FedCT: Federated Collaborative Transfer for Recommendation

Shuchang Liu[1], Shuyuan Xu[1], Wenhui Yu[2], Zuohui Fu[1], Yongfeng Zhang[1], Amélie Marian[1]
[1]Rutgers University, New Brunswick, NJ, USA [2]Alibaba Group, Beijing, China
[1]{shuchang.syt.liu, shuyuan.xu, zuohui.fu, yongfeng.zhang, amelie.marian}@rutgers.edu
[2]{jianlin.ywh}@alibaba-inc.com

ABSTRACT

When a user starts exploring items from a new area of an e-commerce system, cross-domain recommendation techniques come into help by transferring the abundant knowledge from the user’s familiar domains to this new domain. However, this solution usually requires direct information sharing between service providers on the cloud which may not always be available and brings privacy concerns. In this paper, we show that one can overcome these concerns through learning on edge devices such as smartphones and laptops. The cross-domain recommendation problem is formalized under a decentralized computing environment with multiple domain servers. And we identify two key challenges for this setting: the unavailability of direct transfer and the heterogeneity of the domain-specific user representations. We then propose to learn and maintain a decentralized user encoding on each user’s personal space. The optimization follows a variational inference framework that maximizes the mutual information between the user’s encoding and the domain-specific user information from all her interacted domains. Empirical studies on real-world datasets exhibit the effectiveness of our proposed framework on recommendation tasks and its superiority over domain-pairwise transfer models. The resulting system offers reduced communication cost and an efficient inference mechanism that does not depend on the number of involved domains, and it allows flexible plugin of domain-specific transfer models without significant interference on other domains.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**.

KEYWORDS

Recommendation System; Federated Learning; Transfer Learning; User Representation

ACM Reference Format:
Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, Amélie Marian. 2021. FedCT: Federated Collaborative Transfer for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462825

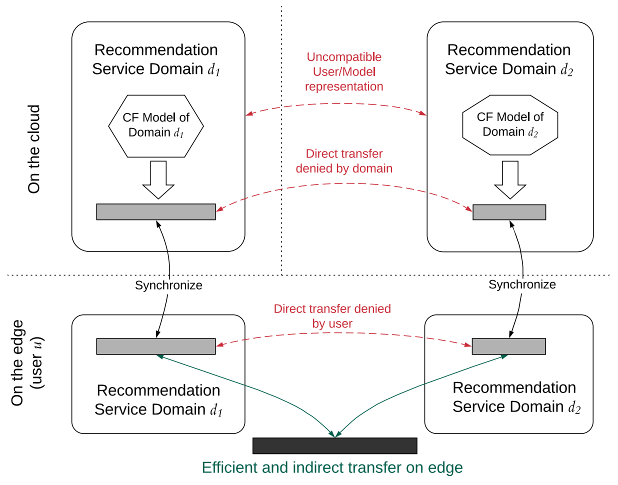
On the cloud
Recommendation Service Domain d_1
CF Model of Domain d_1
Uncompatible User/Model representation
Direct transfer denied by domain
Recommendation Service Domain d_2
CF Model of Domain d_2
Synchronize
Synchronize
On the edge (user u)
Direct transfer denied by user
Recommendation Service Domain d_1
Recommendation Service Domain d_2
Efficient and indirect transfer on edge

Figure 1: Edge_CDR problem - indirect transfer of user information on user’s local device.

1 INTRODUCTION

Recommender system (RS) [1], as one of the principle information retrieval (IR) tools, aims to find items (e.g. products and movies) for users, fulfilling their demands for information search and exploration. The most adopted and efficient solutions nowadays follow the collaborative learning (CL) strategy, which includes collaborative filtering [6, 30] and collaborative reasoning [4, 33]. Generally, the CL-based methods benefit from more elaborate user profiles and longer interaction histories, but they tend to face difficulties when they only observe limited information of the user. An extreme but common case is that a user comes into a new area/domain for exploration with no preceding interaction record, known as the *user* cold-start problem [29]. One distinguishable solution to this problem is the Cross-Domain Recommendation (CDR) approach [11, 19, 36], which transfers knowledge about the user from auxiliary RS data sources (domains) where abundant information and interaction history of this user is available. For example, Kathryn is a new customer for an e-book store. And in another app, she has bought lots of horror and thrilling movies, then these preferences can benefit the e-book store’s recommendation service to Kathryn.

To our knowledge, the CDR problem has been mostly researched under centralized machine learning frameworks, which requires direct user information sharing between different services/domains. However, this is often not an option when domains distrust each other. For example, two domains are in competition and does not want to reveal sensitive information like model design. Furthermore, this direct sharing scheme raises privacy concerns not only for the domains but also for users who may also deny the transfer in order to protect its information [2, 43]. In this paper, we argue that one can overcome these limitations by transfer learning on edge devices and including both the domains and the users into the control of the transfer. Specifically, a new CDR problem, denote as Edge-CDR, is reformulated under a decentralized edge/mobile computing environment along with multiple central server. In addition to each service provider's cloud space that provide domain-specific centralized environments, each user is associated with a personal space (e.g. mobile phones or laptops) whose data and computation are locally maintained on the edge device.

We summarize the key constraints of Edge-CDR as follows: First, **direct transfer between domains** becomes intermittently unavailable as previously mentioned. In contrast, Edge-CDR allows direct transfer between a user's personal space and any domain-specific service, indicating a possible indirect transfer between domains (Figure 1) where the user's personal space serves as intermediate controller. Secondly, Edge-CDR assumes **heterogeneous (user/model) representations** across domains since real-world services do not necessarily share the same way of representing users and generating recommendations, neither do they have to sacrifice their model configurations to support joint training with other services. As a result, methods that involve fixed and complicated transfer mechanism [9, 28] among domains becomes expensive or even impractical. Thirdly, a straightforward assumption for each edge device is the **consistent user state** across domains (e.g. it is the same user using the same mobile phone). Therefore, methods that deal with no-user-overlap scenarios [15] are no longer favorable. In general, we argue that the most natural Edge-CDR setting is the user-overlap no-item-overlap (U-NI) setting. Following such initiatives, the edge environment of users and cloud services of domains should complement each other when solving the Edge-CDR task, as shown in Figure 1. Specifically, each personal space is unable to acquire the knowledge about other users but is well suited for transferring knowledge across domains about the same user; On the other hand, each domain service has difficulty retrieving knowledge from other domains but is well suited for running collaborative learning across users within its domain.

To overcome the aforementioned restrictions, we propose to maintain a Decentralized User Encoding (DUE) on each personal device to engage indirect user knowledge transfer, and the transfer model is separable by domain so that each service can independently manage its domain-specific user representations and its prediction models. In summary, we list our contributions as the following:

- We formalize the Edge-CDR problem and show that maintaining DUEs on edge/personal devices can overcome the challenge of indirect transfer and heterogeneous cross-domain user representations.
- We illustrate our solution as a federated variational inference framework and empirically show its validity on large-scale real-world datasets.
- We also show that our framework is efficient compared to domain-pairwise transfer methods and is flexible for domain plug-in/plug-out.

The remainder of this paper is organized as follows: we discuss the scale of the Edge-CDR problem and its related areas in section 2. In section 3, we first formulate the Edge-CDR problem as a user embedding transfer learning problem and then illustrate the restrictions. In section 4 we present the DUE model and its optimization framework. We elaborate our experimental settings and empirical results in section 5. Finally, section 6 concludes our study and discusses its future expectations.

