

# A cross-domain recommender system using deep coupled autoencoders

Alexandros Gkillas , Dimitrios Kosmopoulos, *Member, IEEE*

**Abstract**—Long-standing data sparsity and cold-start constitute thorny and perplexing problems for the recommendation systems. Cross-domain recommendation as a domain adaptation framework has been utilized to efficiently address these challenging issues, by exploiting information from multiple domains. In this study, an item-level relevance cross-domain recommendation task is explored, where two related domains, that is, the source and the target domain contain common items without sharing sensitive information regarding the users’ behavior, and thus avoiding the leak of user privacy. In light of this scenario, two novel coupled autoencoder-based deep learning methods are proposed for cross-domain recommendation. The first method aims to simultaneously learn a pair of autoencoders in order to reveal the intrinsic representations of the items in the source and target domains, along with a coupled mapping function to model the non-linear relationships between these representations, thus transferring beneficial information from the source to the target domain. The second method is derived based on a new joint regularized optimization problem, which employs two autoencoders to generate in a deep and non-linear manner the user and item-latent factors, while at the same time a data-driven function is learnt to map the item-latent factors across domains. Extensive numerical experiments on two publicly available benchmark datasets are conducted illustrating the superior performance of our proposed methods compared to several state-of-the-art cross-domain recommendation frameworks.

**Index Terms**—Cross-domain recommendation systems, coupled autoencoders, latent factor models, deep learning.

## I. INTRODUCTION

**R**ECOMMENDER systems are automated applications that suggest products to consumers based on their observed interests [1]–[3]. A user’s preferences in items is stored in the form of interaction, such as numerical rating, within a rating matrix. As a result, users, items, and the rating matrix form a domain [4]. The issues of cold start, sparsity, inclusion of new customers or products, and so on may compromise the performance of recommenders [5]–[7]. While these problems are being studied from a single domain viewpoint, cross-domain recommender systems (CDRS) bring a different perspective to their solution [8].

The challenge of recommending specific items to consumers in a target domain (e.g., a resource-scarce market) by using data from neighboring high-resource domains, e.g., using data from a much larger market to boost recommendations in

a target market, is central to the principle of cross-domain recommendation [9]–[13]. We hypothesize that data from one domain can be used to boost advice in a different domain. Such an approach has attracted the interest of many researchers in the recent years, e.g., [14]–[22].

The cross-domain recommendation problem (CDR) has been explored under different perspectives and scenarios. In general, CDR methods can be divided into three major categories, that is content-based frameworks, embedding-based frameworks and rating pattern-based approaches [16]. *Content-based* approaches examine the CDR problem from a content-level relevance point of view. Particularly, this type of methods aim to link various domains by capturing and utilizing similar content information, such as, user-generated reviews [19]. Contrary to these methodologies, embedding-based approaches explore the CDR problem from a user-relevance or item-relevance perspective. Exploiting common users and/or common items, this category extracts embedding knowledge (e.g., user/item latent factors) and then transfer it across domains through domain adaptation techniques such as neural networks [23], [24] and transfer learning [25]–[27]. Finally, rating pattern-based approaches aim to transfer information such as rating patterns from the source to the target domain [28]–[30].

The proposed methods belong to the category of embedding-based approaches. In more detail, we explore an item-level relevance cross-domain recommendation task. We assume two related domains, that is, the source and the target domain, which contain common items without sharing any additional information regarding the users’ behavior, and thus avoiding the leak of user privacy. Our contribution concerns two novel coupled autoencoder-based deep learning methods for cross-domain recommendation:

- The first method, dubbed CACDR (Coupled Autoencoder Cross - Domain Recommendation) aims to simultaneously learn a pair of autoencoders in order to reveal the intrinsic representations of the items in the source and target domains along with a coupled mapping function to model the non-linear relationships between these representations; thus it is able to transfer beneficial information from the source to the target domain.
- The second method, dubbed LFACDR (Latent Factor Autoencoder Cross-Domain Recommendation) is derived based on a new joint regularized optimization problem, which employs two autoencoders to generate in a deep and non-linear manner the user and item-latent factors; at the same time a data-driven function is learnt to map the item-latent factors across domains.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

A. Gkillas is with Graduate department of Electrical and Computer Engineering, Patras University, Patras, Greece

D.Kosmopoulos is with the Faculty of Department of Computer Engineering and Informatics, Patras University, Patras, Greece.

Different from other studies we optimize the autoencoders jointly, thus learning in an end-to-end fashion the intrinsic relationships across domains. Furthermore, we don't use scenarios with shared users across domains requiring the transfer of sensitive information regarding the user's behavior, but we consider the item-relevance scenario, which preserves the user's privacy since we share only the low dimensional item-latent factors that have no information about the users.

The rest of the paper is organized as follows: section II gives an overview of the research related to ours; section IV analyzes the proposed methods; section V validates experimentally the proposed methods using two public datasets as source and target domains; finally, section VI summarizes our contributions and gives future directions.

## II. RELATED WORKS

In literature, there are plethora of studies attempting to address the challenging recommendation issues that emerge, that is, the data sparsity and the cold start by developing CDR strategies. In recent years, the problem of CDR has been tackled from multiple perspectives and different assumptions, thus rendering this problem particularly difficult to describe under a unique generic framework [15], [16]. To that end, in this section several representative studies are briefly presented.

Specifically, Singh et al. [31] used a matrix factorization approach to transfer information across domains by sharing the user latent factor. Pan et al. [32] employed a principled matrix-based transfer learning methodology to extract and transfer knowledge concerning the users and items from the source to the target domain. Agarwal, et al. [33] proposed a collective matrix factorization method exploiting correlated information across domains via localized factor models to tackle the sparsity problem in the target domain. Moreno et al. [18] exploited the information from multiple domains in order to improve the recommendation accuracy for the target domain. Lian et al. [34] combined collaborative filtering and content-based filtering into a multi-view neural network to tackle the CDR problem. Aiming to overcome the data sparsity problem other studies such as [35], [36], [37], [38] employed cluster-level matrix factorization techniques to share common information between users and items across domains.

Recently, Man et al. [24] used a matrix factorization method under the user sharing assumption to extract the latent factors models and a multi-layer perceptron to model and transfer valuable knowledge across domains. Kang et al. proposed a semi-supervised mapping to recommend items for cold start users by exploiting the distribution of shared users across domains. Elkahky et al. [39] proposed a deep learning methodology to map shared users and items to a hidden space where the similarity between users and items is maximized. Zhong et al. [17] utilized a deep learning architecture based on the autoencoders and an attention mechanism to extract and fuse information from multiple closely-related domains, thus enhancing the rating prediction accuracy. Zhu et al. [40] proposed a graphical and attentional framework for the CDR problem based on the rating and content information across domains. Additionally, Kanagawa et al. [41] employed an

unsupervised domain adaptation approach by reformulating the recommendations as an extreme classification task. Zhao et al. [42] captured the interactions of different domains as a whole, and propagated user preferences, based on graph neural networks. He et al. [22] proposed a codebook transfer learning procedure to learn the proper codebook scale balancing both the computational complexity and prediction accuracy for CDR. Gao et al. [21] examined the CDR from a data privacy perspective without sharing any information about the users' data. Ma et al. [43] addressed the problem of insufficient common users by employing a fully connected trust-aware deep learning framework to discover the intrinsic relationships between common and non-common users. Iwata et al. [14] proposed a CDR architecture assuming that the user and item-latent factor models in different domains derived from a common Gaussian distribution. Hu et al. [44] used deep cross connection networks to exploit and transfer information across the domains. Li et al. [45] exploited the merits of the dual transfer learning and the latent embedding methodology to tackle the CDR problem. In more details, an orthogonal matrix was employed to transfer the knowledge from the source to target domain. Yuan et al. [30] utilized a deep domain adaption model to extract and transfer patterns from rating matrices in different domains, without considering any auxiliary information. Contrast to above-mentioned methods, other studies focused on extracting and utilizing content information to tackle the CDR problem. Xin et al. [19] proposed a CDR framework utilizing review text to alleviate data sparsity limitations. Along the lines of the previous method, Fu et al. [20] utilized stacked denoising autoencoders to fuse review text with the rating matrices to tackle the data sparsity and cold start problems in the target domain. Zhao et al. [46] extracted multiple aspects of users and items based on review documents aiming to learn aspect correlations across domains via an attention mechanism.

