between 'Vampire Romance' and 'Crouching Tiger, Hidden Dragon', the module suggests that the cold-start user would enjoy 'Crouching Tiger, Hidden Dragon'. This recommendation aligns more closely with the previously generated 'Adventure' theme. While not every iteration may yield a straightforward transfer of preferences, as the process continues, the module is expected to progressively refine and clarify the depiction of the cold-start user's preferences within the target domain.

(5)
- **Item in source domain**: 1420128612
- **Cold-start User's Rating and Review in the source domain**: 5.0, "vampires"
- **Like-minded User**: A2KGX2X4NYKY9A (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "Oh, I love the Vampire Diaries"

(6)
- **Item in source domain**: 0988624559
- **Cold-start User's Rating and Review in the source domain**: 5.0, "very hot"
- **Like-minded User**: A29FFT26RF63YX (Both Ratings: 5.0)

- **Auxiliary Review chosen from the like-minded user in the target domain**: "Oh so sexy Vampire movie"

(7)
- **Item in source domain**: 0425255808
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Vampires"
- **Like-minded User**: A1L876C8SCT7GV (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "They enjoyed it and watched it often"

(8)
- **Item in source domain**: 0451239814
- **Cold-start User's Rating and Review in the source domain**: 5.0, "Awesome read"
- **Like-minded User**: A1LQ9TMFRRPJU2 (Both Ratings: 5.0)
- **Auxiliary Review chosen from the like-minded user in the target domain**: "great show"

Thus, the final auxiliary reviews document for the cold-start user AKOHBSPLTYBYZ is the concatenation of all the auxiliary reviews generated: **"Fang-tastic, Fun and Freaky <sp> Are you afraid of the Boogeyman? <sp> Oh so sexy Vampire movie.<sp> They did a fantastic job with the picture and sound <sp> Crouching Tiger, Hidden Dragon <sp> Oh, I love the Vampire Diaries <sp> They enjoyed it and watched it often <sp> great show"**

While the ground truth reviews of the user AKOHBSPLTYBYZ has in the target domain are the following:

- '0782009123': 'Vampire romance or soap opera.',
- '0783226934': 'action',
- '0792153189': 'Action',
- '0800137884': 'very good.',
- '157492639X': 'Historical',
- '6302814766': 'Vampire',
- 'B00006CXHU': 'Ice Age',
- 'B000A8AXXG': 'Barbara cartland',
- 'B0059XTU1S': 'Adventure

The concatenated reviews for the target domain is: **"Vampire romance or soap opera. <sp> Action <sp> Action <sp> very good <sp> Historical <sp> Vampire <sp> Ice Age <sp> Barbara Cartland <sp> Adventure"**

The auxiliary reviews document created for the user AKOHBSPLTYBYZ offers a detailed characteristics in the target domain. This document is subsequently fed as input into the target domain feature extractor in the framework to predict ratings.

6 POTENTIAL LIMITATIONS

In this section, we discuss the potential limitations of our method: OmniMatch makes the assumption that users share similar preferences across domains. However, there are domains that share minimal common features. In such scenarios, our proposed approach will suffer.

Our hypothesis is carefully constructed around the notion that while identical ratings across different domains do not necessarily indicate identical interests, they may reflect a shared inclination or like-mindedness within specific contexts. Thus, our method does require "potential connections" between the source and target domain, and it is aiming to capture the connection through the like-mindedness of users.

Importantly, our methodology for integrating auxiliary reviews for cold-start users is designed to capture a broad and nuanced spectrum of user interactions. Unlike approaches that might rely on a constrained dataset of reviews, our model constructs an aggregated review profile for each cold-start user by meticulously selecting one review per purchase record in the source domain. This comprehensive aggregation strategy is pivotal in painting a detailed and multidimensional portrait of user preferences, mitigating the risk of bias that could emerge from an over-reliance on a narrow selection of user reviews.

7 RELATED WORK

In this section, we briefly review three related areas of our work: cross-domain recommendation, review-based recommendation, and contrastive learning.

7.1 Cross-Domain Recommendation

The use of information from a multitude of different domains to address challenges related to data sparsity and cold-start issues in recommender systems has garnered considerable attention among researchers. In the early years, methods such as CMF [20] incorporated interactions from several domains by concatenating multiple rating matrices and learning a global user-embedding matrix that applies to all domains. In recent years, advancements in deep learning techniques have led to the emergence of methods such as EMCDR [17] and TMCDR [30], which focus on learning a mapping function from the source domain features to the target domain features. EMCDR accomplishes this by employing a multi-layer, fully-connected neural network that learns a mapping function from overlapping users. On the other hand, TMCDR introduces the concept of meta-learning to enhance its generalization capabilities. In addition to these approaches, Zhao et al. [26] proposed a modeling technique that bridges users' preferences in the source domain and the target domain (i.e., *preference transfer*) by analyzing review aspects (properties), while learning aspect correlations across domains. HeroGraph [6] is a heterogeneous graph framework using graph convolutional layers and attention mechanisms for obtaining cross-domain information. ALCDR [27] is another graph framework that learns the cross-domain correlations by treating the anchor links between users and domains as learnable parameters. In comparison to the aforementioned methods, OmniMatch differs by directly extracting domain-invariant information across domains via leveraging the content of review texts; it does not use a separate feature transfer module exclusively for the source-to-target domain transfer.