2 RELATED WORK

We locate the scale of Edge-CDR in the joint research area of recommender systems, transfer learning, and mobile computing systems. To our knowledge, there is no existing work that accommodates the Edge-CDR setting, but closely related topics can be found in the joint area of two of the three fields, correspond to the CDR problem, the federated recommendation, as well as the more general federated transfer-learning.

In terms of **cross-domain recommendation**, instead of restricting the transfer scenario to U-NI, there are other scenarios that share items or share both users and items across domains [15]. There are also studies on how to transfer knowledge between different data formats, for example: from knowledge graph or review text to recommendation [10, 13, 41]. In this work, we focus on category-level domain transfer where domains adopt the same data type but differ in their item categories. Pioneer work of this kind considers clustering-based methods [19] to find transferable user/item patterns between domains. Others have shown alternatives that extends the original user-item interaction matrix into a user-item-domain tensor and apply factorization-based methods [11, 36]. Recent developments further improve the state-of-the-art solutions by not only sharing user representations but also model parameters [5, 9, 28]. Though these methods are effective in a centralized setting, they either require direct transfer or assume homogeneous user/model representations between domains. We consider the *embedding and mapping* approaches [14, 20, 24, 32, 40] as the closest setting to Edge-CDR. This type of methods separates the optimization process into a domain-independent embedding learning phase and a cross-domain transfer learning phase so that systems with different user embedding formats can exchange information without much negative influence on their performance. However, they still require direct user embedding transfer. Moreover, most of these existing CDR models learn to transfer between a pair of domains, but a mobile device in Edge-CDR setting typically involves a large number of services/domains, which could induce an excessive quadratic cost (section 5.5).

On the other hand, it has been an increasing interest [17, 21, 34] to migrate centralized algorithms into personal devices since—with sufficient computational power—the decentralized workload can reduce the computation of the central server and can offer faster updates and responses [27] on edge devices. In terms of designing RS model with distributed data and computation, local training on personal devices and cross user collaboration are necessary, which is often referred as horizontal **federated RS** [3, 17]. And [39] shows the convergence rate for the general federated gradient-based optimizations. We believe that, with the development of ubiquitous personal smart agents and the internet of things (IoT) infrastructures, this migration will become a necessity in the near future for not only recommendation tasks but also many other

Symbol	Description
$\mathcal{D}$,$\mathcal{D}^{(u)}$	set of domains and its subset interacted by user u
D	total number of domains
d, d', d_t	domain identifiers, the last one is target domain
$\mathcal{I}_d$	item set of domain d
$\mathcal{U}$,$\mathcal{U}^{(d)}$	set of user and its subset interacted by domain d
M	total number of user
u, i	user and item identifiers
$r, \hat{r}$	ground truth and predicted user response
f_d	domain-specific CF model
$\mathbf{U}_u^{(d)}$	user u's embedding in domain d
L_d	user embedding size in domain d
L_c	embedding size of DUE
g	the transfer model that maps user embeddings
$\mathcal{S}_d$	domain's service space on the cloud
$\mathcal{E}_u$	user's personal space on device
$\mathcal{E}_{d,u}$	domain's service space on device
$\mathbf{z}_u\vert\mathcal{E}_u$	DUE of user u
$\mathbf{z}_u\vert\mathbf{U}_u^{(d)}$	DUE inferred by domain d
$\text{ENC}_d, \text{DEC}_d$	encoder and decoder of domain d
$\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d$	variational output of domain-specific encoder
$\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u$	variational information stored on $\mathcal{E}_u$
Θ_d, Φ_d	model parameters of domain-specific decoder and encoder

Table 1: Notations

web services. In contrast with horizontal federated learning across users, there is a vertical federated learning problem where the data is localized by features or domains [21, 23, 37, 42], which is also closely related to domain transfer problems. In this paper, we regard Edge-CDR as a **federated transfer learning** problem [42] that requires a solution that is both horizontally separated by user and vertically separated by domains. Additionally, in systematic perspective, researchers have also put efforts into improving the privacy guarantee [25, 35] and robustness [31] of the federated learning system. We consider these ideas as complementary to ours while we focus the algorithmic aspects of the Edge-CDR problem.

3 PROBLEM FORMULATION

In this section, we give the Edge-CDR problem definition analog to that of centralized CDR and then describe the restrictions when migrating it to an edge computing environment.

3.1 Edge-CDR with U-NI Scenario

Assume that recommendations services are separately provided in a set of D domains $\mathcal{D} = \{d_1, \dots, d_D\}$ each of which corresponds to an item category (e.g. Movies, Books, Games, etc). For each domain $d \in \mathcal{D}$, denote the set of items as $\mathcal{I}_d$ (of size N_d). As defined in U-NI setting, there is no item overlap between domains such that $\forall d, d' \in \mathcal{D}, d \neq d', \mathcal{I}_d \cap \mathcal{I}_{d'} = \emptyset$. Since we assume consistent user identifier on the same device, denote a universal set of user $\mathcal{U}$ (of size M) across domains. Note that a domain d may only engage with a subset of users $\mathcal{U}^{(d)} \subseteq \mathcal{U}$ and a user $u \in \mathcal{U}$ may only engage with a subset of domains $\mathcal{D}^{(u)} \subseteq \mathcal{D}$. The observed domain data can be expressed as a set of triplets $\Omega_d = \{(u, i, r) | u \in \mathcal{U}, i \in \mathcal{I}_d, r \in \mathcal{R}_d\}$ where $\mathcal{R}_d$ is the domain-specific response space (e.g. {1-5} if ratings, {0,1} if click signal). Typically, each domain separately owns and maintains its CF model f_d which is a function that predicts the response of a given user-item pair (u, i) within its knowledge: $\hat{r}_{u,i,d} = f_d(\mathbf{U}_u^{(d)}, i)$, where $\mathbf{U}_u^{(d)} \in \mathbb{R}^{M\times L_d}$ denotes the user's embedding in domain d and L_d is the dimension size of the user vector. In Edge-CDR setting, f_d along with $\mathbf{U}^{(d)}$ are pretrained by a domain-specific objective $\mathcal{L}_d^{(\text{CF})}$ and then serve as part of the training environment for the cross-domain transfer model.

The ultimate goal is to provide recommendations for a cold-start user in the target domain $d_t \in \mathcal{D}$ with the user's embedding transferred from other source/auxiliary domains. Similar to [24], given pre-trained f_{d_t} as environment and the user information across domains as training data, the problem becomes learning the user representation transfer model $g : \mathbb{R}^{L_{d_1}+L_{d_2}+\dots+L_{d_D}} \rightarrow \mathbb{R}^{L_{d_t}}$ that maps user information from all domains to d_t:

$$\widehat{\mathbf{U}}_u^{(d_t)} = g(\mathbf{X}_u) \tag{1}$$

where both f_d and the user representations $X_u = [\mathbf{U}_u^{(d_1)}, \dots, \mathbf{U}_u^{(d_D)}]$ are assumed fixed during the optimization and inference. In other words, the transfer model is agnostic to the design of f_d and $\mathcal{L}_d^{(\text{CF})}$. The problem is trivial if all domains collaboratively train a shared g on a centralized environment, but as mentioned in section 1, there arises additional restrictions such as unavailable direct transfer and heterogeneous user representations in Edge-CDR setting. In general, we regard this problem as a federated collaborative transfer learning task that requires both federation of multiple domain services and the collaboration between user spaces.