Similar to our work, the study in [24] belongs to the embedding-based frameworks considering also an item-level relevance cross-domain recommendation task. However, this approach employs matrix factorization frameworks to extract the user and item-latent factors, thus rendering it limited only to capture linear and rather shallow features from the complex and non-linear collaborative relationships of the users and items. Furthermore, taking into consideration that the CDR problem is a domain adaptation procedure the learning of the latent factors of the source and target domains independently may result in poor performance, since there is no influence or transferred knowledge between the domains during the learning stage.

Different from the above-mentioned approach, in our study the goal is to capture and model the underlying relationships between the users and items from the source and target domain in a deep and non-linear manner; we employ two novel cross-domain recommendation frameworks based on the coupled autoencoders. We optimize jointly the autoencoders of the source and the target domain and this way we can transfer valuable information across domains during the training stage. Autoencoders have also been employed in other studies addressing different cross-domain recommendation scenarios,

such as multi-domain recommendation tasks [17], [45], [30] and content-based recommendation tasks [19]. *Nevertheless, in these methods the autoencoders of the source and target domains are trained independently. This procedure is piecemeal and thus sub-optimal, since there is no transfer or coupled learning the source domain and the target domain and hence no influence from one to another during the training process.* Contrary to these learning procedures, in this study we argue that better and more meaningful intrinsic representations can be derived, not only based on the available input data in each domain separately, but also taking into account the internal relationships that exist across domains during the learning of the autoencoders.

### III. NOTATIONS AND PROBLEM FORMULATION

Table I summarizes all the required notations of this study. In the under-examined CDR problem an item-level relevance scenario is considered in which there are two domains containing the same items (e.g., movies) without necessarily sharing the same users. In the literature the two domains are often referred as the 'source' and 'target' domains. Without loss of generality, let  $\mathbf{R}^s \in \mathbb{R}^{m \times n}$  and  $\mathbf{R}^t \in \mathbb{R}^{m \times n}$  be the rating matrices representing the ratings between  $m$  items and  $n$  users for the source and the target domain respectively, where  $\mathbf{R}^s(i, j)$  represents the rating of the user  $j$  for the item  $i$  in the source domain and  $\mathbf{R}^t(i, j)$  represents the corresponding rating in the target domain. Furthermore, we denote as  $\mathbf{M}^s = [\mathbf{m}_1^s; \mathbf{m}_2^s, \dots] = \mathbf{R}^S$ ,  $\mathbf{M}^t = [\mathbf{m}_1^t; \mathbf{m}_2^t, \dots] = \mathbf{R}^t$  the item rating matrix of the source and target domain and  $\mathbf{U}^s = [\mathbf{u}_1^s; \mathbf{u}_2^s, \dots] = \mathbf{R}^{sT}$ ,  $\mathbf{U}^t = [\mathbf{u}_1^t; \mathbf{u}_2^t, \dots] = \mathbf{R}^{tT}$  the user rating matrix of the source and the target domain, respectively. In general, the item rating vector  $\mathbf{m}_i^s \in \mathbb{R}^{1 \times n}$  describes the rating relationship between the item  $i$  and all the users in the source domain, whereas the user rating vectors  $\mathbf{u}_i^s \in \mathbb{R}^{1 \times m}$  describes the rating relationship between the user  $i$  and all the items of the source domain. Accordingly, the item and user rating vectors  $\mathbf{m}_i^t$ ,  $\mathbf{u}_i^t$  represent the corresponding rating relations of the target domain.

Taking into consideration that both domains share the same items, our primary goal is to exploit and extract knowledge from the source domain and transfer it to the target domain; this way it is possible to make recommendations for items with no ratings or little information, thus tackling the data sparsity and the cold-start problem in target domain. In more detail, this scenario can be seen as a domain adaptation procedure (transfer learning) [47], which aims to describe the unknown mathematical relationships between the source and target domains. Nonetheless, tackling this kind of problem two major questions emerged and need to be answered: (i) what to transfer - which information is beneficial to transfer across the domains; and (ii) how to transfer - which learning procedure could be employed to transfer the knowledge. To this end, we address these crucial questions by developing two novel CDR frameworks based on a coupled autoencoder approach.

### IV. PROPOSED MODELS

In this section, we derive two coupled autoencoder frameworks that can be used for the CDR problem. Specifically,

TABLE I  
MATHEMATICAL NOTATIONS

Symbol	Description
<i>Source Domain</i>	
$n$	The number of users of the source and target domain
$m$	The number of items of the source and target domain
$\mathbf{R}^s \in \mathbb{R}^{m \times n}$	The rating matrix of the source domain
$\hat{\mathbf{R}}^s \in \mathbb{R}^{m \times n}$	The predicted rating matrix of the source domain
$\mathbf{U}^s \in \mathbb{R}^{n \times m}$	The user rating matrix of the source domain
$\mathbf{M}^s \in \mathbb{R}^{m \times n}$	The item rating matrix of the source domain
$\mathbf{Y}_e^s \in \mathbb{R}^{n \times k}$	The output of the encoder of $\mathbf{U}^s$
$\mathbf{X}_e^s \in \mathbb{R}^{m \times k}$	The output of the encoder of $\mathbf{M}^s$
$\hat{\mathbf{U}}^s \in \mathbb{R}^{n \times m}$	The output of the decoder of $\mathbf{Y}_e^s$
$\hat{\mathbf{M}}^s \in \mathbb{R}^{m \times n}$	The output of the decoder of $\mathbf{X}_e^s$
<i>Target Domain</i>	
$\mathbf{R}^t \in \mathbb{R}^{m \times n}$	The rating matrix of the target domain
$\hat{\mathbf{R}}^t \in \mathbb{R}^{m \times n}$	The predicted rating matrix of the target domain
$\mathbf{U}^t \in \mathbb{R}^{n \times m}$	The user rating matrix of the target domain
$\mathbf{M}^t \in \mathbb{R}^{m \times n}$	The item rating matrix of the target domain
$\mathbf{Y}_e^t \in \mathbb{R}^{n \times k}$	The output of the encoder of $\mathbf{U}^t$
$\mathbf{X}_e^t \in \mathbb{R}^{m \times k}$	The output of the encoder of $\mathbf{M}^t$
$\hat{\mathbf{U}}^t \in \mathbb{R}^{n \times m}$	The output of the decoder of $\mathbf{Y}_e^t$
$\hat{\mathbf{M}}^t \in \mathbb{R}^{m \times n}$	The output of the decoder of $\mathbf{X}_e^t$

the first one, named CACDR employs a coupled autoencoder method to capture and model the complex relationships between the users and items from the source and target domain, while the second one, named LFACDR can be considered as an expansion of the former one utilizing the autoencoders in order to learn in a deep and non-linear manner the user and item-latent factors models in the respective domains. After an initial modeling of domain-specific information in the source, both methods transfer that information to the target domain via a multi-layer perceptron network. It should be highlighted that only the item latent factors are transferred across domains, without sharing any sensitive information regarding the user's behavior and violating privacy politics.

