A cross-domain recommender system using deep coupled autoencoders

ALEXANDROS GKILLAS* and DIMITRIOS KOSMOPOULOS*, Computer Engineering and Informatics Department, University of Patras, Greece

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 effectively 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. Additionally, a user-level relevance scenario is considered, where the two related domains contain common users. In light of these scenarios, 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 in the source and target domains, along with a coupled mapping function to model the non-linear relationships between these representations. 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 latent factors across domains. Extensive numerical experiments are conducted illustrating the superior performance of our proposed methods compared to several state-of-the-art cross-domain recommendation frameworks for the extreme cold start scenario.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Cross-domain recommendation systems, coupled autoencoders, latent factor models, deep learning.

ACM Reference Format:
Alexandros Gkillas and Dimitrios Kosmopoulos. 2025. A cross-domain recommender system using deep coupled autoencoders. In . ACM, New York, NY, USA, 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems are automated applications that suggest products to consumers based on their observed interests [12, 26, 33]. 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 [2]. The issues of cold start, sparsity and inclusion of new customers or products may compromise the performance of recommenders [30, 37, 47]. While these problems have been extensively studied from a single-domain viewpoint, cross-domain recommender systems (CDRS) bring a different perspective to their solution [7].

A core challenge in cross-domain recommendation (CDR) is to improve recommendations in a target domain—often suffering from data sparsity—by leveraging information from a related, data-rich source domain [5, 20, 29, 36, 39]. This challenge becomes especially acute in the *extreme cold-start scenario*, where no interactions are observed for certain users or items in the target domain. In this setting, traditional recommendation algorithms fail, and effective

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Association for Computing Machinery.

Manuscript submitted to ACM

cross-domain knowledge transfer is essential. The extreme cold-start scenario remains one of the most demanding and practically relevant cases for cross-domain recommendation, motivating a growing body of research [4, 8–10, 13, 40].

The CDR problem has been explored under different perspectives and scenarios, aiming to address challenging recommendation issues such as data sparsity. Data sparsity occurs when users interact with only a small fraction of items, resulting in a sparse user-item interaction matrix. This sparsity severely limits the ability of recommender systems to make accurate predictions because there is insufficient data to identify patterns and correlations between users and items [30]. In general, CDR methods can be divided into three major categories, that is content-based frameworks, embedding-based frameworks and rating pattern-based approaches [52]. *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 [8, 22, 40, 50, 51]. 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 transfers it across domains through domain adaptation techniques such as neural networks [21], [27] and transfer learning [28, 34, 46]. Finally, *rating pattern-based* approaches aim to transfer information such as rating patterns from the source to the target domain [16, 24, 41].

In this study, we focus on embedding-based methods because they are designed to directly transfer knowledge across domains by leveraging shared users or items, *without the need for additional content features*—which may not always be available or reliable in real-world scenarios. Embedding-based approaches, and in particular those using autoencoders, offer a compelling balance between competitive performance and computational efficiency. By learning low-dimensional latent representations that capture the underlying structure of user-item interactions, these models have proven highly effective at mitigating the sparsity problem and improving recommendation accuracy [42]. However, **current embedding-based methods have some limitations**. A major shortcoming is that most approaches train the autoencoders for the source and target domains *independently* [27, 40, 49, 54]. This separate training fails to capture the crucial dependencies and shared patterns that exist across domains. For instance, shared users or items often exhibit correlated behaviors in different domains; independent training cannot exploit these signals, resulting in latent representations that are only locally optimal and not aligned across domains. Consequently, cross-domain information is underutilized, leading to suboptimal recommendation accuracy. Furthermore, in these independent schemes, the mapping function used to transfer knowledge from source to target is also learned separately, after the autoencoders have been fixed. This *two-stage pipeline* can lead to significant information loss about the relationships among shared users or items, and forces subsequent works to adopt complex mapping mechanisms or adversarial training to compensate [17, 54]. Such designs not only increase model and training complexity, but also add to the computational burden—**an issue our framework aims to avoid by enabling joint learning and alignment of representations during training**.

To address these limitations, we introduce a family of coupled autoencoder frameworks for cross-domain recommendation. At the heart of our approach is the idea of capturing and aligning the intrinsic representations of shared entities (users or items) across domains, so as to facilitate efficient and effective knowledge transfer. Specifically, we propose two methods: we first establish a foundation based on coupled learning and joint optimization of autoencoders, enabling the direct transfer of knowledge through shared entities across domains. Building upon this, our second and more advanced method introduces a novel hybrid prediction strategy that goes beyond coupling alone, allowing each recommendation in the target domain to be informed not only by global patterns transferred from the source domain, but also by local, target-specific latent factors. This dual integration empowers the model to robustly adapt to both transferable and domain-specific information.

We investigate two recommendation scenarios based on the categorization outlined in [53]: a cross-system recommendation (CSR) task and a cross-domain recommendation (CDR) task. In the first task, we examine an **item-level relevance** recommendation scenario. This assumes that the source and target domains contain common items (item full overlap) but different users (user non-overlap) (see Figure 1). The second task considers a **user-level relevance** scenario where the two related domains have common users (user full overlap) but different items (item non-overlap) (see Figure 2). Note that, the proposed approaches are adaptable between CSR and CDR scenarios. The key difference lies in the 'bridge' element: for CSR, we use common items, while for CDR, we use common users. This interchangeability allows CDR techniques to be applied to CSR scenarios and vice versa.

Our main contributions are as follows:

- **Coupled Autoencoder Cross-Domain Recommendation (CACDR)**: We propose CACDR, a deep model that learns a pair of coupled autoencoders for the source and target domains. When users or items are shared, the source-domain encoder extracts latent representations which are then mapped via a neural network to the target domain, where the decoder generates predictions. The autoencoders and the mapping function are optimized jointly, allowing cross-domain alignment to emerge during training. This setup enables the full latent representation of shared entities to be transferred, facilitating effective recommendations even when the target domain is sparse.
- **Latent Factor Autoencoder Cross-Domain Recommendation (LFACDR):** LFACDR is formulated as a novel joint optimization problem that simultaneously trains two coupled autoencoders—one per domain—along with a mapping function that aligns their latent spaces. Each autoencoder reconstructs the user-item rating matrix of its respective domain and produces latent representations (factors) for both users and items. The model introduces coupling by enforcing that the latent representations of the shared entity (either users or items) across domains remain consistent. This is achieved via a mapping network and a consistency loss that penalizes misalignment between the transferred source representation and the native target one. Target domain predictions are then computed by combining the transferred latent factors of the shared entity (via the mapping network) with latent factors of the non-shared entity learned directly from the target domain. For example, in a user-overlap setting, LFACDR transfers user embeddings from the source and combines them with target-learned item embeddings to generate predictions. This design allows LFACDR to effectively integrate global knowledge from the source domain with local information from the target domain, capturing both transferable and domain-specific patterns. The entire model—autoencoders, mapping function, and decoders—is trained jointly, enabling robust knowledge transfer and improved performance, especially under data-scarce or extreme cold-start conditions.
- **Coupled optimization and cross-domain alignment mechanism**: In both methods, the coupling is realized via a loss term that encourages proximity between the source representation and its mapped version in the target space. However, the coupling operates differently: in CACDR, only the shared entity (users or items) is encoded, mapped, and decoded across domains; in LFACDR, both users and items are encoded in both domains, but only the shared side is transferred via the mapping function. The target prediction then relies on fusing the mapped (shared) latent factor with the target-specific factor. All components are optimized together to reduce

reconstruction and coupling losses, yielding a more principled and effective alignment of latent spaces than traditional two-stage or decoupled methods.

- We evaluate both models under two CDR scenarios—user-level and item-level relevance—with an emphasis on the extreme cold-start case, where all users (or items) are shared between domains but have only limited interactions in the target domain. Our results demonstrate that both CACDR and LFACDR outperform many baseline methods, with LFACDR offering superior accuracy due to its hybrid handling of transferred and local features.

The remainder of the paper is organized as follows. Section 2 provides an overview of related research. Section 3 formally states the problem formulation. Section 4 presents and analyzes the proposed methods. Section 5 validates the effectiveness of the proposed approaches through experiments on two public datasets used as source and target domains. Section 6 reports an ablation study examining the important components of the proposed methods. Finally, Section 7 summarizes our contributions and outlines directions for future research.

2 RELATED WORKS

In literature, there is a 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 [4], [52]. As we mentioned before, the proposed approaches belong to the category of embedding-based methods. In general, these approaches primarily focus on transferring information across domains using matrix factorization and transfer learning techniques based on shallow learning and deep learning methodologies.

Shallow Learning: Focusing on the shallow learning methods, Singh et al. [35] used a matrix factorization approach to transfer information across domains by sharing the user latent factor. Similarly, 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. However, despite their efficiency, these methods can only extract relatively shallow and linear attributes from the datasets in comparison to deep learning models which are able to capture more intricate and non-linear aspects from the collaborative interactions between users and items.

Deep Learning: In recent years, various deep learning approaches have been proposed [7, 9–11, 14, 17, 19, 25, 27, 38, 41, 43, 44, 48, 51]. To analyze the limitations and gaps of existing literature, we provide a categorization of the deep learning methodologies, which allows us to clearly situate our proposed framework within the current research landscape.

Embedding Mapping-Based Methods: This broad class of methods focuses on transferring user and/or item representations across domains by learning mapping functions, performing adversarial alignment, or leveraging graph neural networks and attention, but **without explicitly partitioning the latent space into domain-shared and domain-specific subspaces**. For example, **EMCDR** [27] learns latent user factors for each domain and applies a multi-layer perceptron to map users' representations from the source to the target domain, under the user-overlap assumption. **PTUPCDR** [54] extends this framework by employing a meta-learning strategy to generate a personalized bridge mapping for each user: it summarizes a user's source-domain interaction history via attention, then dynamically produces bridge parameters for adaptive transfer, optimizing the mapping for recommendation accuracy. **DML** [19] introduces a global orthogonal mapping to align independently learned user embeddings between domains, supporting bidirectional transfer via linear projection. **ACDR** [17] adopts an adversarial learning framework, mapping global user representations into domain-specific spaces with a GAN-based generator and discriminator, while also learning domain-specific item embeddings through autoencoders. **DARec** [41] applies a domain-adversarial neural network to extract domain-invariant user embeddings, training the encoder to make user representations indistinguishable between source and target. Other representative works include graph and attention-based models that transfer information through interaction graphs or GNNs [43, 48, 51], and a range of hybrid architectures that integrate side information or additional transfer learning strategies [9–11, 14, 25, 41, 44]. Although these methods have achieved strong empirical results—especially in cold-start and sparse data regimes—they generally rely on a two-stage pipeline, where embeddings are first learned independently and only then mapped or aligned for transfer. As a result, the source and target latent spaces may not be optimally co-adapted, and the mapping functions may struggle to fully capture nuanced relationships between domains, especially in highly heterogeneous settings. Moreover, without hybridizing transferred and target-specific features at prediction time, these approaches may under-utilize local context or fail to adapt to domain-specific patterns.