3.2 Unavailable Direct Transfer

The domain-specific environment f_d as well as the domain-specific user embeddings $\mathbf{U}^{(d)}$ cannot be directly shared with other domains, as mentioned in section 1. Formally, the system defines three data/model space with different level of accessibility:

- $\mathcal{S}_d$: the domain service space on the cloud (where f_d and $\mathbf{U}^{(d)}$ are stored).
- $\mathcal{E}_u$: the user's own space on device. Any domain should request for permission to access its data.
- $\mathcal{E}_{d,u}$: the domain's service space on user's personal device. It can synchronize with $\mathcal{S}_d$ to acquire user's information and communicate (if permitted) with the user space $\mathcal{E}_u$ when exchanging information.

As shown in Figure 1, this setting prohibit direct information transfer between $\{\mathcal{S}_d, \mathcal{E}_{d,u}\}$ and $\{\mathcal{S}_{d'}, \mathcal{E}_{d',u}\}$ for any pair of domain (d, d'). However, we assume that encoded information can be exchanged between the user's personal space $\mathcal{E}_u$ and any domain's space $\mathcal{E}_{d,u}$ on edge. This indicates that the users' personal spaces should serve as intermediate agents that coordinate an indirect transfer framework, and the learning goal must be simultaneously separable by users and separable by domains.

3.3 Heterogeneous User Representation

As described in section 3.1, one domain's user embeddings $\mathbf{U}^{(d)}$ are assumed independent from those from other domains, so both the user vector size L_d and the semantic meaning of each factor could

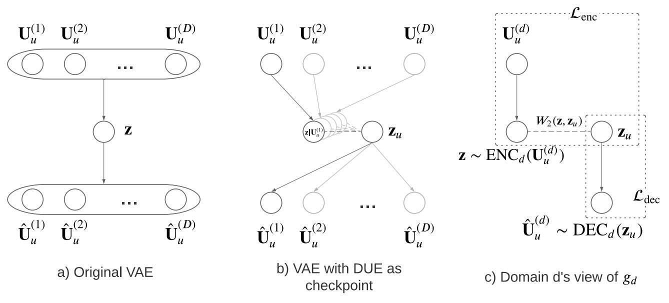
$\mathbf{U}_u^{(1)}$ $\mathbf{U}_u^{(2)}$ $\mathbf{U}_u^{(D)}$
$\mathbf{z}$
$\hat{\mathbf{U}}_u^{(1)}$ $\hat{\mathbf{U}}_u^{(2)}$ $\hat{\mathbf{U}}_u^{(D)}$
a) Original VAE
$\mathbf{z}_u$
b) VAE with DUE as checkpoint
$\mathcal{L}_{\text{enc}}$
$\mathbf{U}_u^{(d)}$
$W_2(\mathbf{z}, \mathbf{z}_u)$
$\mathbf{z} \sim \text{ENC}_d(\mathbf{U}_u^{(d)})$
$\mathcal{L}_{\text{dec}}$
$\hat{\mathbf{U}}_u^{(d)} \sim \text{DEC}_d(\mathbf{z}_u)$
c) Domain d's view of g_d

Figure 2: Graph Model for DUE framework. Each domain distribute its auto-encoders to all interacted users, and its encoding is regulated users' stored encoding z_u on edge. Each user's domain-specific information $\mathbf{U}_u^{(d)}$ is used to train the domain-specific encoders and decoders and does not directly require knowledge from other domains.

vary according to each domain d. We regard this as the vertical heterogeneity of user representations. Intuitively, one can simply learn a mapping function for each pair of domains like that in [24]. Yet, this domain-pairwise transfer scheme would not only violate the constraint of unavailable direct transfer but also become computationally heavy as the number of domain grows. What we need is a framework that can effectively deal with this semantic heterogeneity and its computation should be efficient with respect to $|\mathcal{D}|$. On the other hand, users representations are also horizontally heterogeneous since people typically have different demands and interest, thus, may engage with different subset of domains, where $\forall u \in \mathcal{U}, 1 \leq |\mathcal{D}^{(u)}| \leq |\mathcal{D}|$. In reality, users demands are usually limited, so including more domains induces a higher chance of seeing less common users between a pair of domains, especially for domains that are extremely sparse (in terms of the observed user-item interaction). This would result in a further downgraded performance of domain-pairwise transfer, as we will show in section 5.2. Moreover, in a user's view, she may add new domains or stop visiting previous domains, so the solution should also provide the flexibility of domain plug-in and pull-off mechanism without significant interference to the transfer model g on other domains.

4 LEARN DECENTRALIZED USER ENCODINGS

4.1 Overall Framework

We design our solution as a modified Variational Auto-Encoder (VAE)[16], which is not only a horizontal collaboration framework of users, but also a vertical federation framework across domains. It consists of a set of domain-specific partial encoders and decoders, and each user's personal space $\mathcal{E}_u$ maintains a Decentralized User Encoding (DUE) $\mathbf{z}_u \in \mathbb{R}^{L_c}$ which is a latent vector of size L_c that integrates user representations from all interacted domains. Specifically, we break down the variational model into a set of domain-specific VAEs $g_d, \forall d \in \mathcal{D}$. As shown in figure 2, for a given domain d, g_d consists of an encoder $\text{ENC}_d : \mathbb{R}^{L_d} \to \mathbb{R}^{L_c}$ (with parameters Φ_d) that infer $\mathbf{z}_u$ based on domain-specific representation $\mathbf{U}^{(d)}$, and a decoder $\text{DEC}_d : \mathbb{R}^{L_c} \to \mathbb{R}^{L_d}$ (with parameters Θ_d) that generate $\mathbf{U}^{(d)}$ based on $\mathbf{z}_u$. The objective of the system is to learn a set of decentralized $\mathbf{z}_u$ for each user that contains "complete user information" from all domains, so that each domain-specific $\text{DEC}_d(\cdot)$ can extract accurate $\mathbf{U}_u^{(d)}$ that contains sufficient information for later recommendation services. On one hand, each domain needs to learn its own decoder and encoder, so it can keep its user embedding format without modifying the design of its CF model f_d and the transfer models of other domains, solving the heterogeneous user representation. On the other hand, the only auxiliary information for each domain is the encoded $\mathbf{z}_u$ that is maintained by $\mathcal{E}_u$ from all interacted domains, so this mechanism avoids direct access of other domains for their user features, representation format, and model configurations.

4.2 Learning Separable Model and DUEs

The overall objective of this transfer framework is to maximize the data log likelihood of each user's representations $\mathbf{X}_u$ from all observed domains:

$$\sum_u \log p(\mathbf{X}_u) \tag{2}$$

which is already horizontally separable by user. Then one can assumes that the observed user representation $\mathbf{X}_u$ is generated from the same latent "complete" information $\mathbf{z}_u$ of that user:

$$p_{X|Z}(\mathbf{X}_u|\mathbf{z}_u) = \prod_d p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u) \tag{3}$$

Note that this formulation also indicates conditional independence between domain-specific user information given latent encoding, and mechanically, $p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u)$ and $p_{X|Z}(\mathbf{U}_u^{(d')}|\mathbf{z}_u)$ are modeled by decoders from corresponding domains d and d'.