#### A. The CACDR Method

Autoencoders have demonstrated ground-breaking performance in the unsupervised feature learning domain. Formally, the autoencoder aims to reveal and describe the intrinsic hidden representation of the input by copying its input to its output [48]. However, the autoencoder as a single domain procedure produces intrinsic representations based only on the input data, thus ignoring the valuable underlying relationships that exist across multiple domains. On the other hand, the coupled autoencoder model is able to capture these internal relationships and better representations can be derived as the domains influence each other. In particular, the proposed coupled autoencoder based method for CDR, called CACDR consists of three stages. The first stage employs two autoencoders to reveal and learn the intrinsic hidden representations of the item rating matrices, namely  $\mathbf{M}^s$ ,  $\mathbf{M}^t$  of the source and target domain, respectively. The second stage uses a multi-layer Perceptron network (MLP) to model the relationship across domains by learning a mapping function between the intrinsic representations of the source and target domain. In the previous two stages the autoencoders and the mapping

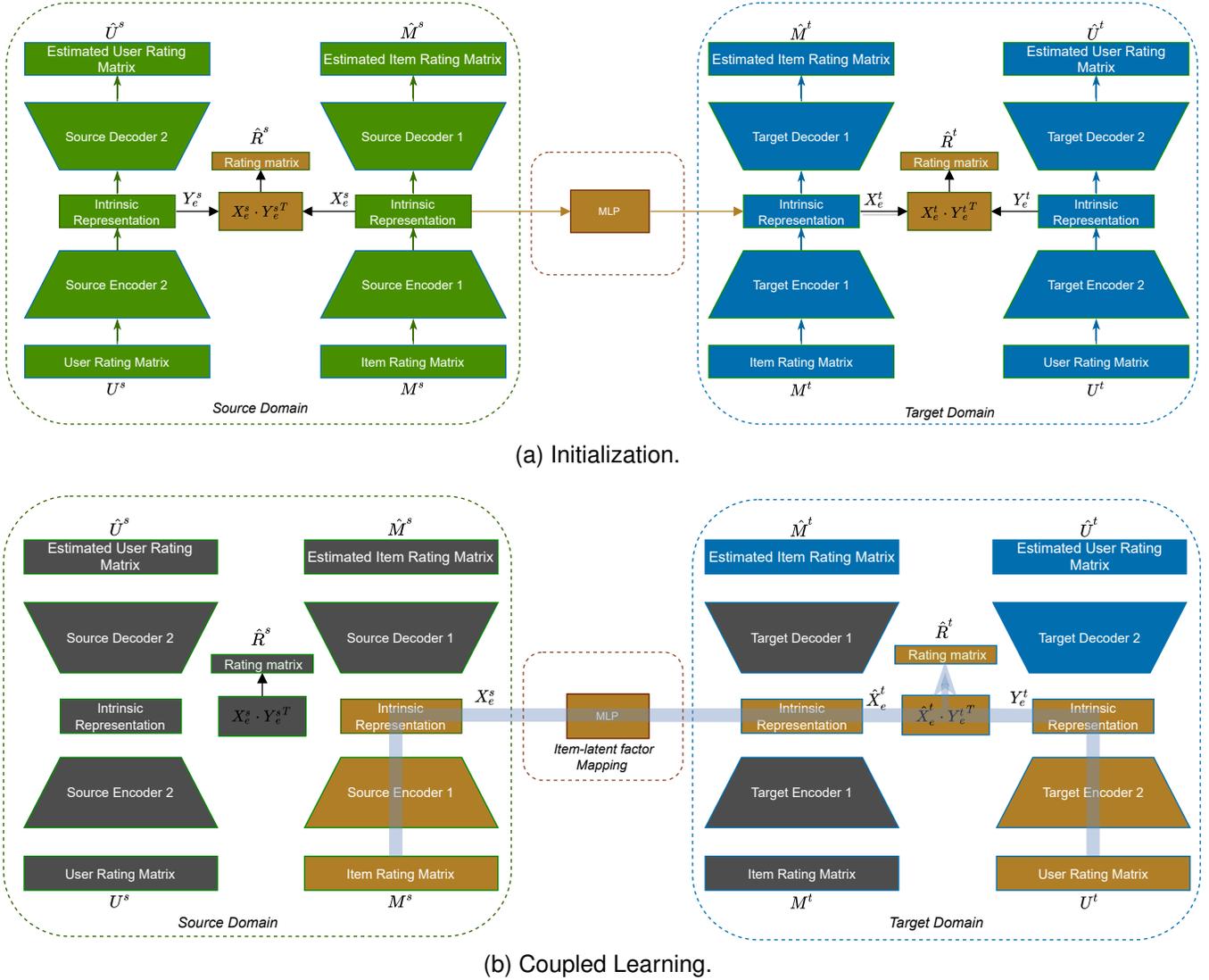


Fig. 1. An illustration of our proposed CACDR model for cross-domain recommendation. (a) Initialization: First the autoencoders are trained to learn the intrinsic representations of the source and target domain (stage 1) and then a mapping function (MLP) is learnt between these representations (stage 2). (b) Coupled Learning: since the autoencoders are trained independently and there is no transfer learning across domains, a coupled autoencoder model is employed in order to jointly optimize all the active parts of the autoencoders involved in the rating prediction in target domain (stage 3).

function are trained independently. So we introduce coupling at the third stage in order to capture the underlying complex relationships across domains and transfer beneficial knowledge from one domain to another during the training procedure. Finally, recommendations can be made for a new item in the target domain based on the intrinsic representations of the same item in the source domain. The complete proposed methodology is depicted in Fig. 1.

1) *Coupled Autoencoder-Based*: Let  $M^s, M^t$  be the item rating matrices of the source and target domains respectively. Then the corresponding source and target autoencoders, which learn the hidden intrinsic representations of the two item matrices, can be obtained by minimizing the following reconstruction errors:

$$\|M^s - \hat{M}^s\|_F^2 \quad \& \quad \|M^t - \hat{M}^t\|_F^2 \quad (1)$$

where  $\hat{M}^s, \hat{M}^t$  denote the estimated item rating matrices of the source and the target domain, respectively. Formally, the autoencoder comprises of the encoding  $\mathcal{E}(\cdot)$  and decoding  $\mathcal{D}(\cdot)$  process.

$$X_e^s = \mathcal{E}^s(M^s)$$

or equivalently,

$$\begin{aligned} X_{e,1}^s &= \varphi(W_{e,1}^s M^s + b_{e,1}^s) \\ X_{e,2}^s &= \varphi(W_{e,2}^s X_{e,1}^s + b_{e,2}^s) \\ &\dots \\ X_e^s &= \varphi(W_{e,L}^s X_{e,L-1}^s + b_{e,L}^s), \end{aligned} \quad (2)$$

where  $W_{e,i}^s, b_{e,i}^s$  ( $i = 1, \dots, L$ ) denote the weight matrices and the bias terms for the encoding layers of the source

autoencoder,  $\varphi(\cdot)$  is the activation function ReLU,  $L$  stands for the number of hidden layers,  $\mathbf{X}_e^s \in \mathbb{R}^{m \times k}$  is the output of the source encoder  $\mathcal{E}^s(\cdot)$  and  $k \ll n$ .

Similarly, the intrinsic representation of the target item rating matrix  $\mathbf{M}^t \in \mathbb{R}^{m \times n}$  can be defined as

$$\mathbf{X}_e^t = \mathcal{E}^t(\mathbf{M}^t)$$

or equivalently,

$$\begin{aligned} \mathbf{X}_{e,1}^t &= \varphi(\mathbf{W}_{e,1}^t \mathbf{M}^t + b_{e,1}^t) \\ \mathbf{X}_{e,2}^t &= \varphi(\mathbf{W}_{e,2}^t \mathbf{X}_{e,1}^t + b_{e,2}^t) \\ &\dots \\ \mathbf{X}_e^t &= \varphi(\mathbf{W}_{e,L}^t \mathbf{X}_{e,L-1}^t + b_{e,L}^t), \end{aligned} \quad (3)$$

where  $\mathbf{W}_{e,i}^t, b_{e,i}^t$  ( $i = 1, \dots, L$ ) denote the weight matrices and the bias terms for the encoding layers of the target autoencoder,  $\varphi(\cdot)$  is the activation function ReLU,  $L$  stands for the number of hidden layers,  $\mathbf{X}_e^t \in \mathbb{R}^{m \times k}$  denotes the output of the target encoder  $\mathcal{E}^t(\cdot)$  and  $k \ll m$ .