Disentanglement-Based Methods. A second major direction in cross-domain recommendation explicitly seeks to separate transferable and domain-specific information within the latent representation space, or to align distributions globally rather than by individual mappings. *Disentanglement-based methods* enforce this separation by decomposing user or item embeddings into domain-shared and domain-specific components, often with mutual information regularization, orthogonality constraints, or adversarial objectives. For example, **DisenCDR** [6] utilizes a variational bipartite graph encoder within a VAE framework, producing three embeddings for each user—one domain-shared and two domain-specific—and applies mutual information penalties to enforce both exclusivity and informativeness of the shared component. **MADD** [45] disentangles user embeddings using domain-adversarial and orthogonal losses, constructing personalized user representations via multi-level attention. Li et al. [23] proposed DisAlign, a framework for Cross-Domain Cold-Start Recommendation (CDCSR) that leveraged information from a 'warm' source domain to improve recommendation performance in a 'cold' target domain. DisAlign utilized rating and auxiliary representations and introduced Stein path alignment to efficiently align latent embedding distributions across domains. While these methods provide improved theoretical guarantees against negative transfer and can yield more interpretable representations, they often introduce additional modeling complexity (e.g., multiple encoders, regularization tuning), may risk discarding informative but non-shared signals, and in many cases base target domain prediction solely on the transferred or shared representation, without hybridizing with domain-specific or locally learned factors.

Among the various categories of cross-domain recommendation, our proposed methods are most closely related to the family of embedding mapping-based approaches, which seek to transfer latent representations across domains via learned mapping functions. Methods in this category, such as PTUPCDR [54], ACDR-GAN [17], DML [19], and DARec [41], have driven much of the recent progress in the field. However, they exhibit fundamental structural limitations that our approaches are designed to overcome. **First, these methods invariably adopt a two-stage training pipeline:** user and item embeddings for source and target domains are learned independently—typically via autoencoders or matrix factorization—with no cross-domain interaction or co-adaptation during embedding learning. The transfer of knowledge is then handled exclusively through a mapping function, which is trained after the embeddings are fixed; this mapping may be linear (DML), meta-learned (PTUPCDR), or adversarial (ACDR-GAN), but in all cases, it is introduced post-hoc. *This separation leads to latent spaces that are not mutually aligned or optimized for cross-domain transfer, making it challenging to discover optimal transformations and reducing the effectiveness of knowledge transfer, especially under data sparsity or cold-start conditions.* **Second, the complexity of the mapping function.** While sophisticated objectives such as adversarial training (ACDR-GAN) or personalized bridges (PTUPCDR) are designed to increase flexibility, they often introduce instability, require extensive hyperparameter tuning, and add to the overall training complexity. Conversely, simpler mappings (e.g., the orthogonal transformation in DML) can be too rigid to capture the nonlinear relationships often required between distinct domains, especially when representations have been trained independently. Furthermore, these methods typically do not structurally distinguish between domain-shared and domain-specific representations, a limitation that has only recently begun to be addressed by disentanglement-based models.

Disentanglement-Based and Distribution Alignment Methods. A parallel line of research in CDR explicitly seeks to decompose the latent space into transferable and domain-specific subspaces, often using information-theoretic, orthogonality, or adversarial regularization. **DisenCDR** [6] utilizes a variational bipartite graph encoder within a VAE framework, producing three embeddings for each user—one domain-shared and two domain-specific—and applies mutual information penalties to enforce both exclusivity and informativeness of the shared component. **MADD** [45] disentangles user embeddings using domain-adversarial and orthogonal losses, constructing personalized user representations via multi-level attention. **DisAlign** [23] instead focuses on aligning the overall distribution of latent representations between domains using a Stein path approach, thereby facilitating transfer under extreme cold-start conditions. These methods provide improved theoretical guarantees against negative transfer and can yield more interpretable representations, especially when the domains are similar and the assumption of common underlying information is true. However, they often introduce additional modeling complexity (e.g., multiple encoders, regularization tuning), may risk discarding informative but non-shared signals, and in several cases base target domain prediction solely on the transferred or shared representation, without hybridizing with domain-specific or locally learned factors.

Addressing These Limitations: Our proposed frameworks directly address the core limitations observed in embedding mapping-based methods. Unlike most prior art, both of our models ensure that latent spaces are co-adapted for transfer during the training process itself, rather than relying on after-the-fact mapping. By jointly training the autoencoders for the source and target domains in a coupled manner, our models capture both the intrinsic structures within each domain and the critical cross-domain dependencies. This approach eliminates the inefficiencies of independent training, allowing the learned latent representations to be naturally aligned during the learning process. As a result, the mapping function that transfers information between domains can be much simpler—typically implemented as a lightweight MLP—without requiring the complex or unstable architectures (such as adversarial or meta-learned mappings) used in previous work. In this way, our models achieve effective and robust knowledge transfer, even in highly sparse or cold-start scenarios, while maintaining low computational and implementation complexity.

The key distinction between our two proposed methods lies in the handling of domain-specific knowledge at prediction time. The first method, CACDR, focuses on transferring the full representation of the shared entity (user or item) from the source to the target domain and uses it directly for prediction. In contrast, our second method, LFACDR, introduces a novel hybrid integration mechanism: for each recommendation in the target domain, LFACDR explicitly fuses the transferred latent factor of the shared entity (via the mapping network) with a latent factor for the non-shared entity that is learned directly from the target domain. This design enables LFACDR to flexibly leverage both global knowledge transferred from the source and local, domain-specific patterns unique to the target, providing a more contextually adaptive and accurate solution. This hybrid strategy stands in contrast to both prior embedding-mapping approaches (which ignore local adaptation) and disentanglement-based methods (which enforce a strict separation), offering a more practical balance. In practical terms, this means that our LFACDR method can utilize both the strengths

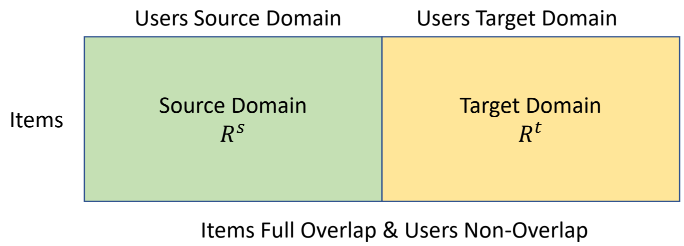


Fig. 1. The item-level relevance CSR task. We assume two related domains, which contain the same items (item full overlap) corresponding to different users (user non-overlap).

of transferred global knowledge and the fine-grained details unique to the target domain, a balance that prior methods struggle to achieve.

3 NOTATIONS AND PROBLEM FORMULATION

Table 1 summarizes all the required notations of this study. In the literature the two domains are often referred as the 'source' and 'target' domains. Without loss of generality, let $\boldsymbol{R}^s \in \mathbb{R}^{m \times n}$ and $\boldsymbol{R}^t \in \mathbb{R}^{m \times n}$ be the rating matrices representing the ratings between m items and n users for the source and the target domain respectively, where $\boldsymbol{R}^s(i, j)$ represents the rating of the user j for the item i in the source domain and $\boldsymbol{R}^t(i, j)$ represents the corresponding rating in the target domain. Furthermore, we denote as $\boldsymbol{M}^s = [[\boldsymbol{m}_1^s]^\top; [\boldsymbol{m}_2^s]^\top; ...] = \boldsymbol{R}^s$, $\boldsymbol{M}^t = [[\boldsymbol{m}_1^t]^\top; [\boldsymbol{m}_2^t]^\top; ...] = \boldsymbol{R}^t$ the item rating matrix of the source and target domain and $\boldsymbol{U}^s = [[\boldsymbol{u}_1^s]^\top; [\boldsymbol{u}_2^s]^\top; ...] = \boldsymbol{R}^{s\top}$, $\boldsymbol{U}^t = [[\boldsymbol{u}_1^t]^\top; [\boldsymbol{u}_2^t]^\top; ...] = \boldsymbol{R}^{t\top}$ the user rating matrix of the source and the target domain, respectively. In general, the item rating vector $\boldsymbol{m}_i^s \in \mathbb{R}^{1 \times n}$ describes the rating relationship between the item i and all the users in the source domain, whereas the user rating vectors $\boldsymbol{u}_i^s \in \mathbb{R}^{1 \times m}$ describes the rating relationship between the user i and all the items of the source domain. Accordingly, the item and user rating vectors $\boldsymbol{m}_i^t$, $\boldsymbol{u}_i^t$ represent the corresponding rating relations of the target domain.

Taking into consideration that both domains share the same items or users, 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 (or users) with no ratings or little information, thus tackling the data sparsity and the cold-start problem in target domain. In more detail, this task can be seen as a domain adaptation procedure (transfer learning) [31], which aims to describe the unknown mathematical relationships between the source and target domains. Nonetheless, by 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.

4 PROPOSED MODELS

In this section, we derive two coupled autoencoder frameworks that can be used for the CDR problem. Specifically, 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 extension 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 modeling domain-specific information in the source domain, both methods transfer that information to the target domain using a deep learning network. This transfer

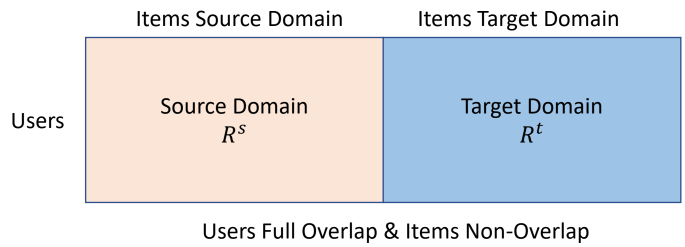


Fig. 2. The user-level relevance CDR task. We assume two related domains, which contain the same users (user full overlap) corresponding to different items (item non-overlap).