Similar to the reasoning of VAE, one can simultaneously optimize the decoder Eq.(3) and learn to infer the encoding $\mathbf{z}_u$ through an encoder $q_{Z|X}(\mathbf{z}_u|\mathbf{X}_u)$ which serves as an approximation of the true intractable posterior. Then, the goal of maximizing the data log-likelihood is lower bounded by:

$$\begin{aligned} &\log p(X_u) \\ &\geq -D_{\text{KL}}(q_{Z|X}(\mathbf{z}_u|\mathbf{X}_u)||p(\mathbf{z}_u)) + \sum_d \mathop{\mathbb{E}}_{\sim q_{Z|X}} \left[\log p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u)\right] \end{aligned} \tag{4}$$

where D_{KL} is the Kullback-Leibler divergence between the user's encoding distribution given by the encoder and the (cross-user) prior distribution $p(\mathbf{z}_u)$. And the second term describes a reconstruction loss of user information $\mathbf{X}_u$ through the encoder $q_{Z|X}$ and decoder $p_{X|Z}$. According to [8], using a reconstruction objective like this guarantees the amount of mutual information between latent $\mathbf{z}_u$ and the observed user embeddings $\mathbf{X}_u$.

In order to avoid the direct transfer of model parameters and allow heterogeneous user embeddings, we also replace the original encoder of $q_{Z|X}(\mathbf{z}_u|\mathbf{X}_u)$ with domain-specific encoders, each of which models the corresponding posterior $q_{Z|X}(\mathbf{z}_u|\mathbf{U}_u^{(d)})$ for certain domain d. In other words, each domain now has its own encoder and decoder, and the only information that is shared across domains is the latent encoding $\mathbf{z}_u$ maintained on each user's personal space $\mathcal{E}_u$. With this model, we modify the original VAE goal

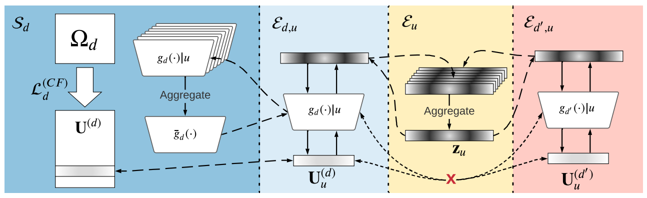

Figure 3: Model Overview. $\mathcal{S}_d$ (horizontally) aggregate user updates of g_d. The domain-specific CF prediction mechanism is kept private in each domain. $\mathcal{E}_{d,u}$ is distributed to the personal device, but cannot reach $\mathcal{E}_{d',u}$ of any other domain $d' \in \mathcal{D}^{(u)}$. User's personal space $\mathcal{E}_u$ serves as the intermediate transfer agent that learns and (vertically) aggregate user information across domains.

into a separable transfer loss with respect to domains:

$$\mathcal{L}_{\text{transfer}} = \mathcal{L}_{\text{dec}} + \mathcal{L}_{\text{enc}} \tag{5}$$
$$\mathcal{L}_{\text{dec}} = \sum_{d \in \mathcal{D}^{(u)}} \Big(\frac{1}{|\mathcal{D}^{(u)}|} D_{\text{KL}}(q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u) || p(\mathbf{z}_u)) - \mathop{\mathbb{E}}_{\sim q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u)} \Big[\log p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u)\Big] \Big)$$
$$\mathcal{L}_{\text{enc}} = \sum_{d \in \mathcal{D}^{(u)}} W_2(q_{Z|X}(\mathbf{z}_u|\mathbf{U}_u^{(d)}), q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u))$$

where $\mathcal{L}_{\text{dec}}$ comes from the original VAE loss (Eq.4), only that the encoding distribution is directly given by $q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u)$ that stored in $\mathcal{E}_u$ rather than the output of the encoder. The second loss term $\mathcal{L}_{\text{enc}}$ uses Wasserstein distance to bring close the distribution between the output of any domain-wise encoder and the $q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u)$. This allows the encoding proposed by different domains become aligned with $\mathbf{z}_u|\mathcal{E}_u$, while $\mathbf{z}_u|\mathcal{E}_u$ learns to aggregate partial information from all interacted domains and thus represents the integrated posterior $q_{X|Z}(\mathbf{z}_u|\mathbf{X}_u)$ in Eq.4.

Combining the two parts of the transfer loss Eq.(5), each personal space will store the variational information of $\mathbf{z}_u|\mathcal{E}_u$ which will serve as an intermediate checkpoint that separates the encoder loss and decoder loss. Note that the encoders only appear in $\mathcal{L}_{\text{enc}}$ and the decoders only appear in $\mathcal{L}_{\text{dec}}$. The separation of the encoder-decoder loss also separate the gradient computation between decoders and encoders as shown in figure 2-c. Consider that if it instead directly passes to the decoders with the mean of the proposed encoding ($\text{mean}_d(\mathbf{z}_u|\mathbf{U}_u^{(d)})$) rather than the stored $\mathbf{z}_u|\mathcal{E}_u$, the resulting computation would be quadratic with respect to $|\mathcal{D}^{(u)}|$. In contrast, our design only induces linear gradient computational cost (with respect to $\mathcal{D}^{(u)}$) because the gradient from decoders does not pass to encoders through $\mathbf{z}_u|\mathcal{E}_u$.

4.3 Shared Protocol of DUE

Since $\mathbf{z}_u|\mathcal{E}_u$ of users are decentralized, the framework requires a shared protocol of the $\mathbf{z}_u$'s format and its prior distribution $p(\mathbf{z}_u)$ across all users and domains to regulate the optimization. Here we adopt the simple and widely used assumption of isotropic standard Gaussian distribution. Specifically, we let $q_{X|Z}(\mathbf{z}_u|\mathcal{E}_u) \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u^2)$ where the variational information $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ are user specific and are optimized on $\mathcal{E}_u$, and the prior $p(\mathbf{z}_u) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is shared by all users. In order to apply back-propagation for encoders, we adopt the reparametrization trick for the posterior $q_{Z|X}(\mathbf{z}_u|\mathbf{U}_u^{(d)})$:

$$\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d = \text{ENC}_d(\mathbf{U}_u^{(d)})$$
$$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\mathbf{z}_u|\mathbf{U}_u^{(d)} = \boldsymbol{\mu}_d + \boldsymbol{\sigma}_d \times \epsilon$$

where the encoder ENC_d outputs the variational information (means and standard deviation) of $\mathbf{z}_u|\mathbf{U}_u^{(d)}$. Then the loss Eq.(5) breaks down into domain-wise losses:

$$\mathcal{L}_{\text{dec}}^{(d)} = -\frac{1}{2|\mathcal{D}^{(u)}|} \sum_{j=1}^{L_c} \Big(-\log \boldsymbol{\sigma}_{u,j} + \boldsymbol{\sigma}_{u,j}^2 + \boldsymbol{\mu}_{u,j}^2 - 1 \Big) - \mathop{\mathbb{E}}_{\sim q_{Z|X}(\mathbf{z}_u|\mathcal{E}_u)} \Big[\log p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u)\Big] \tag{6}$$

$$\mathcal{L}_{\text{enc}}^{(d)} = -\frac{1}{2} \sum_{j=1}^{L_c} \Big(||\boldsymbol{\mu}_{d,j} - \boldsymbol{\mu}_{u,j}||^2 + ||\boldsymbol{\sigma}_{d,j} - \boldsymbol{\sigma}_{u,j}||^2 \Big) \tag{7}$$

In terms of $p_{X|Z}$, we have assumed conditional independence between domain-specific decoders in section 4.2, which means that each domain is allowed to define its own decoder loss in Eq.(6). But for simplicity, we again define a Gaussian kernel:

$$p_{X|Z}(\mathbf{U}_u^{(d)}|\mathbf{z}_u) = \frac{1}{\sqrt{2(\pi\sigma)^{L_d}}} e^{-\frac{||\widehat{\mathbf{U}}_u^{(d)} - \mathbf{U}_u^{(d)}||^2}{2\sigma^{L_d}}} \tag{8}$$

where $\widehat{\mathbf{U}}_u^{(d)} = \text{DEC}_d(\mathbf{z}_u)$ and the variance σ is fixed as a common setting in our framework. Taking the logarithm of this formulation would simplify the reconstruction part of Eq.(6) into a square loss optimization. The details of the symbols are listed in table 1.