**Decoding process:** Accordingly, the reconstructed item rating matrices  $\hat{\mathbf{M}}^s, \hat{\mathbf{M}}^t$  can be derived by

$$\hat{\mathbf{M}}^s = \mathcal{D}^s(\mathbf{X}_e^s) \quad (4)$$

$$\hat{\mathbf{M}}^t = \mathcal{D}^t(\mathbf{X}_e^t), \quad (5)$$

where  $\mathcal{D}^s(\cdot)$  denotes the decoding procedure of the source autoencoder and  $\mathcal{D}^t(\cdot)$  is the decoding procedure of the target autoencoder. Note that both decoders consist also of  $L$  fully connected layers.

To put in a nutshell, the encoding procedure aims to learn a concrete representation of the input in order to capture the complex relationships between the items and users. On the other hand, the decoding process seeks to decode the hidden representations back to the original item rating matrices. The accurate decoding procedure enables the autoencoder to learn the rating patterns between items and users and make rating predictions for new items and users.

2) *Non Linear Mapping:* After obtaining the intrinsic representations of the item rating matrices, an MLP is employed to capture and model the underlying relationship between the intrinsic representations of the source and target domain ( $\mathbf{X}_e^s, \mathbf{X}_e^t$ ), thus transferring the appropriate knowledge from the source to target domain. Mathematically, the non linear mapping function  $\mathcal{F}(\cdot)$  can be written as

$$\hat{\mathbf{X}}_e^t = \mathcal{F}(\mathbf{X}_e^s)$$

or equivalently,

$$\begin{aligned} \mathbf{X}_{e,1}^t &= \varphi(\mathbf{W}_{p,1} \mathbf{X}_e^s + b_{p,1}) \\ \mathbf{X}_{e,2}^t &= \varphi(\mathbf{W}_{p,2} \mathbf{X}_{e,1}^t + b_{p,2}) \\ &\dots \\ \hat{\mathbf{X}}_e^t &= \varphi(\mathbf{W}_{p,L} \mathbf{X}_{e,L-1}^t + b_{p,L}) \end{aligned} \quad (6)$$

where  $\mathbf{W}_{p,i}, b_{p,i}$  ( $i = 1, \dots, L$ ) denote the weight matrices and the bias terms,  $\varphi(\cdot)$  is the activation function ReLU,  $L$  stands for the number of hidden layers and  $\hat{\mathbf{X}}_e^t$  is the estimated intrinsic representation of the target domain.

3) *Cross-domain Rating Predictions:* The goal of the proposed framework is to recommend new items in the target domain leveraging upon the knowledge of the same items belonging in the source domain. In particular, given an item  $j$  in the target domain, the following methodology is used to recover its predicted rating:

1. The same item is found in the source domain and its intrinsic representation is obtained by employing the autoencoder of the source domain (encoding procedure) according to equation (2).
2. The corresponding intrinsic representation of the item in the target domain can be estimated via the intrinsic representation of the item in the source domain and the MLP network based on relation (6).
3. Finally, the predicted rating of the item in the target domain is recovered based on target autoencoder (decoding procedure) according to relation (5).

4) *Model Learning:* The most critical part of the proposed architecture is the optimization and coupling of the autoencoders along with the mapping function. However, by learning the autoencoders first and then the mapping function (based on the estimated intrinsic representations) may lead to poor performance, since the autoencoders and the mapping function are optimized independently. In other words, there is no transfer or coupled learning between the source domain (i.e., source item rating matrix) and the target domain (i.e., target item rating matrix). This procedure is piecemeal and thus sub-optimal, since there is no influence from one to another during the training process. Nevertheless, this methodology can be used as initialization process of the model.

In light of the fact that the ultimate goal of the CACDR method is to efficiently predict the item ratings of the target domain (or the item rating matrix,  $\mathbf{M}^t = \mathbf{R}^t$ ), the proposed objective function for optimizing jointly the two autoencoders (source and target) along with the MLP network may be written as:

$$\begin{aligned} &\left\| \mathbf{M}^t - \hat{\mathbf{M}}^t \right\|_F^2 \xrightarrow{(5)} \left\| \mathbf{M}^t - \mathcal{D}^t(\hat{\mathbf{X}}_e^t) \right\|_F^2 \xrightarrow{(6)} \\ &\left\| \mathbf{M}^t - \mathcal{D}^t(\mathcal{F}(\mathbf{X}_e^s)) \right\|_F^2 \xrightarrow{(2)} \left\| \mathbf{M}^t - \mathcal{D}^t(\mathcal{F}(\mathcal{E}^s(\mathbf{M}^s))) \right\|_F^2. \end{aligned} \quad (7)$$

Note that now in relation (7) the source encoder  $\mathcal{E}^s(\cdot)$ , the mapping neural network  $\mathcal{F}(\cdot)$  and the target decoder  $\mathcal{D}^t(\cdot)$  are all explicitly involved in the reconstruction of the desired output  $\hat{\mathbf{M}}^t$ . Hence, in order to couple the two autoencoders with the mapping function a **coupled deep network** is employed, where its first network component is the source encoder; the second network component is the mapping neural network and its final network component is the target decoder. Fig. 1b illustrates the proposed coupled architecture. Having obtained, the stacked network architecture the back-propagation algorithm is used to optimize (7). Algorithm 1 summarizes the proposed methodology.

## B. The LFACDR Method

As previously mentioned, the autoencoders constitute an ideal mathematical tool to reveal and learn complex low di-