7.2 Review-Based Recommendation

Review-based recommender systems [3, 5, 15, 16, 25, 28] normally factorize review words and pass them into Collaborative Filtering. Zheng et al. [28] proposed the DeepCoNN model, which utilizes two parallel convolutional neural networks as well as word embeddings to capture latent representations of all reviews' words associated with a specific user-item pair. The model concatenates the user and item representations and passes them to a regression layer that employs a Factorization Machine for rating predictions.

According to Wu et al. [25], prior research has relied on extracting features from reviews and ratings in an independent and static manner, thereby failing to capture user preferences. To address this limitation, the authors proposed a novel approach that involves employing two distinct learning components to extract review-based and interaction-based features. These features are then integrated into a Factorization Machine to generate final predictions.Liu et al. [16] propose a hybrid neural recommendation model called HRDR, which captures user and item features from both reviews and ratings. They employ a Multilayer Perceptron (MLP) as the rating-based encoder. The MLP is capable of acquiring higher-order information from rating patterns, as documented in previous literature [5, 15].Furthermore, rather than solely integrating representations obtained from reviews and ratings, they propose a novel attention-based network to select useful reviews for users or items based on the rating-based representation, thereby maximizing the exploitation of the inherent relationship between reviews and ratings.

These approaches are primarily designed in homogeneous settings, and in OmniMatch, we incorporate convolutional neural networks to extract users' and items' features, and apply domain adversarial training techniques to extend the homogeneous settings into heterogeneous settings.

8 CONCLUSION

In this paper, we propose OmniMatch – a novel model for cross-domain recommendation for cold-start users. Particularly, our method focuses on mining the domain-invariant information from a user's source and target domain features based on the assumption that user preferences maintain consistency across diverse domains. Supervised contrastive learning and domain adversarial training are employed to enhance the domain-invariant features extraction. For cold-start users, who have no reviews in the target domain, the method generates auxiliary reviews from their like-minded users. Our results for the cold-start, cross-domain recommendation problem, using the Amazon Review dataset and the Douban dataset, demonstrate that our method outperforms all existing models in providing cross-domain recommendations.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017).

[2] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Advances in Neural Information Processing Systems* 29 (2016).

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *Proceedings of the World Wide Web Conference.*

[4] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. Big Self-Supervised Models Are Strong Semi-Supervised Learners. *Advances in Neural Information Processing Systems* 33 (2020).

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the Workshop on Deep Learning for Recommender Systems.*

[6] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation.. In *Workshop on Online Recommender Systems and User Modeling.*

[7] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[8] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *International Conference on Machine Learning*. PMLR.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems* 27 (2014).

[10] Ruining He and Julian McAuley. [n.d.]. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the International Conference on World Wide Web*. WWW.

[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 International ACM SIGIR conference on Research and Development in Information Retrieval.*

[12] 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.*

[13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *Advances in Neural Information Processing Systems* 33 (2020).

[14] Shah Khusro, Zafar Ali, and Irfan Ullah. 2016. Recommender Systems: Issues, Challenges, and Research Opportunities. In *Information Science and Applications (ICISA) 2016*. Springer.

[15] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[16] Hongtao Liu, Yian Wang, Qiyao Peng, Fangzhao Wu, Lin Gan, Lin Pan, and Pengfei Jiao. 2020. Hybrid Neural Recommendation with Joint Deep Representation Learning of Ratings and Reviews. *Neurocomputing* 374 (2020).

[17] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 17. IJCAI.

[18] Michael J Pazzani. 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review* 13 (1999).

[19] AM Rashid, KL Shyong, and KL George. [n.d.]. A Highly Scalable Hybrid Model & Memory-Based CF Algorithm. *Proceedings of the WEBKDD of ACM* ([n. d.]).

[20] Ajit P Singh and Geoffrey J Gordon. 2008. Relational Learning Via Collective Matrix Factorization. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[21] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[22] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval.*

[23] Jianshu Weng, Chunyan Miao, Angela Goh, Zhiqi Shen, and Robert Gay. 2006. Trust-Based Agent Community for Collaborative Recommendation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems.*

[24] Libing Wu, Cong Quan, Chenliang Li, and Donghong Ji. 2018. Parl: Let Strangers Speak Out What You Like. In *Proceedings of the ACM International Conference on Information and Knowledge Management.*

[25] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. [n.d.]. A Context-Aware User-Item Representation Learning for Item Recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 ([n. d.]).

[26] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-Domain Recommendation for Cold-Start Users Via Aspect Transfer Network. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[27] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. 2023. Beyond the Overlapping Users: Cross-Domain Recommendation via Adaptive Anchor Link Learning. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[28] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining.*

[29] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management.*

[30] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[31] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining.*