4.4 Federated Optimization

During training, we adopt stochastic optimization based on Eq.(6), Eq.(7), and Eq.(5) that simultaneously updates all decoders, encoders, and the DUE until convergences. Since each user typically trains on its own personal space, the derived losses are optimized user-wise and the mini-batch size will always be one. Algorithm 1 elaborates the optimization steps using federated averaging for mitigation. The updates of line 19-21 are horizontal federation steps across users within each domain, and line 16 correspond to the vertical mitigation of DUE across domains on the personal space. Figure 3 illustrates part of its mechanism in the view of a domain. Empirically, compared to centralized training, horizontal federated averaging across devices still provides convergence guarantee only with slower convergence rate [39]. And it is trivial to prove that calculating the average of $\mathbf{z}'_u|d$ on the device (line 16 in algorithm 1) is identical to the update $\mathbf{z}'_u \leftarrow \mathbf{z}_u - \alpha \frac{\partial}{\partial \mathbf{z}_u} \mathcal{L}_{\text{transfer}}$. In this paper, we assume that every personal space has an equal chance of sending updates to $\mathcal{S}_d$ (one aggregation step per gradient step) and always agree on the same learning rate, only that some devices may sometimes drop-out during a training epoch (as simulated by line 6). And here we remind the readers that there exist other mitigation strategies that deal with more complicated scenarios like heterogeneous objectives [38] and heterogeneous devices [22], which could complement our work.

4.5 Inference from DUE

For any cold-start user of a target domain d_t, we aim to do the prediction based on the transferred user embedding from auxiliary domains. Suppose that our training framework stabilizes, then the user encoding $\mathbf{z}_u|\mathcal{E}_u$ should contain information of the user across all interacted domains, and the decoder DEC_{d_t} is already well-learned collaboratively by other users in d_t. Thus, we can extract the domain-specific user information then pass it to the later prediction:

$$\hat{r}_{u,i,d_t} = f_{d_t}(\mathrm{DEC}_{d_t}(\mathbf{z}_u|\mathcal{E}_u), i) \tag{9}$$

where $\mathbf{z}_u$ is given by the personal space $\mathcal{E}_u$ rather than a calculated encoding from auxiliary domains. This allows the target domain to acquire necessary information without directly acquiring information from other domains even if they have gone offline or have been blocked. Note that our framework only trains the DECs, ENCs, and DUEs, and they are agnostic to the design of f, so the evaluation of f_{d_t} is also domain-specific. For simplicity, we assume all domains either simultaneously deploy rating prediction services or simultaneously rank their items.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets. Our main experiments uses the Amazon rating data [26] which consists user-item-rating triplets for 29 different domains. To observe the effect of number domains on the model performance, we picked 4 of its most populated domain (Books, Electronics, Home, and Clothing) and denote this subset as **Amazon-4** and denote the entire dataset as **Amazon-Full**. We filter the original data so that each domain has at least 10-core. We also include an additional MovieLens1M dataset [7] to observe how the amount of common users affects our model. We use the movie genres as item domains and preprocess the original data into 18 separate domain datasets. In this data, one item may appear in multiple domains, and we regard the same item ID that appeared in different domains as different items since we assume U-NI scenario where items are non-overlapping. We also picked 4 of its domains Comedy, Drama, Action, and Thriller with the most number of common users and denote this dataset as **ML1M-4** while the original dataset with all domains as **ML1M-Full**. And the statistical properties for these datasets are summarized in table 2.

Environment. We use the biased matrix factorization model [18] for f_d of each domain. During pretraining of these environment models, we adopt BPR loss with 2 random negative samples per record if it is the top-k recommendation task, and mean squared error loss for the rating prediction task. In the environment of the Amazon dataset, we set $L_d = 32$ for all domains, batch size = 2048. And for MovieLens dataset, we set $L_d = 8$ and batch size = 128. For both environments, we found the best learning rate around 0.001 and the L2 regularization coefficient around 0.001. The environment models are initialized by the default setting of PyTorch and optimized by Adam optimizer. We terminate the pretraining of these environment models when their performance converge on the validation set. During the training of transfer models, the only

Algorithm 1 Learning DUE with Mitigation Of Mapping (MOM)

```
1: procedure MOM
2:     Input: Pretrained $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)}$ from $\mathcal{S}_1, \ldots, \mathcal{S}_D$ respectively
3:     Initialize $\mathbf{z}_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in each $\mathcal{E}_u$.
4:     Initialize $\Theta_d$ and $\Phi_d$ on $\mathcal{S}_d$.                    ▷ $\forall d \in \mathcal{D}$
5:     while Not Converge do
6:         Sample a subset of user $\mathcal{U}_{\text{subset}} \subseteq \mathcal{U}$
7:         for $u \in \mathcal{U}_{\text{subset}}$ do
8:             Copy $\Theta_d, \Phi_d$ to apace $\mathcal{E}_{d,u}$.
9:             for $d \in \mathcal{D}^{(u)}$ do                          ▷ In space $\mathcal{E}_{d,u}$
10:                $\Theta'_d|u \leftarrow \Theta_d - \alpha \frac{\partial}{\partial \Theta_d} \mathcal{L}^{(d)}_{\text{dec}}$
11:                $\Phi'_d|u \leftarrow \Phi_d - \alpha \frac{\partial}{\partial \Phi_d} \mathcal{L}^{(d)}_{\text{enc}}$
12:                Upload proposed $\Theta'_d|u, \Phi'_d|u$ to cloud space $\mathcal{S}_d$
13:                $\mathbf{z}'_u|d \leftarrow \mathbf{z}_u - \alpha \frac{\partial}{\partial \mathbf{z}_u} \mathcal{L}^{(d)}_{\text{transfer}}$
14:                Send proposed $\mathbf{z}'_u|d$ to user space $\mathcal{E}_u$
15:            end for
16:            $\mathbf{z}_u \leftarrow \frac{1}{|\mathcal{D}^{(u)}|} \sum_{d \in \mathcal{D}^{(u)}} \mathbf{z}'_u|d$          ▷ In Space $\mathcal{E}_u$
17:        end for
18:        Mitigation of mapping in each domain:
19:        for $d \in \mathcal{D}$ do                                ▷ In each space $\mathcal{S}_d$
20:            $\Theta_d \leftarrow \frac{1}{|\mathcal{U}_{\text{subset}}|} \sum_{u \in \mathcal{U}_{\text{subset}}} \Theta'_d|u$
21:            $\Phi_d \leftarrow \frac{1}{|\mathcal{U}_{\text{subset}}|} \sum_{u \in \mathcal{U}_{\text{subset}}} \Phi'_d|u$
22:        end for
23:    end while
24: end procedure
```

training data comes from the user representations $\mathbf{X}_u$. The evaluation is conducted on a single-CPU machine in order to observe the total running time during inference.