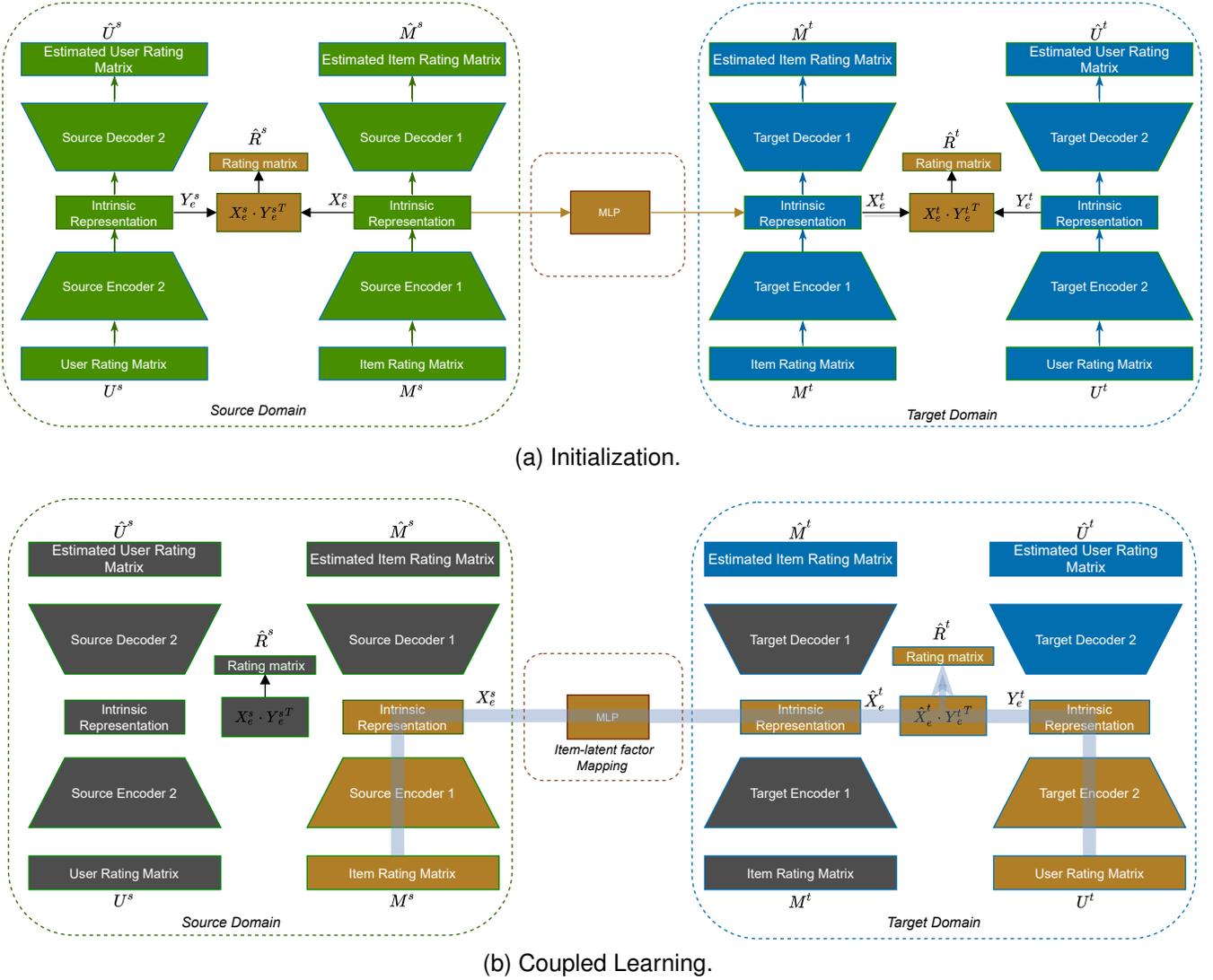


Fig. 2. An illustration of the proposed LFACDR model for cross-domain recommendation. (a) Initialization: First, the autoencoders are trained to obtain the item and user-latent factors of the source and target domain (stage 1) and then a mapping function (MLP) is learnt between the item latent factor matrices of the source and target domain (stage 2). (b) Coupled Learning: A coupled autoencoder model is employed in order to jointly optimize all the active parts of the autoencoders involved in the rating prediction in target domain (stage 3).

mensional representations while at the same time they preserve the underlying structure of the input data. This consideration motivates the ensuing cross domain recommendation methodology that a joint optimization problem is proposed in order to recover in a deep and non-linear manner the user and item-latent factors in both source and target domains. To this end, for each domain two autoencoders are employed to jointly learn the intrinsic representations of user and item rating matrices and decompose the rating matrix into two low-rank matrices, that is the user and item-latent factor matrix. The proposed framework is shown in Fig. 2.

1) *Latent Factor Modeling based on Autoencoders*: Consider the user rating matrix  $U^s \in \mathbb{R}^{n \times m}$ , the item rating matrix  $M^s \in \mathbb{R}^{m \times n}$  and the rating matrix  $R^s \in \mathbb{R}^{n \times m}$  of the source domain. To obtain the user and item-latent factors of the source domain the following joint constrained optimization problem is proposed, which includes one autoencoder for the items and one autoencoder for the users. Hence, the proposed

problem is formulated as

$$\begin{aligned} \mathcal{L}^s &= \left\| M^s - \hat{M}^s \right\|_F^2 + \left\| U^s - \hat{U}^s \right\|_F^2 = \\ & \left\| M^s - \mathcal{D}_m^s(X_e^s) \right\|_F^2 + \left\| U^s - \mathcal{D}_u^s(Y_e^s) \right\|_F^2 \\ & \text{s.t. } X_e^s Y_e^{sT} = R^s, X_e^s = \mathcal{E}_m^s(M^s), Y_e^s = \mathcal{E}_u^s(U^s) \end{aligned} \quad (8)$$

where  $\mathcal{D}_m^s(\cdot)$ ,  $\mathcal{E}_m^s(\cdot)$  denote the decoding and encoding procedure of the first source autoencoder (items),  $X_e^s \in \mathbb{R}^{m \times k}$  is the item-latent factor matrix derived from the output of the first encoder,  $\mathcal{D}_u^s(\cdot)$ ,  $\mathcal{E}_u^s(\cdot)$  stand for the decoding and encoding procedure of the second source autoencoder (users) and  $Y_e^s \in \mathbb{R}^{n \times k}$  denotes the user-latent factor matrix derived from the output of the second encoder. It should be highlighted that the encoding and decoding processes of two autoencoders can be written alternatively according to equations (2) and (4).

The two autoencoders aim to learn jointly the intrinsic representations of the item and user rating matrices. At the same

time, through the proposed constraint optimization problem these intrinsic representations that is the  $\mathbf{X}_e^s$  and  $\mathbf{Y}_e^s$  act as the desired item and user-latent factor matrices, respectively. Note that similar procedure can be employed to derive the corresponding item and user-latent factor matrices  $\mathbf{X}_e^t$  and  $\mathbf{Y}_e^t$  of the target domain.

2) *Non Linear Mapping*: Having acquired the latent factor matrices  $\{\mathbf{X}_e^s, \mathbf{Y}_e^s, \mathbf{X}_e^t, \mathbf{Y}_e^t\}$  of items and users in the source and target domain, similar to the previous proposed method, an MLP is again used to learn the mapping function between the item latent factor matrices of the source and target domain ( $\mathbf{X}_e^s, \mathbf{X}_e^t$ ). This can be expressed as follows,

$$\hat{\mathbf{X}}_e^t = \mathcal{F}(\mathbf{X}_e^s)$$

or equivalently,

$$\begin{aligned} \mathbf{X}_{e,1}^t &= \varphi(\mathbf{W}_{p,1}\mathbf{X}_e^s + b_{p,1}) \\ \mathbf{X}_{e,2}^t &= \varphi(\mathbf{W}_{p,2}\mathbf{X}_{e,1}^t + b_{p,2}) \\ &\dots \\ \hat{\mathbf{X}}_e^t &= \varphi(\mathbf{W}_{p,L}\mathbf{X}_{e,L-1}^t + b_{p,L}) \end{aligned} \quad (9)$$

where  $\mathbf{W}_{p,i}, b_{p,i}$  ( $i = 1, \dots, L$ ) denote the weight matrices and the bias terms,  $\varphi(\cdot)$  is the activation function ReLU,  $L$  stands for the number of hidden layers and  $\hat{\mathbf{X}}_e^t$  denotes the estimated item latent factor matrix of the target domain.

3) *Rating prediction*: In general, given a new item in the target domain with little information, we are not able to calculate an accurate latent factor for making recommendation. In light of this, the corresponding latent factor is learnt from the source domain and a latent factor is derived for the same item in the target domain via the mapping function (9). To be more specific, the predicted rating between item  $i$  and user  $j$  in the target domain is given by the following relation,

$$\hat{\mathbf{R}}^s(i, j) = \hat{\mathbf{X}}_e^s(i, :) \mathbf{Y}_e^s(j, :)^T \quad (10)$$

where  $\hat{\mathbf{X}}_e^s(i, :) \in \mathbb{R}^{1 \times k}$  denote the estimated item-latent factor of item  $i$  in the target domain based on the corresponding item-latent factor of item  $i$  in the source domain via relation

(9) and  $\hat{\mathbf{Y}}_e^s(i, :) \in \mathbb{R}^{1 \times k}$  stands of the row vector of matrix  $\mathbf{Y}_e^s$ , representing the user-latent factor of user  $j$  in the target domain.

4) *Model Learning*: Similar to the previous procedure the training can be divided into two stages: the first stage is the initialization and the second stage is the coupling.

*Initialization*: First, the autoencoders are trained to obtain the item and user-latent factors of the source and target domain and then the mapping function is learnt.