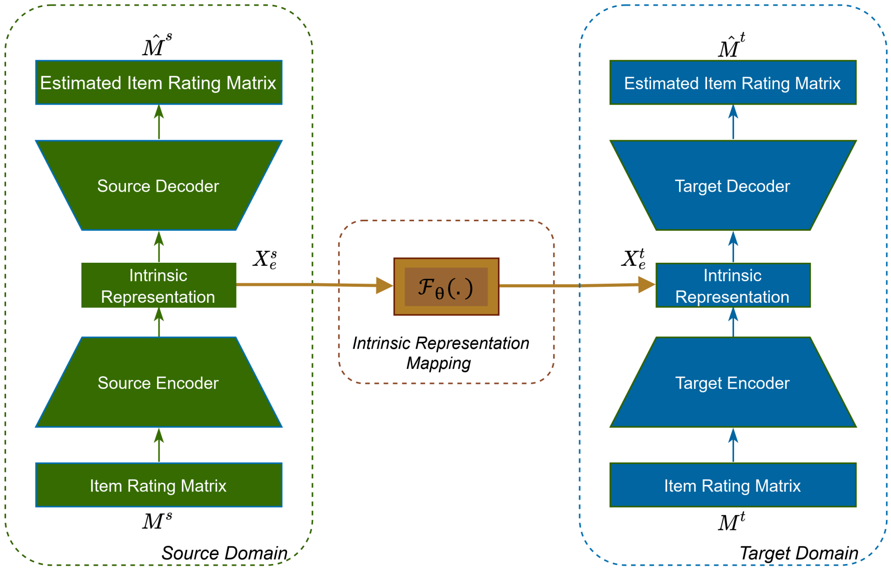
Table 1. Mathematical Notations

Symbol	Description
	Source Domain
n	The number of users of the source and target domain
m	The number of items of the source and target domain
k	The dimension of the autoencoder hidden intrinsic representation
$\boldsymbol{R}^s \in \mathbb{R}^{m \times n}$	The rating matrix of the source domain
$\hat{\boldsymbol{R}}^s \in \mathbb{R}^{m \times n}$	The predicted rating matrix of the source domain
$\boldsymbol{U}^s \in \mathbb{R}^{n \times m}$	The user rating matrix of the source domain
$\boldsymbol{M}^s \in \mathbb{R}^{m \times n}$	The item rating matrix of the source domain
$\boldsymbol{Y}_e^s \in \mathbb{R}^{n \times k}$	The output of the encoder of $\boldsymbol{U}^s$
$\boldsymbol{X}_e^s \in \mathbb{R}^{m \times k}$	The output of the encoder of $\boldsymbol{M}^s$
$\hat{\boldsymbol{U}}^s \in \mathbb{R}^{n \times m}$	The output of the decoder of $\boldsymbol{Y_e}^s$
$\hat{\boldsymbol{M}}^s \in \mathbb{R}^{m \times n}$	The output of the decoder of $\boldsymbol{X_e}^s$
	Target Domain
$\boldsymbol{R}^t \in \mathbb{R}^{m \times n}$	The rating matrix of the target domain
$\hat{\boldsymbol{R}}^t \in \mathbb{R}^{m \times n}$	The predicted rating matrix of the target domain
$\boldsymbol{U}^t \in \mathbb{R}^{n \times m}$	The user rating matrix of the target domain
$\boldsymbol{M}^t \in \mathbb{R}^{m \times n}$	The item rating matrix of the target domain
$\boldsymbol{Y}_e^t \in \mathbb{R}^{n \times k}$	The output of the encoder of $\boldsymbol{U}^t$
$\boldsymbol{X}_e^t \in \mathbb{R}^{m \times k}$	The output of the encoder of $\boldsymbol{M}^t$
$\hat{\boldsymbol{U}}^t \in \mathbb{R}^{n \times m}$	The output of the decoder of $\boldsymbol{Y_e}^t$
$\hat{\boldsymbol{M}}^t \in \mathbb{R}^{m \times n}$	The output of the decoder of $\boldsymbol{X_e}^t$

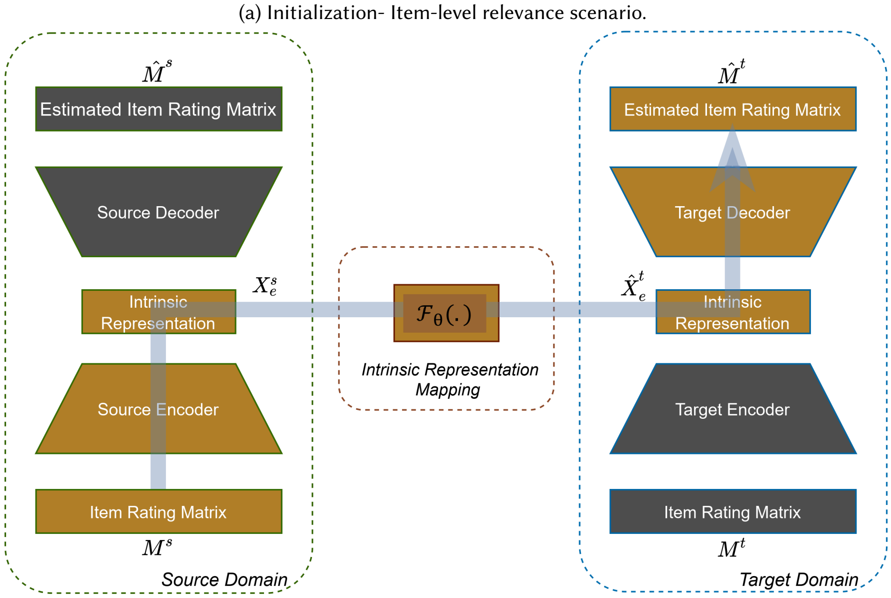
mechanism can be implemented through either a simpler architecture e.g., a Multilayer Perceptron (MLP) or a more complex one such as a Generative Adversarial Network (GAN).

4.1 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 [3]. 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



(a) Initialization- Item-level relevance scenario.



(b) Coupled Learning - Item-level relevance scenario.

Fig. 3. 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 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 is employed in order to jointly optimize all the active parts of the autoencoders i.e., the source encoder, the mapping network and the target decoder) involved in the rating prediction in target domain (stage 3). **Note that we follow similar procedure for the user-level relevance task using the corresponding User rating matrices in source and target domain**

hand, the coupled autoencoder model is able to capture these internal relationships and derive better representations 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 that learn separately the intrinsic hidden representations of the item rating matrices (item-level relevance scenario) or user rating matrices (user-level relevance scenario) of the source and target domain, respectively. The second stage uses a neural network 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 function are trained independently. Thus, 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 the other during the training procedure. Finally, recommendations can be made for a new item or user in the target domain based on the intrinsic representations of the same item or user in the source domain. The complete proposed methodology is depicted in Fig. 3.

4.1.1 *Coupled Autoencoder-Based*.

Focusing on the item-level relevance scenario, we consider the item rating matrices of the source and target domains denoted as $\boldsymbol{M}^s$, $\boldsymbol{M}^t$ respectively. Note that for the user-level relevance scenario, we employ the user rating matrices in the source and target domain denoted as $\boldsymbol{U}^s$, $\boldsymbol{U}^t$. Next, we describe the method only for the item-level relevance task, since the methodology is identical for the user-level relevance scenario as well. The corresponding source and target autoencoders, which learn the hidden intrinsic representations of the two matrices, can be obtained by minimizing the following loss functions:

$$\begin{aligned} \mathcal{L}^s_{autoenc} &= \left\| \boldsymbol{M}^s - \hat{\boldsymbol{M}}^s \right\|_F^2 = \left\| \boldsymbol{M}^s - \mathcal{D}^s(\mathcal{E}^s(\boldsymbol{M}^s)) \right\|_F^2 \\ \mathcal{L}^t_{autoenc} &= \left\| \boldsymbol{M}^t - \hat{\boldsymbol{M}}^t \right\|_F^2 = \left\| \boldsymbol{M}^t - \mathcal{D}^t(\mathcal{E}^t(\boldsymbol{M}^t)) \right\|_F^2, \end{aligned} \tag{1}$$

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

Encoding process: The intrinsic representation of the source item rating matrix $\boldsymbol{M}^s \in \mathbb{R}^{m \times n}$ is given by

$$\boldsymbol{X}^s_e = \mathcal{E}^s(\boldsymbol{M}^s)$$

or equivalently,

$$\begin{aligned} \boldsymbol{X}^s_{e,1} &= \varphi(\boldsymbol{W}^s_{e,1}\boldsymbol{M}^s + b^s_{e,1}) \\ \boldsymbol{X}^s_{e,2} &= \varphi(\boldsymbol{W}^s_{e,2}\boldsymbol{X}^s_{e,1} + b^s_{e,2}) \\ &\cdots \\ \boldsymbol{X}^s_{e} &= \varphi(\boldsymbol{W}^s_{e,L}\boldsymbol{X}^s_{e,L-1} + b^s_{e,L}), \end{aligned} \tag{2}$$

where $\boldsymbol{W}^s_{e,i}$, $b^s_{e,i}$ $(i = 1, \ldots, L)$ denote the weight matrices and the bias terms for the encoding layers of the source autoencoder, $\varphi(.)$ is the activation function ReLU, L stands for the number of hidden layers, $\boldsymbol{X}^s_e \in \mathbb{R}^{m \times k}$ is the output of the source encoder $\mathcal{E}^s(.)$ and $k \ll n$.

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

$$\boldsymbol{X}^t_e = \mathcal{E}^t(\boldsymbol{M}^t)$$

or equivalently,

$$\begin{aligned} X^t_{e,1} &= \varphi(W^t_{e,1} M^t + b^t_{e,1}) \\ X^t_{e,2} &= \varphi(W^t_{e,2} X^t_{e,1} + b^t_{e,2}) \\ &\dots \\ X^t_e &= \varphi(W^t_{e,L} X^t_{e,L-1} + b^t_{e,L}), \end{aligned} \tag{3}$$

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

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

$$\hat{M}^s = \mathcal{D}^s(X^s_e) \tag{4}$$

$$\hat{M}^t = \mathcal{D}^t(X^t_e), \tag{5}$$

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

4.1.2 Non Linear Mapping. After obtaining the intrinsic representations of the item (or user) rating matrices, a deep learning network is employed to capture and model the underlying relationship between the intrinsic representations of the source and target domain (X^s_e, X^t_e), thus transferring the appropriate knowledge from the source to target domain. This transfer mechanism can be implemented through either a simpler architecture like a Multilayer Perceptron (MLP) or a more complex one such as a Generative Adversarial Network (GAN). Mathematically, the non-linear mapping function $\mathcal{F}_\theta(.)$ can be written as

$$\hat{X^t_e} = \mathcal{F}_\theta(X^s_e) \tag{6}$$

where θ denote the weight parameters of the mapping network. Thus, the parameters of the mapping network can be learned minimizing the following loss function

$$\mathcal{L}_{mlp} = \left\| X^t_e - \hat{X^t_e} \right\|^2_F = \left\| X^t_e - \mathcal{F}_\theta(X^s_e) \right\|^2_F . \tag{7}$$

4.1.3 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 (item-level relevance scenario) or new users in target domain using information of the same users in source domain (user-level relevance scenario). In particular, given an item or user j in the target domain, the following methodology is used to recover its predicted rating:

1. The same item or user 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/user in the target domain can be estimated via the intrinsic representation of the item/or user in the source domain and the mapping network based on (6).
3. Finally, the predicted rating of the item/user in the target domain is recovered based on target autoencoder (decoding procedure) according to (5).