Model and Baselines. We use algorithm 1 as the training procedure, and we consider the original VAE model and a variation that reduces to a simple auto-encoder (by ignoring all σ, σ_d, and σ_u). Denote the model as **DUE_VAE** and its auto-encoder version as **DUE_AE**. Though there is no existing work that can provide a valid solution in the Edge-CDR setting, we still include the EMCDR [24] framework to observe how effective our framework can be compared to a centralized solution. EMCDR also separates the domain-specific CF model from the transfer model learning phase but it still involves the direct transfer between domains and has to learn a mapping function for each pair of domains. We modify the model by applying user embedding averaging when multiple domains transfer their user embedding to the same target domain. We include two variants of this model based on the design of the cross-domain mapping function: **EMCDR_Linear** uses linear mapping and **EMCDR_MLP** uses multi-layer perceptron (MLP).

Evaluation Protocol: For each evaluation, we pick one target domain and hold out 10% of its users along with their records as the cold-start test set, and for each domain, we hold out another 10% of its user for validation. We consider the top-k recommendation task as the main experiment so we choose metrics including **F1@K** score and the Normalized Discounted Cumulative Gain**NDCG@K** [12] of the top-K ranked items with all items as candidates. For both metrics, we regard it as a "hit" if the recommended item in the list of size K appeared in the user history, and positive improvements of their value mean better model performance. Additionally, we also include rating prediction task and estimate how much each model reduces the **RMSE** between the predicted and the ground truth rating for cold-start users. Note that we are evaluating how the transferred user embedding works for the given f_d, instead of directly evaluating the transfer model g. We train each transfer model until its performance converges on the validation set and evaluate the model on the test set. To achieve significant estimation (with $p < 0.05$), we repeat each experiment for five rounds with different randomized splits of data and report the average results.

5.2 Recommendation for Cold-start Users

We provide the performance results of cold-start user recommendation (ranking task) in table 3 and table 4. When training DUE models, we set the user dropout rate as 30% for each epoch. For DUE frameworks, each encoder and decoder is set to an MLP with 2 layers, each with size 512 and ReLU activation (no dropout and normalization). As described in section 4.4, during training, we assume that each personal device is capable of providing one aggregation step in each training epoch, then the entire simulation is approximately a stochastic gradient descent with a batch size of one only with occasional user dropout. One approach this optimization on a GPU machine, but we observe similar speed to CPU-based training. Note that EMCDR represents a centralized model and it trains on the common users for each pair of domains, so we tune the batch size to 512 and used GPU acceleration. For all transfer models, we apply grid search and found the best learning rate around 0.0003 and an L2 regularization coefficient of 0.001. During inference, EMCDR models need to average the transferred embeddings from each auxiliary domain like the mitigation step of MOM (line 16 of algorithm 1). For each experiment, we include the option **No_Transfer** which only uses the randomly initialized embedding for the cold-start user as an additional baseline to show the improvement induced by transfer learning.

In all experiments, transfer learning models (both DUE-based models and EMCDR-based models) effectively improve F1 score and NDCG of recommendation over the "No_Transfer" baseline. Compared with EMCDR models, our proposed framework (both DUE_AE and DUE_VAE) show superior performance. This improvement of recommendation performance is consistent over different list sizes as shown in table 3, and is also consistent over different choices of target domains as shown in table 4. And in most cases, we observe slightly better performance from DUE_VAE compared to DUE_AE which ignores variation, but VAE-based DUE models usually exhibit less stable behavior.

Note that EMCDR still requires a direct transfer between domains, and we have found no existing solutions that accommodate Edge-CDR setting as mentioned in section 2. In contrast, the DUE framework can effectively solve the Edge-CDR problem without violating the restrictions. Rather than directly probing other domains, each domain only interact with the public personal space $\mathcal{E}_u$ which only maintains encoded information of the user, and the only information it has to offer is its updates for this encoding (line 13 of algorithm 1). As illustrated in figure 3, both the raw user information $\mathbf{U}_u^{(d)}$ and its auto-encoder g_d are private to the domain d and cannot be accessed by outside services even if it is on the personal device. In the most extreme cases where the gradient of $\mathbf{z}_u|d$ is exposed to privacy attacks and a highly risk-sensitive domain service still wants to protect its user information, the domain can simply take away those sensitive features from $\mathbf{U}_u^{(d)}$ to avoid these attacks, otherwise, it has to apply privacy-preserving techniques[35, 43].

5.3 Dealing with Heterogeneity

As explained in section 3.3, users may have different representation format across domains, and may have different interaction domain subset $\mathcal{D}^{(u)}$. Though in the previous section we have shown results for independent domain-specific user embeddings, we would like to observe how the transfer models behave differently when the setting becomes more heterogeneous. Specifically, we follow the same experimental procedure as in the previous section but manually assign different domain embedding size L_d for domains. Details of the environment setting are provided in table 5 and the corresponding results can be seen in table 5. For both Amazon-4(d_t: Books) and Amazon-4(d_t:Electronics) data, with heterogeneous L_d, all transfer models appear to be more effective compared to that in table 3 and table4 where L_d is the same across domains. This indicates that the heterogeneous embedding sizes, if well-picked, may naturally improve each domain's f_d and user representations no matter the choice transfer model. However, EMCDR becomes less stable and less effective on the Amazon-Full dataset when using heterogeneous embedding sizes, as shown in table 5. In contrast, DUE models continue to prove their effectiveness and achieve similar or better results than its counterpart in table 3. We argue that the simplified view of each domain potentially reduces the chance of overfitting, especially when the number of domains involved is large. Each domain solely needs to deal with its auto-encoder and needs no aggregation of transferred information from all domains.

5.4 Dataset Characteristics

In order to observe how dataset characteristics affect the models' performances, we include the MovieLens dataset where domains have much larger proportion of common users as illustrated in table 2. We also include an additional rating prediction task with RMSE as evaluation metrics in order to observe the effect of different data types. We summarize the results in Table 6. Except for a slight improvement of RMSE in the ML1M-4 dataset, the overall recommendation performances (F1 and NDCG) of the DUE framework and EMCDR baselines are usually indistinguishable on ML1M-4 and ML1M-Full, unlike that in Amazon datasets. We argue that this is related to the fact that pairwise transfer model like EMCDR trains on the common user between domains. As shown in table 2, compared to Amazon datasets, the probabilities of sharing common users between a random pair of domains in both ML1M-4 and ML1M-Full datasets are over 70% with respect to the total number of user, which are much higher than that in Amazon dataset. This indicates that the EMCDR can perform well when they can learn from the majority of the users and they are expected to provide better generalization performances than it is trained with a smaller portion of users. However, in a realistic Edge-CDR setting, datasets

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}_d\rvert$	$\lvert\Omega_d\rvert$	avg(sparsity)	$\overline{P}$(shared user between d_i,d_j)	$\overline{P}$(unique users)
Amazon-4	276,120	4	24,343 - 31,453 - 33,610	1,520,760 - 1,573,695 - 3,428,087	0.0644%	22,72%	19.77%
Amazon-Full	366,710	29	321 - 12,137 - 33,610	7,057 - 490,070 - 3,428,087	0.1176%	6.23%	2.76%
ML1M-4	2,871	4	380 - 597 - 815	150,793 - 237,067 - 279,122	14.84%	88.50%	0.54%
ML1M-Full	3,082	18	40 - 177 - 815	6,166 - 58,588 - 279,122	14.54%	72.05%	0.15%

Table 2: Dataset Summary. (min-median-max) values are provided for $\lvert\mathcal{I}_d\rvert$ and $\lvert\Omega_d\rvert$. Last two columns gives the average probability of shared user between a pair of domain and the average probability of a user is unique in a domain.