*Coupling*: Since the aim of the proposed method is to accurately predict the ratings of new items in the target domain, the objective function for jointly optimizing the autoencoders extracting the item and user-latent factor matrices along with the MLP network is given by

$$\begin{aligned} \|\mathbf{R}^t - \hat{\mathbf{R}}^t\|_F^2 &\Rightarrow \|\mathbf{R}^t - \hat{\mathbf{X}}_e^t (\mathbf{Y}_e^t)^T\|_F^2 \stackrel{(9)}{\Rightarrow} \\ &\|\mathbf{R}^t - \mathcal{F}(\mathbf{X}_e^s) (\mathbf{Y}_e^t)^T\|_F^2 \end{aligned} \quad (11)$$

where

$$\mathbf{X}_e^s = \mathcal{E}_m^s(\mathbf{M}^s) \ \& \ \mathbf{Y}_e^t = \mathcal{E}_u^t(\mathbf{U}^t).$$

From relation (11) it is easy to verify that the source encoder for the items  $\mathcal{E}_m^s(\cdot)$ , the target decoder for the users  $\mathcal{E}_u^t(\cdot)$  and MLP network are all explicitly involved in the reconstruction of the desired output  $\hat{\mathbf{R}}^t$ . Thus, the three network units can be coupled together by jointly optimizing them through the back propagation algorithm. Fig. 2b demonstrates the proposed coupled framework. The overall methodology is summarized in Algorithm 2.

## V. EXPERIMENTAL VALIDATION

To validate the efficacy and applicability of the two proposed methods, extensive experiments were conducted in the context of the CDR problem. In particular, this study examines an item-level relevance cross-domain recommendation scenario, where two domains (source and target) share common items (e.g., movies) and contain different users.

---

### Algorithm 1 : CACDR method

---

**Require:** The item rating matrices of the source and target domain  $\mathbf{M}^s \in \mathbb{R}^{m \times k}$ ,  $\mathbf{M}^t \in \mathbb{R}^{m \times k}$

**Ensure:** The predicted rating matrix of the target domain  $\hat{\mathbf{R}}^t \sim \hat{\mathbf{M}}^t$

{**Stage A: Initialization**}

- 1: Initialize the source domain autoencoder by learning the intrinsic representation  $\mathbf{X}_e^s \in \mathbb{R}^{m \times k}$  of the matrix  $\mathbf{M}^s$  via (2).
- 2: Initialize the target domain autoencoder by learning the intrinsic representation  $\mathbf{X}_e^t \in \mathbb{R}^{m \times k}$  of the matrix  $\mathbf{M}^t$  via (3).
- 3: Initialize the MLP network by learning the mapping function from  $\mathbf{X}_e^s$  to  $\mathbf{X}_e^t$  via (6).

{**Stage B: Coupled Learning**}

- 4: Construct the CACDR model via (7) and make recommendation for items in the target domain.
- 

---

### Algorithm 2 : LFACDR method

---

**Require:** The item rating matrices of the source and target domain  $\mathbf{M}^s \in \mathbb{R}^{m \times n}$ ,  $\mathbf{M}^t \in \mathbb{R}^{m \times n}$  and corresponding user rating matrices  $\mathbf{U}^s \in \mathbb{R}^{n \times m}$ ,  $\mathbf{U}^t \in \mathbb{R}^{n \times m}$

**Ensure:** The predicted rating matrix of the target domain  $\hat{\mathbf{R}}^t \sim \hat{\mathbf{M}}^t$

{**Stage A: Initialization**}

- 1: Initialize the two source domain autoencoders by learning the item-latent factor matrix  $\mathbf{X}_e^s \in \mathbb{R}^{m \times k}$  and the user-latent factor matrix  $\mathbf{Y}_e^s \in \mathbb{R}^{n \times k}$  via (8).
- 2: Initialize the two target domain autoencoders by learning the item-latent factor matrix  $\mathbf{X}_e^t \in \mathbb{R}^{m \times k}$  and the user-latent factor matrix  $\mathbf{Y}_e^t \in \mathbb{R}^{n \times k}$  via (8).
- 3: Initialize the MLP network by learning the mapping function from  $\mathbf{X}_e^s$  to  $\mathbf{X}_e^t$  via (9).

{**Stage B: Coupled Learning**}

- 4: Construct the LFACDR model via (11) and make recommendation for items in the target domain.
-

TABLE II  
STATISTICS OF THE DATASETS

Datasets	Domains	#Movies	#Users	Density
No. 1	MovieLens	5986	48575	1.83%
	Netflix	5986	49681	2.98%
No. 2	MovieLens	5819	40000	2.80%
	Netflix	5819	40000	5.16%
No. 3	MovieLens	6000	30100	0.64%
	Netflix	6000	30040	0.97%
No. 4	MovieLens	4200	44850	0.38%
	Netflix	4200	43775	0.53%

### A. Experimental Setup

**Datasets:** Two publicly available benchmark datasets were employed to demonstrate the merits of the proposed recommendation frameworks. Namely, we used the MovieLens<sup>1</sup> and Netflix<sup>2</sup> datasets, which contain a large portion of same movies, thereby forming an item-level relevance scenario. According to IMDB information more than 5500 movies are the same across MovieLens and Netflix datasets. In light of this, the MovieLens is taken as source domain and Netflix is used as target domain, seeking to ameliorate the recommendation performance in target domain by exploiting and transferring valuable knowledge from the source domain. Moreover, due to the size of the datasets we followed the same methodology as in [21] by randomly sub-sampling a certain portion of users along with their respective ratings, ensuring users had enough number of interaction in each domain. Following this pre-processing stage we ended up with four different datasets with different sparsity levels. Table II provides the detailed statistics of the four resulting datasets. **Experimental Settings:** In our extensive experiments the two datasets were randomly divided into the training set (80%) and the testing set (20%). Regarding the testing set, we removed all the rating information from the target domain (Netflix), thus considering these entities as cold-start movies. Furthermore, taking into account that different training and testing sets may affect the recommendation performance, this splitting procedure was repeated 10 times and the average results were reported. Finally, it should be noted that we normalized the scale of ratings between 0 and 1 following the same strategy as in [45].

**Parameter Settings:** The parameters of our proposed methods are determined to be ideal via exploration of the parameter space. Concretely, for the first method, referred as **CACDR** the sizes of the autoencoders layers were set to {256, 128, 64, 64, 128, 256}, the sizes of the MLP layers are set to {64, 128} and the batch size was 32. Regarding the training of the CACDR method for the initialization stage the number of epochs is 250, the learning rate was set to  $10^{-3}$  and the  $l_2$  regularizer term was  $10^{-5}$ . Accordingly, for the coupling stage the number of epochs was set to 300, the learning rate and regularization

term was  $10^{-5}$ . To proceed further, for the second proposed method, called **LACDR** the sizes of the autoencoders layers were set to {512, 256, 128, 128, 256, 512}, the sizes of the MLP layers were set to {128, 256} and the batch size was 500. Furthermore, for the initialization training stage the number of epochs was set to 250, while the learning rate and the regularizer term were  $10^{-3}$  and  $10^{-5}$ , respectively. Additionally, for the coupling training stage the number of epochs was set to 300 and the learning rate was  $10^{-5}$ , while the regularizer term remained constant. Finally, the Adam optimizer was employed to train the proposed models and the ReLU was used as activation function.

**Loss function and Evaluation metrics:** Concerning the loss function, in this study we employed the Masked Root Mean Squared Error loss following the approach from [49], since the zero values should be ignored during the training stage of the proposed models. In addition, to evaluate the recommendation performance we adopted the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics.