4.1.4 Coupled 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 via relation (1) and then the mapping function (based on the estimated intrinsic representations) via relation (6) 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 rating matrix) and the target domain (i.e., target rating matrix). This procedure is piecemeal and thus sub-optimal, since there is no influence from one to the other 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 (i.e., the item rating matrix, $\boldsymbol{M}^t = \boldsymbol{R}^t$ in the item-level relevance scenario or the user matrix, $\boldsymbol{U}^t$ in the user-level relevance scenario), the proposed objective function for optimizing jointly the two autoencoders (source and target) along with the mapping network may be written as:

$$\left\|\boldsymbol{M}^t - \hat{\boldsymbol{M}}^t\right\|_F^2 \xrightarrow{(5)} \left\|\boldsymbol{M}^t - \mathcal{D}^t(\hat{\boldsymbol{X}_e^t})\right\|_F^2 \xrightarrow{(6)}$$
$$\left\|\boldsymbol{M}^t - \mathcal{D}^t(\mathcal{F}_\theta(\boldsymbol{X}_e^s)\right\|_F^2 . \tag{8}$$

Thus, using equation (2), the corresponding loss function that we seek to minimize during the coupled learning is given by

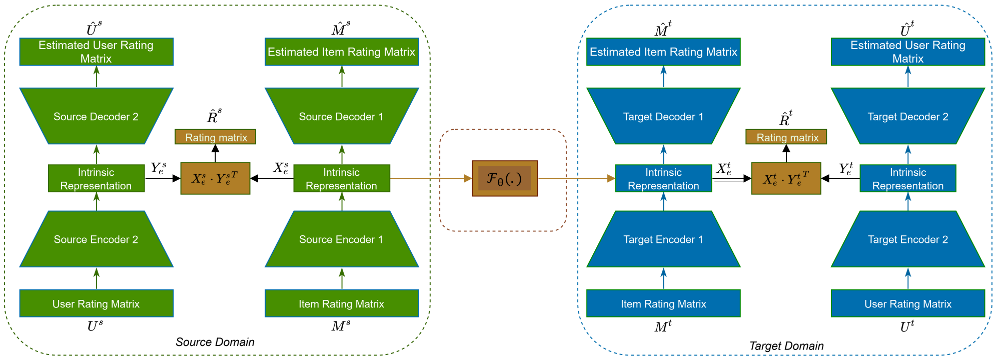
$$\mathcal{L}_{coupled\_learning} = \left\|\boldsymbol{M}^t - \mathcal{D}^t(\mathcal{F}_\theta(\mathcal{E}^s(\boldsymbol{M}^s)\right\|_F^2 . \tag{9}$$

Note that now in relation (9) the source encoder $\mathcal{E}^s(.)$, the mapping neural network $\mathcal{F}_\theta(.)$ and the target decoder $\mathcal{D}^t(.)$ are all explicitly involved in the reconstruction of the desired output $\hat{\boldsymbol{M}}^t$. *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. 3b illustrates the proposed coupled architecture. Having obtained, the stacked network architecture the back-propagation algorithm is used to optimize (9). Algorithm 1 summarizes the proposed methodology.

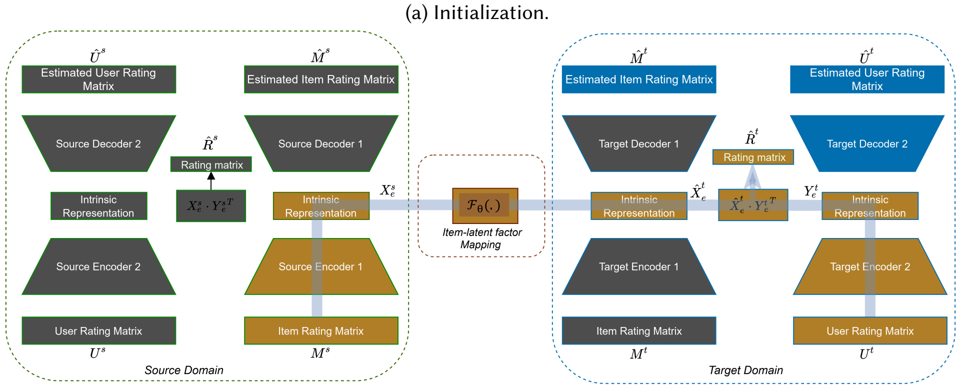
Algorithm 1 : CACDR learning method

Require: The item rating matrices of the source and target domain $\boldsymbol{M}^s \in \mathbb{R}^{m\times n}$, $\boldsymbol{M}^t \in \mathbb{R}^{m\times n}$ for the item-level relevance scenario, or the item user rating matrices $\boldsymbol{U}^s \in \mathbb{R}^{n\times m}$, $\boldsymbol{U}^t \in \mathbb{R}^{n\times m}$ for the user-level relevance scenario
Ensure: The predicted rating matrix of the target domain $\hat{\boldsymbol{R}}^t$

{***Stage A**: Initialization*}
1: Initialize the source domain autoencoder by learning the intrinsic representation $\boldsymbol{X}_e^s \in \mathbb{R}^{m\times k}$ of the matrix $\boldsymbol{M}^s$(item-level relevance scenario) or $\boldsymbol{U}^s$ (user-level relevance scenario) via (2).
2: Initialize the target domain autoencoder by learning the intrinsic representation $\boldsymbol{X}_e^t \in \mathbb{R}^{m\times k}$ of the matrix $\boldsymbol{M}^t$(item-level relevance scenario) or $\boldsymbol{U}^t$ (user-level relevance scenario) via (3).
3: Initialize the mapping network by learning the mapping function from $\boldsymbol{X}_e^s$ to $\boldsymbol{X}_e^t$ via (6).
{***Stage B**: Coupled Learning*}
4: Construct the CACDR model via (9).



(a) Initialization.



(b) Coupled Learning - Item-level relevance scenario.

Fig. 4. 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 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 (i.e., the Source Encoder 1, the mapping network and the Target Encoder 2) involved in the rating prediction in target domain (stage 3). Note that we follow similar procedure for the user-level relevance task using the corresponding User rating matrices in source and target domain.

4.2 The LFACDR Method

As previously mentioned, the autoencoders constitute an ideal mathematical tool to reveal and learn complex low dimensional 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. 4.

4.2.1 Latent Factor Modeling based on Autoencoders. Consider the user rating matrix $\boldsymbol{U}^s \in \mathbb{R}^{n\times m}$, the item rating matrix $\boldsymbol{M}^s \in \mathbb{R}^{m\times n}$ and the rating matrix $\boldsymbol{R}^s \in \mathbb{R}^{n\times m}$ 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 optimization problem is formulated as

$$\begin{aligned}\mathcal{L}^s = &\left\|\boldsymbol{M}^s - \mathcal{D}^s_m(\boldsymbol{X}^s_{\boldsymbol{e}})\right\|^2_F + \left\|\boldsymbol{U}^s - \mathcal{D}^s_u(\boldsymbol{Y}^s_{\boldsymbol{e}})\right\|^2_F + \\ &\left\|\boldsymbol{Y}^s_{\boldsymbol{e}} - \mathcal{E}^s_u(\boldsymbol{U}^s)\right\|^2_F \quad + \left\|\boldsymbol{X}^s_{\boldsymbol{e}} - \mathcal{E}^s_m(\boldsymbol{M}^s)\right\|^2_F + \\ &\lambda \left\|\boldsymbol{X}^s_{\boldsymbol{e}} \boldsymbol{Y}^{sT}_{\boldsymbol{e}} - \boldsymbol{R}^s\right\|^2_F,\end{aligned} \tag{10}$$

where $\mathcal{D}^s_m(.)$, $\mathcal{E}^s_m(.)$ denote the decoding and encoding procedure of the first source autoencoder (items), $\boldsymbol{X}^{\boldsymbol{e}}_{\boldsymbol{s}} \in \mathbb{R}^{m\times k}$ is the item-latent factor matrix derived from the output of the first encoder, $\mathcal{D}^s_u(.)$, $\mathcal{E}^s_u(.)$ stand for the decoding and encoding procedure of the second source autoencoder (users) and $\boldsymbol{Y}^{\boldsymbol{e}}_{\boldsymbol{s}} \in \mathbb{R}^{n\times k}$ denotes the user-latent factor matrix derived from the output of the second encoder. Note 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 $\boldsymbol{X}^s_e$ and $\boldsymbol{Y}^s_e$ 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 $\boldsymbol{X}^t_e$ and $\boldsymbol{Y}^t_e$ of the target domain.