Models	Amazon-4(d_t:Books)							Amazon-Full(d_t:Books)						
	F1			NDCG			Inference	F1			NDCG			Inference
	@1	@5	@10	@1	@5	@10	time(s)	@1	@5	@10	@1	@5	@10	time(s)
No_Transfer	0.0010	0.0041	0.0049	0.0048	0.0074	0.0085	—	**0.0011**	0.0040	0.0050	0.0079	0.0099	0.0101	—
EMCDR_Linear	0.0015	0.0057	0.0074	0.0099	0.0113	0.0126	4254	0.0010	0.0052	0.0069	0.0094	0.0120	0.0127	31814
EMCDR_MLP	0.0018	0.0057	0.0076	0.0099	0.0111	0.0123	4460	**0.0011**	0.0050	0.0071	0.0094	0.0117	0.0127	37233
DUE_AE	**0.0025**	**0.0060**	**0.0082**	**0.0118**	**0.0123**	**0.0141**	2122	**0.0011**	0.0064	0.0089	0.0092	0.0146	0.0155	2712
DUE_VAE	0.0024	0.0057	0.0078	0.0115	0.0117	0.0133	2230	0.0010	**0.0067**	**0.0094**	**0.0094**	**0.0150**	**0.0160**	3121

Table 3: Cold-start user recommendation performance on Amazon datasets.

Models	Amazon-4(d_t:Books)				Amazon-4(d_t:Electronics)				Amazon-4(d_t:Home)				Amazon-4(d_t:Clothing)			
	F1		NDCG		F1		NDCG		F1		NDCG		F1		NDCG	
	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10
No_Transfer	0.0010	0.0049	0.0048	0.0085	0.0015	0.0087	0.0145	0.0143	0.0006	0.0037	0.0040	0.0059	0.0006	0.0028	0.0029	0.0042
EMCDR_Linear	0.0015	0.0074	0.0099	0.0126	0.0024	0.0115	0.0223	0.0192	0.0015	0.0054	0.0093	0.0093	0.0006	0.0041	0.0035	0.0061
EMCDR_MLP	0.0018	0.0076	0.0099	0.0123	0.0024	0.0118	0.0227	0.0198	**0.0016**	0.0053	0.0094	0.0093	0.0006	0.0041	0.0035	0.0060
DUE_AE	**0.0025**	**0.0082**	**0.0118**	**0.0141**	**0.0029**	0.0131	0.0314	0.0223	**0.0016**	**0.0056**	**0.0099**	0.0100	0.0013	**0.0050**	**0.0061**	**0.0075**
DUE_VAE	0.0024	0.0078	0.0115	0.0133	**0.0029**	**0.0138**	**0.0315**	**0.0233**	**0.0016**	**0.0056**	0.0097	**0.0102**	**0.0016**	0.0045	**0.0061**	0.0074

Table 4: Cold-start user recommendation performance on different target domains.

Models	Amazon-4(d_t:Books)				Amazon-4(d_t:Electronics)				Amazon-Full(d_t:Book)			
	F1		NDCG		F1		NDCG		F1		NDCG	
	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10	@1	@10
No_Transfer	0.0007	0.0051	0.0038	0.0079	0.0009	0.0085	0.0077	0.0129	0.0010	0.0057	0.0077	0.0103
EMCDR_Linear	0.0016	0.0080	0.0087	0.0130	0.0026	0.0132	0.0247	0.0215	0.0011	0.0065	0.0077	0.0116
EMCDR_MLP	0.0016	**0.0082**	0.0086	0.0132	0.0025*	0.0130*	0.0237*	0.0214*	0.0010*	0.0066*	0.0077*	0.0118*
DUE_AE	**0.0036***	0.0077*	**0.0182***	**0.0150***	0.0028*	**0.0140***	**0.0314***	**0.0240***	**0.0016***	0.0088*	0.0142*	0.0162*
DUE_VAE	0.0026	0.0080	0.0134	0.0147	**0.0029**	0.0138	**0.0314**	0.0233	0.0015*	**0.0090***	**0.0146***	**0.0171***

Table 5: Recommendation performance with heterogeneous user embedding size. Entries with "*" indicates direct result of one round instead of an averaged result.

have a similar taste to Amazon where the common user sharing between a pair of domains is rather rare (23% in Amazon-4 and 6% in Amazon-Full). In such scenarios, common users with other domains are less representative of the remaining majority of users. And compared to the Amazon-4 dataset, we observe an even larger performance gap from EMCDR to DUE on Amazon-Full, as the latter has an even smaller sharing probability of user. Differently, in the DUE framework, each domain-specific transfer model learns its mapping models on the intermediate $\mathbf{z}_u$, which exists as long as the user has interacted with at least one auxiliary domain. In other words, the only users that are outside of the transfer learning are those who interact with only a single domain. As shown in table 2, the average probability of a user to be unique to a specific domain is always below 20% in all datasets. This means that for all datasets, each domain-specific transfer model in the DUE framework trains with more than 80% of the domain's users.

Besides, we observe that the number of domains used in the training also affects the performance in multiple ways. Specifically, compared to results on ML1M-4 and Amazon-4, using the full set with all the domains consistently generate better NDCG values but sometimes exhibit worse F1 score (as shown in table 3, table 5, and table 6). This phenomenon is observed in both MovieLens data (from ML1M-4 to ML1M-Full) and Amazon data (from Amazon-4 to Amazon-Full). Thus, we provide a detailed view for different recommendation list size in Table 3, where we find a different trend for the top-1 recommendation: the first item usually performs worse when models are trained on the full domain dataset instead of the domain subset. This may due to the fact that, when more domains are involved in the training, transfer models will mitigate more transferred embeddings for each user, thus become averaged towards a better cluster of ranked items and less likely to pin-point the best item. In the case of rating prediction, the RMSE of all models (including the No_Transfer baseline) on the full set of domains is usually worse than that on the subset (for both MovieLens and Amazon data), possibly an indicator of a reduced regression accuracy for predicting user ratings because of the noise introduced by extra auxiliary domains.

Models	ML1M-4(d_t:Action)			ML1M-Full(d_t:Action)			Amazon-4(d_t:Books)			Amazon-Full(d_t:Books)		
	F1@5	NDCG@5	RMSE	F1@5	NDCG@5	RMSE	F1@5	NDCG@5	RMSE	F1@5	NDCG@5	RMSE
No_Transfer	0.0932	0.6240	0.9619	0.0894	0.6560	1.0025	0.0041	0.0074	0.8543	0.0040	0.0099	0.8617
EMCDR_Linear	0.0951	0.6524	0.9493	0.0924	0.6710	0.9792	0.0057	0.0113	0.8430	0.0052	0.0120	0.8430
EMCDR_MLP	**0.0975**	**0.6572**	0.9458	0.0931	0.6759	0.9789	0.0057	0.0111	0.8413	0.0050	0.0117	0.8426
MoM_AE	0.0955	0.6497	**0.9437**	0.0935	0.6735	**0.9796**	0.0057	0.0123	0.8413	0.0064	0.0146	0.8420
MoM_VAE	0.0957	0.6521	0.9442	**0.0941**	**0.6793**	0.9791	**0.0060**	**0.0132**	**0.8405**	**0.0067**	**0.0150**	**0.8419**

Table 6: Performance on both Amazon and MovieLens data with both ranking and rating prediction tasks.