**Compared methods:** To showcase the added value of the proposed methods (CACDR and LFACDR), we compared with the following CDR frameworks:

- ◊ **DDTCDR** [45]: It is a recent state-of-the-art CDR model. In more details, DDTCDR exploits the merits of the dual transfer learning and the feature embedding method to transfer knowledge across domains. Furthermore it employs a orthogonal mapping to preserve user relations in latent space.
- ◊ **DARec** [30]: Collaborative Cross Networks (CoNet) is able to transfer valuable knowledge across domains by employing deep cross connection networks.
- ◊ **EMCDR** [24]: This CDR framework utilizes a matrix factorization methodology to learn the latent factors and then a multi-layer perceptron is used to model the mapping function between the latent factors of the source and the target domain. This method provides four frameworks and we chose the two best ones, namely the MF\_EMCDR\_MLP employing MF (matrix factorization) as its latent factor modeling and the BPR\_EMCDR\_MLP employing BPR (bayesian personalized ranking) as its latent factor modeling.
- ◊ **CST** [32]: It compacts the sparsity problem and enhances the recommendation performance by transferring the latent factors obtained from the source domain into the target domain. The model employs matrix factorization to deduce the user and item-latent factors in the source domain, and transfer them into the target domain via a regularization method.
- ◊ **LFM** [33]: It uses a collective matrix factorization method exploiting correlated information across domains via Localized factor models. Each user and item has a global latent factor common across domains.

### B. Performance Evaluation

Table III summarizes the average performance results in terms of RMSE and MAE for the two examined datasets

<sup>1</sup><https://grouplens.org/datasets/movielens/>

<sup>2</sup><https://www.kaggle.com/netflix-inc/netflix-prize-data>

presented in Table II. It is evident that the proposed coupled autoencoder-based frameworks, (i.e., CACDR and LFACDR) gave better results than the other CDR methods. Moreover, it is noteworthy that both proposed methods (and especially the LFACDR method) were able to maintain low RMSE and MAE values for different sparsity values of the datasets compared to the other baseline models, where their performance degraded for high levels of sparsity level. Although the DDTCDR and DARec models exhibited good results, our models performed even better and that can be attributed to the following reasons. Firstly, EMCDCR methods utilize matrix factorization techniques to obtain the latent factor models, thus these methods can only capture rather shallow and linear characteristics from the datasets compared to our models that employ deep coupled autoencoders allowing them to capture more complex and non-linear features from the collaborative relationships of the users and items. Secondly, although the DDTCDR and DaRec models employ non-linear functions to extract the latent factors (autoencoders) the learning of the latent factors of the source and target domain along with the mapping function are learnt separately, and hence the transferred knowledge between the domains during the learning stage is rather limited. Different from that, due to the coupled learning stage analyzing in Sections IV-A4 and IV-B4, the autoencoders of the source and the target domain were optimized jointly and the non-linear relations across domains could be transferred much more effectively during the training stage.

### C. Impact of Coupled learning stage

To demonstrate the impact of the coupled learning on the recommendation performance of the proposed methods, we conduct some experiments with and without the coupled learning procedure during the training procedure of our methods. According to Table IV, the coupled learning notably improves the performance of the proposed methods, thus validating our claims that the coupled autoencoders are able to capture not only the existing relationships in each domain separately, but more importantly to model the underlying relationships between the source and target domains.

### D. Impact of Latent Dimension

The latent dimension constitutes a crucial factor effecting the efficacy of different cross-domain recommendation models, hence in this experiment the impact of latent dimension ( $k$ ) on the proposed models is investigated. In more details, fixing the other parameters of our CDR methods, we examined a broad range of latent dimensions ( $k$ ), namely 8, 32, 64, 128, 256. Table V summarizes the results. The best results for the CACDR and LFACDR occurred when the latent dimension was set to 64 and 128 respectively. It should be highlighted that the LFACDR method exhibited better performance compared to the CACDR method in all cases. This finding is mostly attributed to the fact that the LFACDR method exploits not only the information of the items but also the information deriving from the users. Note that only the item latent factors are transferred across domains, without sharing any sensitive information regarding the user's behavior and violating privacy

politics. Additionally, from table V we can deduce that the performance of the proposed models was only slightly affected by the change of the latent dimension, thus indicating their robustness.

## VI. CONCLUSIONS

We have explored an item-level relevance CDR task where the source and the target domain contain common items without sharing any additional information regarding the users' behavior, and thus avoiding the leak of user privacy. We proposed two novel coupled autoencoder-based deep learning methods for CDR that are able to represent the items in the source and target domains along with their coupled mapping function to model the non-linear relationships between these representations. The second method seeks to model the user and item-latent factors, while the first one does not make this assumption. Different from other studies that consider scenarios with shared users across domains requiring the transfer of sensitive information regarding the user's behavior, in our case the considered item-relevance scenario preserves the users' privacy; we share only the low dimensional item-latent factors that have no information about the users.

We demonstrated some very promising results, in comparison to some popular methods on cross-domain recommendation. We used portions of the MovieLens and Netflix datasets with different sparsity levels and quantified the effect on our results. We also demonstrated the effect of learning the mapping function from one domain to the other, which turns out to be a significant part of the proposed method.

## REFERENCES

- [1] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417412002825>
- [2] Q.-Y. Hu, Z.-L. Zhao, C.-D. Wang, and J.-H. Lai, "An item orientated recommendation algorithm from the multi-view perspective," *Neurocomput.*, vol. 269, no. C, p. 261–272, Dec. 2017. [Online]. Available: <https://doi.org/10.1016/j.neucom.2016.12.102>
- [3] J. Ma, J. Wen, M. Zhong, W. Chen, and X. Li, "MMM: Multi-source Multi-net Micro-video Recommendation with Clustered Hidden Item Representation Learning," *Data Science and Engineering 2019 4:3*, vol. 4, no. 3, pp. 240–253, sep 2019. [Online]. Available: <https://link.springer.com/article/10.1007/s41019-019-00101-4>
- [4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, jul 2013.
- [5] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. Association for Computing Machinery, 2011, p. 448–456.
- [6] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Systems with Applications*, vol. 149, p. 113248, jul 2020.
- [7] Z. Zhang, Y. Zhang, and Y. Ren, "Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering," *Information Retrieval Journal 2020 23:4*, vol. 23, no. 4, pp. 449–472, jun 2020. [Online]. Available: <https://link.springer.com/article/10.1007/s10791-020-09378-w>
- [8] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. International World Wide Web Conferences Steering Committee, 2015, p. 278–288.

TABLE III  
RECOMMENDATION PERFORMANCE

Dataset	Metrics	CST	LFM	MF_EMCDR	BPR_EMCDR	DDTCDR	DARec	CACDR	LFACDR
No. 1	RMSE	0.2314	0.2228	0.2079	0.2018	0.1956	0.1868	0.1738	<b>0.1711</b>
	MAE	0.1813	0.1761	0.1643	0.1617	0.1557	0.1496	0.1356	<b>0.1331</b>
No. 2	RMSE	0.2399	0.2341	0.2289	0.2214	0.2032	0.1783	0.1772	<b>0.1740</b>
	MAE	0.1877	0.1819	0.1757	0.1744	0.1648	0.1426	0.1385	<b>0.1360</b>
No. 3	RMSE	0.2578	0.2513	0.2387	0.2324	0.2107	0.2093	0.1980	<b>0.1922</b>
	MAE	0.2181	0.2123	0.1868	0.1849	0.1699	0.1678	0.1560	<b>0.1511</b>
No. 4	RMSE	0.2673	0.2617	0.2494	0.2435	0.2229	0.2221	0.2139	<b>0.2047</b>
	MAE	0.2302	0.2282	0.2047	0.2009	0.1817	0.1789	0.1687	<b>0.1607</b>

TABLE IV  
RECOMMENDATION PERFORMANCE OF THE PROPOSED MODELS WITH AND WITHOUT COUPLED LEARNING STAGE

Dataset	Metrics	CACDR without Coupled learning	CACDR with Coupled learning	LFACDR without Coupled learning	LFACDR with Coupled learning
No. 2	RMSE	0.1827	0.1772	0.1849	0.1740
	MAE	0.1441	0.1385	0.1477	0.1360