4.2.2 *Non Linear Mapping.*

Having acquired the latent factor matrices $\{\boldsymbol{X}^e_s, \boldsymbol{Y}^e_s, \boldsymbol{X}^e_t, \boldsymbol{Y}^e_t\}$ of items and users in the source and target domain, similar to the previous proposed method, a neural network such as a simple MLP or a more complex network is again used to learn the mapping function between the item latent factor matrices of the source and target domain $(\boldsymbol{X}^s_e, \boldsymbol{X}^t_e)$ for the item-level relevance scenario or the the user latent factor matrices of the source and target domain $(\boldsymbol{Y}^s_e, \boldsymbol{Y}^t_e)$ for the user-level relevance scenario. Concerning, the item-level relevance scenario, this can be expressed as follows,

$$\hat{\boldsymbol{X}^t_e} \;= \mathcal{F}_\theta(\boldsymbol{X}^s_e) \tag{11}$$

Similarly, the parameters of the mapping network can be learned solving the following loss function

$$\mathcal{L}_{mlp} = \left\|\boldsymbol{X}^t_e - \hat{\boldsymbol{X}^t_e}\right\|^2_F = \left\|\boldsymbol{X}^t_e - \mathcal{F}_\theta(\boldsymbol{X}^s_e)\right\|^2_F. \tag{12}$$

Note that for the user-level relevance scenario, the mapping function is defined as follows

$$\hat{\boldsymbol{Y}^t_e} \;= \mathcal{F}_\theta(\boldsymbol{Y}^s_e) \tag{13}$$

4.2.3 *Rating prediction.*

In general, given a new item (item-level relevance task) or a new user (user-level relevance task) 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 or same user in the target domain via the mapping function (11). Concerning, the item-level relevance task the predicted rating between item i and user j in the target domain is given by the following relation,

$$\hat{\boldsymbol{R}}^t(i,j) = \hat{\boldsymbol{X}^t_e}(i,:)\,\boldsymbol{Y}^t_e(j,:)^T \tag{14}$$

where $\hat{\boldsymbol{X}^t_e}(i,:) \in \mathbb{R}^{1\times k}$ 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 (11) and $\boldsymbol{Y}^t_e(i,:) \in \mathbb{R}^{1\times k}$ stands of the row vector of matrix $\boldsymbol{Y}^t_e$, representing the user-latent factor of user j in the target domain.

Similarly, regarding the user-level relevance the predicted rating between item i and user j in the target domain is given by the following relation,

$$\hat{\boldsymbol{R}}^t(i,j) = \boldsymbol{X}_e^t(i,:)\,\hat{\boldsymbol{Y}}_e^t(j,:)^T \tag{15}$$

where $\hat{\boldsymbol{Y}}_e^t(j,:) \in \mathbb{R}^{1\times k}$ denote the estimated user-latent factor of user j in the target domain based on the corresponding user-latent factor of user j in the source domain via the mapping function in relation (11)

4.2.4 Coupled Learning. Similar to the previous procedure the training can be divided into two phases: the first phase is the initialization and the second phase is the coupled learning. *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.
Coupled learning - Item-level relevance scenario: Since the aim of the proposed method in this scenario 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 mapping network is given by

$$\left\|\boldsymbol{R}^t - \hat{\boldsymbol{R}}^t\right\|_F^2 \Rightarrow \left\|\boldsymbol{R}^t - \hat{\boldsymbol{X}}_e^t\,(\boldsymbol{Y}_e^t)^\top\right\|_F^2 \xRightarrow{(11)} \left\|\boldsymbol{R}^t - \mathcal{F}_\theta(\boldsymbol{X}_e^s)\,(\boldsymbol{Y}_e^t)^\top\right\|_F^2 \tag{16}$$

where $\boldsymbol{X}_e^s = \mathcal{E}_m^s(\boldsymbol{M}^s)$ and $\boldsymbol{Y}_e^t = \mathcal{E}_u^t(\boldsymbol{U}^t)$. Hence, we aim to minimize the following loss function during the coupled learning stage

$$\mathcal{L}_{coupled\_learning} = \left\|\boldsymbol{R}^t - \mathcal{F}_\theta(\boldsymbol{X}_e^s)\,(\boldsymbol{Y}_e^t)^\top\right\|_F^2 + \left\|\boldsymbol{X}_e^s - \mathcal{E}_m^s(\boldsymbol{M}^s)\right\|_F^2 + \left\|\boldsymbol{Y}_e^t - \mathcal{E}_u^t(\boldsymbol{U}^t)\right\|_F^2. \tag{17}$$

From relation (17) it is easy to verify that the source encoder for the items $\mathcal{E}_m^s(.)$, the target encoder for the users $\mathcal{E}_u^t(.)$ and mapping network are all explicitly involved in the reconstruction of the desired output $\hat{\boldsymbol{R}}^t$. Thus, the three network units can be coupled together by jointly optimizing them through the back propagation algorithm. *Coupled learning - User-level relevance scenario*: The aim of the proposed method in this scenario is to accurately predict the ratings of new users in the target domain, the objective function for jointly optimizing the autoencoders extracting the item and user-latent factor matrices along with the mapping network is given by

$$\left\|\boldsymbol{R}^t - \hat{\boldsymbol{R}}^t\right\|_F^2 \Rightarrow \left\|\boldsymbol{R}^t - \boldsymbol{X}_e^t\,(\hat{\boldsymbol{Y}}_e^t)^T\right\|_F^2 \xRightarrow{(11)} \left\|\boldsymbol{R}^t - \boldsymbol{X}_e^t\,(\mathcal{F}_\theta(\boldsymbol{Y}_e^s))^\top\right\|_F^2 \tag{18}$$

where $\boldsymbol{X}_e^t = \mathcal{E}_m^t(\boldsymbol{M}^t)$ and $\boldsymbol{Y}_e^s = \mathcal{E}_u^s(\boldsymbol{U}^s)$. Thus, the loss function during this stage is

$$\mathcal{L}_{coupled\_learning} = \left\|\boldsymbol{R}^t - \boldsymbol{X}_e^t\,(\mathcal{F}_\theta(\boldsymbol{Y}_e^s))^T\right\|_F^2 + \left\|\boldsymbol{X}_e^t - \mathcal{E}_m^t(\boldsymbol{M}^t)\right\|_F^2 + \left\|\boldsymbol{Y}_e^s - \mathcal{E}_u^s(\boldsymbol{U}^s)\right\|_F^2. \tag{19}$$

Fig. 4b demonstrates the proposed coupled framework. The overall methodology is summarized in Algorithm 2.

5 EXPERIMENTAL VALIDATION

To validate the efficacy and applicability of the two proposed methods, extensive experiments were conducted in the context of the extreme cold-start problem. In particular, this study examines two scenarios: the first one focuses on an item-level relevance Cross-System Recommendation scenario, where two domains (source and target) share common

Algorithm 2 : LFACDR learning method

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

Ensure: The predicted rating matrix of the target domain $\hat{\boldsymbol{R}}^t$

{***Stage A***: *Initialization*}

1: Initialize the two source domain autoencoders by learning the item-latent factor matrix $\boldsymbol{X}_e^s \in \mathbb{R}^{m \times k}$ and the user-latent factor matrix $\boldsymbol{Y}_e^s \in \mathbb{R}^{n \times k}$ via (10).

2: Initialize the two target domain autoencoders by learning the item-latent factor matrix $\boldsymbol{X}_e^t \in \mathbb{R}^{m \times k}$ and the user-latent factor matrix $\boldsymbol{Y}_e^t \in \mathbb{R}^{n \times k}$ via (10).

3: Initialize the mapping network by learning the mapping function from $\boldsymbol{X}_e^s$ to $\boldsymbol{X}_e^t$ for the the item-level relevance scenario via (11) or $\boldsymbol{Y}_e^s$ to $\boldsymbol{Y}_e^t$ for the user-level relevance scenario via (13) .

{***Stage B***: *Coupled Learning*}

4: Construct the LFACDR model via (17) for the item-level relevance scenario or (19) for the user-level relevance scenario.

items (e.g., movies) and contain different users; the second scenario examines a user-level relevance cross-domain recommendation scenario, where the source and target domain share common users and contain different items e.g., (the source domain may contain movies as items and the target domain may contain books as items) . Our goals are to demonstrate:

- that the proposed methods are capable to exploit and transfer valuable information from the source domain to improve the recommendation performance in the target domain, thus tackling effectively the sparsity problem, in which items (e.g., movies) or users in target domain have limited historic information.
- the effect of different sparsity levels of the rating matrices in the source and target domains.
- the performance of the methods in comparison to other state-of-the-art approaches.

5.1 Datasets

Item-level relevance CSR scenario - Datasets: Two publicly available benchmark datasets were employed to demonstrate the merits of the proposed recommendation frameworks. Namely, we used the MovieLens[1] and Netflix[2] datasets, which contain a large portion of same movies, thereby forming an item-level relevance scenario, as depicted in Figure 1. According to IMDB information more than 6000 movies are the same across MovieLens and Netflix datasets. Regarding user numbers, the Netflix dataset includes over 2, 000, 000 users, while the MovieLens dataset encompasses more than 150, 000 users.

User-level relevance CDR scenario - Datasets: Concerning the CDR scenario with the shared users, we employed the Douban dataset [27], where users provide ratings to different items such as movies, books and music, thus forming a natural user-level relevance scenario, as depicted in Figure 2.

5.2 Experimental Setup

Experimental Setting:

Regarding the **item-level relevance CSR scenario** for completeness purposes, first we took the MovieLens as source domain and Netflix as target domain and then we used the Netflix as source domain and the MovieLens as

[1] https://grouplens.org/datasets/movielens/
[2] https://www.kaggle.com/netflix-inc/netflix-prize-data

Table 2. Statistics of the four Data Subsets derived from the Movielens and Netflix datasets.

Data subsets	Domains	#Common Movies	#Users	Sparsity
No. 1	MovieLens	5819	40000	97.22%
	Netflix	5819	40000	94.80%
No. 2	MovieLens	5986	48575	98.17%
	Netflix	5986	49681	97.02%
No. 3	MovieLens	6000	30100	99.35%
	Netflix	6000	30040	98.99%
No. 4	MovieLens	4200	44830	99.99%
	Netflix	4200	43765	99.86%

Table 3. Statistics of the Douban Datasets.

Data subsets	Domains	#Common Users	#Items	Sparsity
No. 1	Douban Book	2120	8659	99.41%
	Douban Movie	2120	16584	98.20%
No. 2	Douban Music	1580	6888	99.45%
	Douban Book	1580	7545	99.50%

target domain. Taking into account the huge number of users in both domains (i.e., more than 2.000.000 users in Netflix and more than 150.000 users in MovieLens platform), we followed the same strategy as in [9, 49] and we randomly sub-sampled a certain number of users in the MovieLens and Netflix dataset, ensuring that the selected users had at least 5 interactions with the considered movies in the domain that belong. In order to examine the effect of different sparsity levels of the rating matrices, we repeated the above procedure four times, selecting randomly different users with varying number of interactions. Hence, we ended up with four data subsets (i.e., No 1, No 2, No.3, No. 4) where the sparsity level increase from data subset No. 1 to data subset No. 4. Table 2 provides the detailed statistics of the four resulting data subsets in which the number of the common movies, the number of the selected users and the sparsity of the rating matrices are listed.

Regarding the **user-level relevance CDR scenario**, first we took the book as source domain and the movie as target domain and then we employed the music as source domain and the book as target domain. The datasets are shown in Table 3.

Training and Testing Setting: *Item-level scenario*: In our extensive experiments the four data subsets presented in Table 2 were randomly divided into the training set (80%) and the testing set (20%). In more detail, the 80% of the common movies was used during the training phase and the rest 20% was used during the testing. Regarding the testing set, we removed all the rating information from the movies in the target domain. The testing movies from the source domain were employed in order to transfer valuable information to the target domain. 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 [18]. *User-level scenario*: Similar procedure was employed. 80% of the common users was used during the training phase and the rest 20% was used during testing.

Parameter Settings: The parameters of our proposed methods are determined to be ideal via exploration of the parameter space. We explicitly followed a rigorous validation process to determine the best hyperparameters. For the first random split, a small subset (approximately 10%) of the training data was reserved as a validation set. We conducted hyperparameter tuning based on performance on this validation subset. Once the optimal hyperparameters were selected, they were fixed and used across all subsequent randomized splits. In those splits, the model was trained on the entire 80% training set (without withholding validation data) and evaluated on the independent 20% test set.

For **CACDR** the sizes of the autoencoders layers were set to $\{256, 128, 64, 64, 128, 256\}$ 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 coupled learning stage the number of epochs was set to 300, the learning rate and regularization term was 10^{-5}.

For **LFACDR** the sizes of the autoencoders layers were set to $\{512, 256, 128, 128, 256, 512\}$ 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 coupled 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.

Mapping function: Regarding the Non Linear Mapping function, this transfer mechanism we employed a simple architecture based on a Multiplayer Perceptron (MLP). The sizes of the MLP layers are set to 64, 128.

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 [15], since the zero values should be ignored during the training stage of the proposed models.

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

- **DOML**[19]: This model combines dual learning with metric learning. Using a pretrained autoencoder, DOML generates user and item embeddings and learns an orthogonal mapping to minimize the distance between overlapping user embeddings across domains.
- **PTUPCDR**[54]: This method utilizes a meta mapping network as a personalized bridge function within the EMCDR framework [27].
- **ACDR-GAN**[17]: This model employs adversarial learning to capture both global user preferences and domain-specific preferences across different domains.
- **DDTCDR**[18]: 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 an orthogonal mapping to preserve user relations in latent space.
- **DARec**[41]: This state-of-the-art framework employs a deep domain adaption model to extract and transfer patterns from rating matrices in different domains, without considering any auxiliary information.
- **EMCDR**[27]: 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**[1]: 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.

The source code of our methods is available in this link[3].

5.3 Performance Evaluation - Item-level and User-level relevance scenario

Table 4 and Table 5 summarize the average **item-level** relevance CSR performance results in terms of RMSE and MAE for the four examined data subsets presented in Table 2. Additionally, Table 6 summarizes the **user-level** relevance recommendation performance results (i.e., common users across the source and target domain) for the examined datasets given in Table 3 using firstly the Book domain as source domain and the Movie domain as target domain and next using the Music domain as source domain and the Book domain as target domain. It is evident that the proposed coupled autoencoder-based frameworks, (i.e., CACDR and LFACDR) gave better results than the other CDR methods for both the item-level and user-level relevance recommendation scenarios. 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.

Comparison against shallow learning approaches. The proposed models significantly outperform the CST, LFM and EMCDR approaches. In more detail, the above approaches 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. Furthermore, another major difference between the proposed models and the EMCDR methods is that, the EMCDR approaches learn the latent factors separately from the mapping function. However, this procedure is sub-optimal, since there is no influence between the domains during the learning of the latent factors. On the other hand, the proposed models, utilize the coupled autoencoders to extract the latent representations and transfer valuable information across the domains during the training phase.

Comparison against deep learning approaches. Although the state-of-the-art baselines that is the DDTCDR, MODL and PTUPCDR methods exhibit good recommendation results, our proposed models performed even better and that can be attributed to the following reason. The DDTCDR, DaRec, and PTUPCDR 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. This procedure is 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. In contrast, our methods utilize a coupled learning framework that jointly optimizes the autoencoders and the mapping function. This joint optimization ensures that the representations are well-aligned and transfer knowledge effectively across domains. More details regarding of the impact of coupled learning stage on the recommendation performance of our models that justifies their superiority is given in Section 6.2. Moreover, although the ACDR-GAN methodology demonstrates better performance compared to other considered methodologies, our approach produces superior results. This can be attributed to the instability and complexity inherent in training large GAN models. By utilizing a simple MLP

[3]https://drive.google.com/drive/folders/1Z_vxBsYw3GJSqSJ54sLq0CJKEVbQQDfW?usp=sharing

Table 4. Item-level relevance recommendation performance using the MovieLens dataset as source domain and the Netflix dataset as target domain. The best baseline values are underlined, and LFACDR's relative improvement over the best baseline is shown in parentheses.

Data Subset	Metrics	CST	LFM	MF_EMCDR	BPR_EMCDR	DDTCDR	DARec	DOML	PTUPCDR	ACDR-GAN	CACDR	LFACDR
No. 1	RMSE	0.2399	0.2341	0.2289	0.2214	0.2032	0.1783	0.1787	0.1790	0.1786	0.1772	**0.1740 (2.41%)**
	MAE	0.1877	0.1819	0.1757	0.1744	0.1648	0.1426	0.1371	0.1375	0.1405	0.1385	**0.1360 (0.80%)**
No. 2	RMSE	0.2314	0.2228	0.2079	0.2018	0.1956	0.1868	0.1811	0.1836	0.1802	0.1738	**0.1711 (5.08%)**
	MAE	0.1813	0.1761	0.1643	0.1617	0.1557	0.1496	0.1426	0.1449	0.1419	0.1356	**0.1331 (6.65%)**
No. 3	RMSE	0.2578	0.2513	0.2387	0.2324	0.2107	0.2093	0.2087	0.2099	0.2010	0.1980	**0.1922 (8.18%)**
	MAE	0.2181	0.2123	0.1868	0.1849	0.1699	0.1678	0.1679	0.1687	0.1654	0.1560	**0.1511 (9.96%)**
No. 4	RMSE	0.2687	0.2672	0.2508	0.2478	0.2229	0.2243	0.2198	0.2240	0.2197	0.2154	**0.2097 (4.60%)**
	MAE	0.2324	0.2292	0.2076	0.2042	0.1817	0.1805	0.1779	0.1798	0.1773	0.1698	**0.1665 (6.39%)**

and coupled autoencoders, our methodologies achieve better recommendation outcomes. The coupling of autoencoders allows for a simple and computationally efficient mapping function, enhancing the recommendation accuracy. Since the autoencoders are jointly optimized, the latent representations are aligned, making a simple MLP sufficient to map the aligned representations between domains. Finally, in Table 7, we compared our approaches with the best-performing method, specifically the ACDR-GAN approach, using ranking-based metrics. These additional metrics provide a more complete picture of the performance of our proposed methods. The results show that our approaches, CACDR and LFACDR, not only outperform the GAN method in terms of RMSE and MAE but also achieve better performance in ranking-based metrics, such as Precision@5 and Recall@5. Note that the metrics Precision@5 and Recall@5 were computed following the localized evaluation protocol described in [17], where the top-5 predictions are selected from the subset of items that each user has actually rated in the test set. This approach, commonly used in matrix completion literature, allows evaluation within the observed data distribution rather than across the full item catalog.

Comparison CACDR and LFACDR. Focusing on the proposed methods, the LFACDR method exhibits better results compared to the CACDR method in most 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. In CACDR, the full latent representation of the shared entity (user or item) is learned in the source domain and directly transferred to the target domain via a coupled mapping function. This method is efficient and captures domain-level alignment but relies entirely on the transferred representation to predict target interactions, without explicitly modeling the local (non-shared) entity in the target domain. By contrast, LFACDR improves on this by disentangling the interaction modeling process: Latent Factor Modeling of Both Users and Items: In LFACDR, instead of transferring a single representation learned from an autoencoder (as in CACDR), the model explicitly learns latent factor matrices for both users and items in both source and target domains, using four autoencoders. These factor matrices are derived from a joint constrained optimization problem (Eq. 10), which enforces both reconstruction fidelity and matrix factorization constraints. This leads to more structured and informative representations. Decomposition and Local–Global Fusion: Unlike CACDR, which relies solely on transferring the representation of the shared entity (user or item), LFACDR combines the mapped latent representation of the shared entity with the locally learned representation of the non-shared entity in the target domain. For example, in the item-level scenario, the predicted rating is computed as the dot product of the mapped source item latent factor and the target user latent factor. This combination ensures that both transferred knowledge and domain-specific patterns are leveraged in the final prediction.

Table 5. Item-level relevance recommendation performance using the Netflix dataset as source domain and MovieLens dataset as target domain. Best-performing baselines are underlined. LFACDR or CACDR improvement over the best baseline is shown in parentheses.

Data Subset	Metrics	CST	LFM	MF-EMCDR	BPR-EMCDR	DDTCDR	DARec	DOML	PTUPCDR	ACDR-GAN	CACDR	LFACDR
No. 1	RMSE	0.2378	0.2233	0.2206	0.2210	0.1998	0.1824	0.1803	0.1801	0.1713	0.1692	**0.1683** (0.58%)
	MAE	0.1868	0.1807	0.1741	0.1740	0.1589	0.1471	0.1421	0.1418	0.1354	0.1298	**0.1289** (0.48%)
No. 2	RMSE	0.2302	0.2208	0.2027	0.2002	0.1943	0.1812	0.1815	0.1813	0.1772	**0.1674**(5.52%)	0.1692
	MAE	0.1806	0.1747	0.1624	0.1604	0.1498	0.1457	0.1434	0.1428	0.1350	**0.1282**(5.04%)	0.1301
No. 3	RMSE	0.2541	0.2504	0.2333	0.2301	0.2088	0.2000	0.2081	0.2089	0.1998	0.1932	**0.1837** (8.07%)
	MAE	0.2153	0.2114	0.1851	0.1821	0.1644	0.1601	0.1601	0.1607	0.1565	0.1494	**0.1418** (9.37%)
No. 4	RMSE	0.2654	0.2672	0.2499	0.2464	0.2193	0.2010	0.2071	0.2031	0.2014	0.1993	**0.1975** (1.93%)
	MAE	0.2298	0.2293	0.2067	0.2014	0.1699	0.1652	0.1597	0.1683	0.1549	0.1534	**0.1511** (2.45%)

Table 6. User-level relevance recommendation performance using the Douban dataset. Best-performing baselines are underlined. LFACDR improvement over the best baseline is shown in parentheses.