TABLE V  
THE IMPACT OF DIFFERENT LATENT DIMENSIONS ON RECOMMENDATION PERFORMANCE OF THE PROPOSED MODELS

Latent Dimension	k = 8		k=32		k=64		k=128		k=256	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
CACDR	0.1784	0.1391	0.1781	0.1389	0.1738	0.1356	0.1792	0.1400	0.1806	0.1413
LFACDR	0.1747	0.1367	0.1739	0.1359	0.1733	0.1354	<b>0.1711</b>	<b>0.1331</b>	0.1742	0.1361

- [9] D. Cao, X. He, L. Nie, X. Wei, X. Hu, S. Wu, and T.-S. Chua, "Cross-platform app recommendation by jointly modeling ratings and texts," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, Jul. 2017. [Online]. Available: <https://doi.org/10.1145/3017429>
- [10] T.-H. Lin, C. Gao, and Y. Li, "Cross: Cross-platform recommendation for social e-commerce," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 515–524. [Online]. Available: <https://doi.org/10.1145/3331184.3331191>
- [11] T. Mei, B. Yang, X.-S. Hua, and S. Li, "Contextual video recommendation by multimodal relevance and user feedback," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, Apr. 2011. [Online]. Available: <https://doi.org/10.1145/1961209.1961213>
- [12] J. Tang, S. Wu, J. Sun, and H. Su, "Cross-domain collaboration recommendation," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1285–1293. [Online]. Available: <https://doi.org/10.1145/2339530.2339730>
- [13] C.-D. Wang, Z.-H. Deng, J.-H. Lai, and P. S. Yu, "Serendipitous recommendation in e-commerce using innovator-based collaborative filtering," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2678–2692, 2019.
- [14] T. Iwata and T. Koh, "Cross-domain recommendation without shared users or items by sharing latent vector distributions," in *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, vol. 38. PMLR, 09–12 May 2015, pp. 379–387.
- [15] I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi, "Cross-domain recommender systems," in *Recommender Systems Handbook, Second Edition*. Springer US, Jan 2015, pp. 919–959.
- [16] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu, "Cross-domain recommendation: Challenges, progress, and prospects," in *IJCAI*, 2021.
- [17] S.-T. Zhong, L. Huang, C.-D. Wang, J.-H. Lai, and P. S. Yu, "An autoencoder framework with attention mechanism for cross-domain recommendation," *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.
- [18] O. Moreno, B. Shapira, L. Rokach, and G. Shani, "TALMUD: Transfer learning for multiple domains," in *ACM International Conference Proceeding Series*. ACM Press, 2012, pp. 425–434.
- [19] X. Xin, Z. Liu, C.-Y. Lin, H. Huang, X. Wei, and P. Guo, "Cross-domain collaborative filtering with review text," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2015, pp. 1827–1834.
- [20] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, "Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems," in *33rd AAAI Conference on Artificial Intelligence*, vol. 33, no. 01. AAAI Press, Jul 2019, pp. 94–101.
- [21] C. Gao, K. Zhao, X. Chen, X. He, D. Jin, F. Feng, and Y. Li, "Cross-domain recommendation without sharing user-relevant data," in *The World Wide Web Conference*. Association for Computing Machinery, Inc, May 2019, pp. 491–502.
- [22] M. He, J. Zhang, and S. Zhang, "ACTL: Adaptive Codebook Transfer Learning for Cross-Domain Recommendation," *IEEE Access*, vol. 7, pp. 19 539–19 549, 2019.
- [23] J. Liu, P. Zhao, F. Zhuang, Y. Liu, V. S. Sheng, J. Xu, X. Zhou, and H. Xiong, "Exploiting aesthetic preference in deep cross networks for cross-domain recommendation," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2768–2774. [Online]. Available: <https://doi.org/10.1145/3366423.3380036>
- [24] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: An embedding and mapping approach," in *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 2464–2470.
- [25] Z. Zhang, X. Jin, L. Li, G. Ding, and Q. Yang, "Multi-domain active learning for recommendation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, p. 2358–2364.

- [26] J. Shang, M. Sun, and K. Collins-Thompson, "Demographic inference via knowledge transfer in cross-domain recommender systems," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1218–1223.
- [27] J. Manotumruksa, D. Rafailidis, C. Macdonald, and I. Ounis, "On cross-domain transfer in venue recommendation," in *Advances in Information Retrieval*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham: Springer International Publishing, 2019, pp. 443–456.
- [28] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, ser. IJCAI'09, 2009, p. 2052–2057.
- [29] B. Loni, Y. Shi, M. Larson, and A. Hanjalic, "Cross-domain collaborative filtering with factorization machines," in *Advances in Information Retrieval*, M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, Eds. Cham: Springer International Publishing, 2014, pp. 656–661.
- [30] F. Yuan, L. Yao, and B. Benatallah, "Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ser. IJCAI'19. AAAI Press, 2019, p. 4227–4233.
- [31] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2008, pp. 650–658.
- [32] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer Learning in Collaborative Filtering for Sparsity Reduction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, no. 1, jul 2010.
- [33] D. Agarwal, B.-C. Chen, and B. Long, "Localized factor models for multi-context recommendation," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2011, p. 609–617.
- [34] J. Lian, F. Zhang, X. Xie, and G. Sun, "Cccfnet: A content-boosted collaborative filtering neural network for cross domain recommender systems," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 817–818. [Online]. Available: <https://doi.org/10.1145/3041021.3054207>
- [35] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, "Cross-domain recommendation via cluster-level latent factor model," in *Machine Learning and Knowledge Discovery in Databases*, H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 161–176.
- [36] N. Mirbakhsh and C. X. Ling, "Improving top-n recommendation for cold-start users via cross-domain information," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 4, Jun. 2015. [Online]. Available: <https://doi.org/10.1145/2724720>
- [37] D. Rafailidis and F. Crestani, "A collaborative ranking model for cross-domain recommendations," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2263–2266. [Online]. Available: <https://doi.org/10.1145/3132847.3133107>
- [38] Y. Wang, C. Feng, C. Guo, Y. Chu, and J.-N. Hwang, "Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 717–725. [Online]. Available: <https://doi.org/10.1145/3289600.3290973>
- [39] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015, p. 278–288. [Online]. Available: <https://doi.org/10.1145/2736277.2741667>
- [40] F. Zhu, Y. Wang, C. Chen, G. Liu, and X. Zheng, "A graphical and attentional framework for dual-target cross-domain recommendation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI-20, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3001–3008, main track.
- [41] H. Kanagawa, H. Kobayashi, N. Shimizu, Y. Tagami, and T. Suzuki, "Cross-domain recommendation via deep domain adaptation," in *Advances in Information Retrieval*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham: Springer International Publishing, 2019, pp. 20–29.
- [42] C. Zhao, C. Li, and C. Fu, "Cross-domain recommendation via preference propagation graphnet," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2165–2168. [Online]. Available: <https://doi.org/10.1145/3357384.3358166>
- [43] G. Ma, Y. Wang, X. Zheng, X. Miao, and Q. Liang, "A trust-aware latent space mapping approach for cross-domain recommendation," *Neurocomputing*, vol. 431, pp. 100–110, mar 2021.
- [44] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 667–676. [Online]. Available: <https://doi.org/10.1145/3269206.3271684>
- [45] P. Li and A. Tuzhilin, "DDTCDDR: Deep dual transfer cross domain recommendation," *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 331–339, jan 2020.
- [46] C. Zhao, C. Li, R. Xiao, H. Deng, and A. Sun, "Catn: Cross-domain recommendation for cold-start users via aspect transfer network," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 229–238. [Online]. Available: <https://doi.org/10.1145/3397271.3401169>
- [47] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [48] P. V. Ca, L. T. Edu, I. Lajoie, Y. B. Ca, and P.-A. M. Ca, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pascal Vincent Hugo Larochelle Yoshua Bengio Pierre-Antoine Manzagol," Tech. Rep., 2010.
- [49] O. Kuchaiev and B. Ginsburg, "Training Deep AutoEncoders for Collaborative Filtering," *arXiv*, aug 2017.