Data	Metrics	CST	LFM	MF_EMCDR	BPR_EMCDR	DDTCDR	DARec	DOML	PTUPCDR	ACDR-GAN	CACDR	LFACDR
Book to Movie No.5	RMSE	0.2387	0.2347	0.2286	0.2245	0.1984	0.1888	0.1784	0.1881	0.1680	0.1652	**0.1647** (1.96%)
	MAE	0.1967	0.1935	0.1863	0.1824	0.1587	0.1506	0.1478	0.1500	0.1333	0.1299	**0.1276** (4.28%)
Music to Book No.6	RMSE	0.2573	0.2420	0.2348	0.2311	0.2002	0.1893	0.1778	0.1884	0.1776	0.1751	**0.1693** (4.67%)
	MAE	0.2128	0.2014	0.1987	0.1941	0.1597	0.1461	0.1425	0.1458	0.1411	0.1343	**0.1313** (6.94%)

Table 7. User-level relevance recommendation performance using the Douban dataset

Data	Metrics	ACDR-GAN	CACDR	LFACDR
Book to Movie No.5	RMSE	0.1680	0.1652	**0.1647**
	MAE	0.1333	0.1299	**0.1276**
	Pec@5	0.9708	0.9824	0.9835
	Recall@5	0.9967	0.9976	0.9987
Music to Book No.6	RMSE	0.1776	0.1751	**0.1693**
	MAE	0.1411	0.1343	**0.1313**
	Pec@5	0.9696	0.9756	0.9778
	Recall@5	0.9953	0.9968	0.9987

6 ABLATION STUDY

6.1 Impact of the mapping function

To demonstrate the impact of the mapping function on the recommendation performance of the proposed methods, we conducted experiments using both simpler and more complex architectures. Specifically, we employed a simpler architecture based on a Multilayer Perceptron (MLP) and a more complex one based on a Generative Adversarial Network (GAN). For the MLP-based mapping function, we used a two-layer architecture. The architecture of our GAN model includes a Generator and a Discriminator. The Generator is a two-layer fully connected feedforward neural network with ReLU activation functions. The Discriminator is also a two-layer fully connected feedforward neural network, utilizing LeakyReLU in the hidden layer and a Sigmoid activation function in the output layer.

As shown in Tables 8 and 9, the proposed methodologies with the GAN-based mapping function generally outperformed those with the MLP-based mapping function, although the latter still performed comparably well. Notably, our GAN architecture is significantly smaller compared to other approaches that utilize GAN models such as ACDR-GAN

Table 8. The performance of the proposed models utilizing two different mapping functions using the MovieLens dataset as source domain and the Netflix dataset target domain

Data	Metrics	CACDR GAN	CACDR MLP	LFACDR GAN	LFACDR MLP
No. 1	RMSE	0.1760	0.1772	0.1735	0.1740
	MAE	0.1375	0.1385	0.1354	0.1360
No. 2	RMSE	0.1728	0.1738	0.1702	0.1711
	MAE	0.1351	0.1356	0.1328	0.1331
No. 3	RMSE	0.1946	0.1980	0.1907	0.1922
	MAE	0.0152	0.1560	0.1502	0.1511
No. 4	RMSE	0.2108	0.2154	0.2075	0.2097
	MAE	0.1657	0.1698	0.1630	0.1665

Table 9. The performance of the proposed models utilizing two different mapping functions (Duban dataset)

Data	Metrics	CACDR GAN	CACDR MLP	LFACDR GAN	LFACDR MLP
No.5	RMSE	0.1631	0.1652	0.1620	0.1647
	MAE	0.1263	0.1299	0.1234	0.1276
No.6	RMSE	0.1644	0.1751	0.1611	0.1693
	MAE	0.1297	0.1343	0.1268	0.1313

method [17]. This can be attributed to two main factors. First, in our case, the GAN models only the intrinsic representations generated by the coupled autoencoders in a much lower, compact dimensional space, allowing for a simpler structure. Second, the coupling of autoencoders enhances the efficiency of the GAN, leading to better performance with less computational overhead. The jointly optimized autoencoders produce aligned latent representations, making a simple MLP sufficient to map the representations between domains.

6.2 Impact of Coupled learning stage

To demonstrate the impact of the coupled learning on the recommendation performance of the proposed methods, we conducted some experiments with and without the coupled learning stage during the training procedure of our methods. According to Table 10 and Table 11 which summarize the results for the item-level (we observed similar results using the Netflix dataset as source domain and Movielens domain as target domain) and user-level relevance scenario respectively, 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.

6.3 Impact of Initialization stage

As mentioned above the proposed methods consist of two main stages, i.e., the Initialization stage and the Coupled learning stage (see Figures 2 and 3). The initialization stage is a critical component affecting the recommendation performance of the proposed models. To highlight its importance, we conducted experiments with two training schemes.

Table 10. The performance of the proposed models with and without Coupled learning stage using the MovieLens dataset as source domain and the Netflix dataset target domain

Data	Metrics	CACDR without	CACDR with	LFACDR without	LFACDR with
No. 1	RMSE	0.1827	0.1772	0.1849	0.1740
	MAE	0.1441	0.1385	0.1477	0.1360
No. 2	RMSE	0.1815	0.1738	0.2101	0.1711
	MAE	0.1420	0.1356	0.1642	0.1331
No. 3	RMSE	0.2183	0.1980	0.2094	0.1922
	MAE	0.1723	0.1560	0.1651	0.1511
No. 4	RMSE	0.2333	0.2154	0.2288	0.2097
	MAE	0.1860	0.1698	0.1816	0.1665

Table 11. The performance of the proposed models with and without Coupled learning stage using the Douban Dataset

Data	Metrics	CACDR without	CACDR with	LFACDR without	LFACDR with
No.5	RMSE	0.1730	0.1652	0.1728	0.1647
	MAE	0.1347	0.1299	0.1342	0.1276
No.6	RMSE	0.1836	0.1751	0.1785	0.1693
	MAE	0.1435	0.1343	0.1411	0.1313

During the first scheme, called *with Initialization*, the two proposed models trained in two stages, the initialization stage where the involved autoencoders along with the mapping network are trained independently and the coupled learning stage where the autoencoders and the mapping network are optimized end-to-end. During the second scheme, called *without Initialization*, we employed the coupled learning stages of the two proposed methods without the autoencoders and mapping networks are initialized properly (i.e., without the Initialization stage). Table 12 and Table 13 summarizes the results. It is evident that the initialization stage plays an important role, improving the performance of our methods.

Considering the vastness of the parameter space of a deep learning model, an inappropriate initialization procedure of the proposed models may affect their performance. In other words, since the neural networks are non-convex functions containing a large number of local minima, an improper initialization of the parameters of the models may lead the optimization process to get stuck on local minima with suboptimal performance. On the other hand, the proposed two stage training process guarantees that during the initialization stage the models will be initialized with proper parameters adapted to the statistical distributions of the training data, and during the coupled learning stage, where the models are optimized end-to-end, these parameters will be preserved allowing the proposed methods to further adapt to the structure of the data and hence improve the performance.

6.4 Impact of Latent Dimension and Complexity Analysis

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

Table 12. The performance of the proposed models with and without Initialization stage using the MovieLens dataset as source domain and the Netflix dataset target domain

Data	Metrics	CACDR without	CACDR with	LFACDR without	LFACDR with
No. 1	RMSE	0.1801	0.1772	0.1814	0.1740
	MAE	0.1406	0.1385	0.1435	0.1360
No. 2	RMSE	0.1769	0.1738	0.1799	0.1711
	MAE	0.1394	0.1356	0.1398	0.1331
No. 3	RMSE	0.2066	0.1980	0.2004	0.1922
	MAE	0.1597	0.1560	0.1591	0.1511
No. 4	RMSE	0.2205	0.2154	0.2126	0.2097
	MAE	0.1740	0.1698	0.1692	0.1665

Table 13. The performance of the proposed models with and without Initialization stage using the Douban dataset

Data	Metrics	CACDR without	CACDR with	LFACDR without	LFACDR with
No.5	RMSE	0.1725	0.1652	0.1746	0.1647
	MAE	0.1341	0.1299	0.1357	0.1276
No.6	RMSE	0.1812	0.1751	0.1799	0.1693
	MAE	0.1421	0.1343	0.1412	0.1313

8, 32, 64, 128, 256. Table 15 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 mentioned that the LFACDR method exhibited better performance compared to the CACDR method in most 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. Additionally, from table 15 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. Similar results, we observed using the Douban dataset, exploring the user-level relevance scenario.

Complexity analysis for the proposed methods and the other compared recommendation approaches is summarized in Table 14. To simplify the analysis, only the main steps of each approach are considered, where the n denote the number of users, m represents the number of items, k denotes the latent dimension, L is the number of the layers of the neural network, and T stands for the number of iterations.

7 CONCLUSIONS

We have explored an item-level relevance CSR 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. Additionally, we examined a user-level relevance CDR scenario where the two related domains contain common users. 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

Table 14. Complexity analysis of the proposed methods and the compared baselines .

Methods	Computational Complexity
CACDR	$O(n * L * k * T)$
LFACDR	$O((n + m) * L * k * T)$
DARec	$O((n + m) * L * k * T)$
DDTCDR	$O((n + m) * L * k * T)$
EMCDR	$O((n + m) * k * T)$
CST	$O((n + m) * k * T)$
LFM	$O((n + m) * k * T)$

Table 15. The impact of different latent dimensions on Recommendation performance of the proposed models using the Movielens as source domain and the Netflix as target domain

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	**0.1711**	**0.1331**	0.1742	0.1361

does not make this assumption. We demonstrated some very promising results, in comparison to some popular methods on cross-domain recommendation. We used portions of the MovieLens, Netflix and Douban 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.

The limitations of the proposed approach can be summarized in the following:

- Privacy and Regulation: Coupling inherently requires joint optimization involving data from both domains, which could raise privacy and regulatory concerns, especially in federated or multi-organization scenarios.
- Incomplete Overlap: Our current evaluation explicitly focuses on complete overlap scenarios. While coupling can be theoretically extended to partial overlaps, it introduces additional complexity in alignment and may require careful handling.
- Computational Complexity: Coupling via fine-tuning and joint training can increase computational complexity, as it requires simultaneous optimization of multiple networks.

We aim to further investigate these issues in our future research.

8 ACKNOWLEDGMENTS

This work is partially supported by the Greek Secretariat for Research and Innovation and the EU, Project Muselearn, T1EDK-00502 within the framework of "Competitiveness, Entrepreneurship and Innovation" (EPAnEK) Operational Programme 2014-2020.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Bo Long. 2011. 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, 609–617.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (jul 2013), 109–132.

[3] Pascal Vincent@umontreal Ca, Larocheh@cs Toronto Edu, Isabelle Lajoie, Yoshua Bengio@umontreal Ca, and Pierre-Antoine Manzagol@umontreal Ca. 2010. *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pascal Vincent Hugo Larochelle Yoshua Bengio Pierre-Antoine Manzagol*. Technical Report. 3371–3408 pages.

[4] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender Systems Handbook, Second Edition*. Springer US, 919–959.

[5] Da Cao, Xiangnan He, Liqiang Nie, Xiaochi Wei, Xia Hu, Shunxiang Wu, and Tat-Seng Chua. 2017. Cross-Platform App Recommendation by Jointly Modeling Ratings and Texts. *ACM Trans. Inf. Syst.* 35, 4, Article 37 (July 2017), 27 pages. https://doi.org/10.1145/3017429

[6] Jiangxia Cao, Xixun Lin, Xin Cong, Jing Ya, Tingwen Liu, and Bin Wang. 2022. DisenCDR: Learning Disentangled Representations for Cross-Domain Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 267–277. https://doi.org/10.1145/3477495.3531967

[7] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. 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* (Florence, Italy) *(WWW '15)*. International World Wide Web Conferences Steering Committee, 278–288.

[8] 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 *33rd AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, 94–101.

[9] Chen Gao, Kai Zhao, Xiangning Chen, Xiangnan He, Depeng Jin, Fuli Feng, and Yong Li. 2019. Cross-domain recommendation without sharing user-relevant data. In *The World Wide Web Conference*. Association for Computing Machinery, Inc, 491–502.

[10] Ming He, Jiuling Zhang, and Shaozong Zhang. 2019. ACTL: Adaptive Codebook Transfer Learning for Cross-Domain Recommendation. *IEEE Access* 7 (2019), 19539–19549.

[11] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 667–676. https://doi.org/10.1145/3269206.3271684

[12] Qi-Ying Hu, Zhi-Lin Zhao, Chang-Dong Wang, and Jian-Huang Lai. 2017. An Item Orientated Recommendation Algorithm from the Multi-View Perspective. *Neurocomput.* 269, C (Dec. 2017), 261–272. https://doi.org/10.1016/j.neucom.2016.12.102

[13] Tomoharu Iwata and Takeuchi Koh. 2015. 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, 379–387.

[14] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. 2019. Cross-Domain Recommendation via Deep Domain Adaptation. In *Advances in Information Retrieval*, Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (Eds.). Springer International Publishing, Cham, 20–29.

[15] Oleksii Kuchaiev and Boris Ginsburg. 2017. Training Deep AutoEncoders for Collaborative Filtering. *arXiv* (aug 2017). arXiv:1708.01715.

[16] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (Pasadena, California, USA) *(IJCAI'09)*. 2052–2057.

[17] Pan Li, Brian Brost, and Alexander Tuzhilin. 2022. Adversarial Learning for Cross Domain Recommendations. *ACM Trans. Intell. Syst. Technol.* 14, 1, Article 5 (nov 2022), 25 pages. https://doi.org/10.1145/3548776

[18] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep dual transfer cross domain recommendation. *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining* (jan 2020), 331–339. https://doi.org/10.1145/3336191.3371793

[19] Pan Li and Alexander Tuzhilin. 2021. Dual metric learning for effective and efficient cross-domain recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 321–334.

[20] Tzu-Heng Lin, Chen Gao, and Yong Li. 2019. CROSS: Cross-Platform Recommendation for Social E-Commerce. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 515–524. https://doi.org/10.1145/3331184.3331191

[21] Jian Liu, Pengpeng Zhao, Fuzhen Zhuang, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Xiaofang Zhou, and Hui Xiong. 2020. Exploiting Aesthetic Preference in Deep Cross Networks for Cross-Domain Recommendation. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2768–2774. https://doi.org/10.1145/3366423.3380036

[22] Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. 2021. Leveraging Distribution Alignment via Stein Path for Cross-Domain Cold-Start Recommendation. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=dYGFRxCf7P

[23] Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. 2021. Leveraging Distribution Alignment via Stein Path for Cross-Domain Cold-Start Recommendation. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 19223–19234. https://proceedings.neurips.cc/paper_files/paper/2021/file/a0443c8c8c3372d662e9173c18faaa2c-Paper.pdf

[24] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-Domain Collaborative Filtering with Factorization Machines. In *Advances in Information Retrieval*, Maarten de Rijke, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann (Eds.). Springer International Publishing, Cham, 656–661.

[25] Guofang Ma, Yuexuan Wang, Xiaolin Zheng, Xiaoye Miao, and Qianqiao Liang. 2021. A trust-aware latent space mapping approach for cross-domain recommendation. *Neurocomputing* 431 (mar 2021), 100–110.

[26] Jingwei Ma, Jiahui Wen, Mingyang Zhong, Weitong Chen, and Xue Li. 2019. MMM: Multi-source Multi-net Micro-video Recommendation with Clustered Hidden Item Representation Learning. *Data Science and Engineering 2019 4:3* 4, 3 (sep 2019), 240–253. https://doi.org/10.1007/S41019-019-00101-4

[27] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI-17*. 2464–2470.

[28] Jarana Manotumruksa, Dimitrios Rafailidis, Craig Macdonald, and Iadh Ounis. 2019. On Cross-Domain Transfer in Venue Recommendation. In *Advances in Information Retrieval*, Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (Eds.). Springer International Publishing, Cham, 443–456.

[29] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. 2011. Contextual Video Recommendation by Multimodal Relevance and User Feedback. *ACM Trans. Inf. Syst.* 29, 2, Article 10 (April 2011), 24 pages. https://doi.org/10.1145/1961209.1961213

[30] Senthilselvan Natarajan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H. Gandomi. 2020. Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert Systems with Applications* 149 (jul 2020), 113248. https://doi.org/10.1016/j.eswa.2020.113248

[31] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[32] Weike Pan, Evan W Xiang, Nathan N Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24.

[33] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert Systems with Applications* 39, 11 (2012), 10059–10072. https://doi.org/10.1016/j.eswa.2012.02.038

[34] Jin Shang, Mingxuan Sun, and Kevyn Collins-Thompson. 2018. Demographic Inference Via Knowledge Transfer in Cross-Domain Recommender Systems. In *2018 IEEE International Conference on Data Mining (ICDM)*. 1218–1223. https://doi.org/10.1109/ICDM.2018.00162

[35] 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*. ACM Press, 650–658.

[36] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-Domain Collaboration Recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. Association for Computing Machinery, New York, NY, USA, 1285–1293. https://doi.org/10.1145/2339530.2339730

[37] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '11)*. Association for Computing Machinery, 448–456.

[38] Cheng Wang, Mathias Niepert, and Hui Li. 2020. RecSys-DAN: Discriminative Adversarial Networks for Cross-Domain Recommender Systems. *IEEE Transactions on Neural Networks and Learning Systems* 31, 8 (2020), 2731–2740. https://doi.org/10.1109/TNNLS.2019.2907430

[39] Chang-Dong Wang, Zhi-Hong Deng, Jian-Huang Lai, and Philip S. Yu. 2019. Serendipitous Recommendation in E-Commerce Using Innovator-Based Collaborative Filtering. *IEEE Transactions on Cybernetics* 49, 7 (2019), 2678–2692. https://doi.org/10.1109/TCYB.2018.2841924

[40] Xin Xin, Zhirun Liu, Chin-Yew Lin, Heyan Huang, Xiaochi Wei, and Ping Guo. 2015. Cross-Domain Collaborative Filtering with Review Text. In *Proceedings of International Joint Conference on Artificial Intelligence*. 1827–1834.

[41] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 4227–4233.

[42] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2022. A Survey on Cross-Domain Recommendation: Taxonomies, Methods, and Future Directions. *ACM Trans. Inf. Syst.* 41, 2, Article 42 (dec 2022), 39 pages. https://doi.org/10.1145/3548455

[43] Qian Zhang, Wenhui Liao, Guangquan Zhang, Bo Yuan, and Jie Lu. 2021. A Deep Dual Adversarial Network for Cross-domain Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3132953

[44] Qian Zhang, Jie Lu, Dianshuang Wu, and Guangquan Zhang. 2019. A Cross-Domain Recommender System With Kernel-Induced Knowledge Transfer for Overlapping Entities. *IEEE Transactions on Neural Networks and Learning Systems* 30, 7 (2019), 1998–2012. https://doi.org/10.1109/TNNLS.2018.2875144

[45] Xinyue Zhang, Jingjing Li, Hongzu Su, Lei Zhu, and Heng Tao Shen. 2023. Multi-level Attention-based Domain Disentanglement for BCDR. *ACM Trans. Inf. Syst.* 41, 4, Article 97 (mar 2023), 24 pages. https://doi.org/10.1145/3576925

[46] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-Domain Active Learning for Recommendation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2358–2364.

[47] Zhipeng Zhang, Yao Zhang, and Yonggong Ren. 2020. Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering. *Information Retrieval Journal 2020 23:4* 23, 4 (jun 2020), 449–472. https://doi.org/10.1007/S10791-020-09378-W

[48] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-Domain Recommendation via Preference Propagation GraphNet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2165–2168. https://doi.org/10.1145/3357384.3358166

[49] Shi-Ting Zhong, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and Philip S. Yu. 2020. An Autoencoder Framework With Attention Mechanism for Cross-Domain Recommendation. *IEEE Transactions on Cybernetics* (2020), 1–13. https://doi.org/10.1109/TCYB.2020.3029002

[50] 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 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1533–1542. https://doi.org/10.1145/3357384.3357992

[51] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. 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*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3001–3008. https://doi.org/10.24963/ijcai.2020/415 Main track.

[52] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-Domain Recommendation: Challenges, Progress, and Prospects. In *IJCAI*.

[53] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu. 2023. A Unified Framework for Cross-Domain and Cross-System Recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 1171–1184. https://doi.org/10.1109/TKDE.2021.3104873

[54] 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 fifteenth ACM international conference on web search and data mining*. 1507–1515.