d	$\|\mathcal{U}_d\|$	$\|\mathcal{I}_d\|$	$\|\Omega_d\|$	L_d
Beauty	4,160	321	6,491	8
Magazine	4,336	346	7,261	8
Gift_Cards	7,001	337	10,176	8
Fashion	9,641	783	13,669	8
Appliances	22,947	1,612	31,672	8
Software	19,776	1,358	36,018	8
Luxury	28,073	2,561	64,697	8
Music	19,003	4,229	82,499	8
Prime_Pantry	17,379	3,647	102,773	16
Instruments	34,637	6,408	158,864	16
Industrial	93,537	7,709	187,970	16
Arts	53,485	8,475	226,155	16
Games	76,690	9,851	337,293	16
CDs	40,150	10,935	399,296	16
Toys	85,088	16,355	451,197	16
Automotive	129,625	12,137	473,938	32
Garden	146,962	16,615	516,752	32
Grocery	103,223	13,537	518,263	32
Phones	76,297	27,992	601,192	32
Office	173,377	16,509	615,947	32
Kindle	53,628	17,062	722,651	32
Tools	166,843	19,760	795,454	32
Pet	123,513	14,511	826,764	32
Sports	122,283	28,192	883,248	32
Movies	68,299	17,764	1,214,837	48
Home	169,704	31,257	1,387,462	48
Clothing	208,654	24,343	1,441,081	48
Electronics	92002	31,648	1,486,594	48
Books	71,104	33,610	3,332,789	48

Table 7: Embedding size setting for heterogeneous test.

5.5 Efficient Star-Shape Transfer

For models like EMCDR that learns a mapping for each pair of domains, it will result in an undesirable quadratic $D(D-1)$ cost with respect to the number of domains. In contrast, our DUE framework reduces the number of required mapping functions to $2D$ (D encoders and D decoders) and forms a star-like transfer scheme as shown in Figure 4. In terms of complexity, this means reduced storage, reduced computation, and reduced communication overheads: each domain only needs to maintain a mapping of size $O(L_d L_c)$ (or $O(L_d H + H^2 + HL_c)$ if using MLP where H is the hidden layer size), while domain-pairwise transfer model like EMCDR has to keep mappings for each of the other domains inducing a cost in the order of $O(L_d^2 D)$ (or $O(L_d HD + H^2 D)$ if MLP). Though L_c is usually larger than L_d, we typically set it to be much smaller than $L_d D$ which means that the cost of the DUE-based model is much smaller than

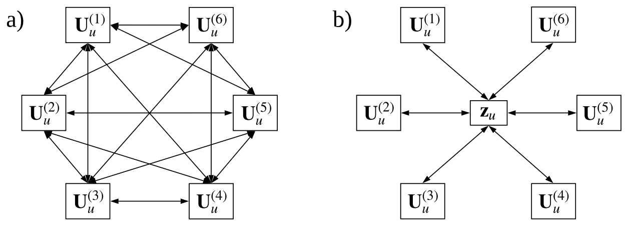

Figure 4: Transfer schema of 6 domains. a) domain-pairwise transfer; b) Using DUE as intermediate representation

that of a domain-pairwise transfer model. We list the inference time of the transfer models in table 3, and the corresponding DUE size $L_c = 64$ and domain-specific embedding size $L_d = 32$. Compared to the domain pairwise model EMCDR whose running time depends on the number of domains involved, the inference time of the DUE model is almost the same level no matter the size of $\mathcal{D}$. In the case of mobile phones, users typically interact with lots of services (e.g. 10 mobile apps every day, 30 every month[1]) and the number of domains is even larger considering the potential sub-domains in each service. One would observe an even larger efficiency boost of DUE in such cases. Note that each domain has to communicate the transfer model g_d in addition to its prediction model, but it is a necessary cost for transfer learning and the cost is much smaller than domain-pairwise models as previously discussed.

6 CONCLUSION AND DISCUSSION

In this paper, we present the Edge-CDR as a federated collaborative learning task, which restricts the direct transfer of user embeddings and assumes heterogeneous user representations across domains. By maintaining a decentralized user embedding on each user personal space and optimize them with a federated VAE, the proposed solution can effectively solve cross-domain rating prediction and recommendation for cold-start users, and it is more effective and efficient than domain-pairwise transfer models when users typically interact with a large number of domains. A major point for future work is to figure out the standard representation format of DUE since it requires agreement from all domains and users. And the problem may become even more challenging when user content and environmental context information are encoded on the personal space, or when heterogeneous objectives (e.g. between rating and ranking) and heterogeneous devices are taken into consideration.

[1]http://files.appannie.com.s3.amazonaws.com/reports/1705_Report_Consumer_App_Usage_EN.pdf

REFERENCES

[1] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.

[2] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. 2011. " You Might Also Like:" Privacy Risks of Collaborative Filtering. In *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 231–246.

[3] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning for Recommendation. *ArXiv* abs/1802.07876 (2018).

[4] Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. 2021. Neural Collaborative Reasoning. *WWW* (2021). https://arxiv.org/abs/2005.08129

[5] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain Recommendation Without Sharing User-relevant Data. In *The World Wide Web Conference*. 491–502.

[6] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

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

[8] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bklr3j0cKX

[9] 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*. 667–676.

[10] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. MTNet: a neural approach for cross-domain recommendation with unstructured text. *KDD Deep Learning Day* (2018), 1–10.

[11] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*. 595–606.

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[13] Yaru Jin, Shoubin Dong, Yong Cai, and Jinlong Hu. 2020. RACRec: Review Aware Cross-Domain Recommendation for Fully-Cold-Start User. *IEEE Access* 8 (2020), 55032–55041.

[14] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1563–1572.

[15] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross domain recommender systems: a systematic literature review. *ACM Computing Surveys (CSUR)* 50, 3 (2017), 1–34.

[16] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2014).

[17] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*. https://arxiv.org/abs/1610.05492

[18] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

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

[20] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 331–339. https://doi.org/10.1145/3336191.3371793

[21] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2019. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873* (2019).

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).

[23] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang. 2020. A Secure Federated Transfer Learning Framework. *IEEE Intelligent Systems* 35, 4 (2020), 70–82.

[24] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*. 2464–2470.

[25] Christian Meurisch, Bekir Bayrak, and Max Mühlhäuser. 2020. Privacy-preserving AI Services Through Data Decentralization. In *Proceedings of The Web Conference 2020*. 190–200.

[26] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.

[27] Petar Popovski, Jimmy J Nielsen, Cedomir Stefanovic, Elisabeth De Carvalho, Erik Strom, Kasper F Trillingsgaard, Alexandru-Sabin Bana, Dong Min Kim, Radoslaw Kotaba, Jihong Park, et al. 2018. Wireless access for ultra-reliable low-latency communication: Principles and building blocks. *Ieee Network* 32, 2 (2018), 16–23.

[28] Dimitrios Rafailidis and Fabio Crestani. 2019. Neural Attentive Cross-Domain Recommendation. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 165–172.

[29] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. 2002. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*. 127–134.

[30] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[31] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* (2019).

[32] Jiarui Shi and Quanmin Wang. 2019. Cross-Domain Variational Autoencoder for Recommender Systems. In *2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT)*. IEEE, 67–72.

[33] Shaoyun Shi, Hanxiong Chen, Weizhi Ma, Jiaxin Mao, Min Zhang, and Yongfeng Zhang. 2020. Neural Logic Reasoning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1365–1374.

[34] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE internet of things journal* 3, 5 (2016), 637–646.

[35] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1310–1321.

[36] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 650–658.

[37] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*. 4424–4434.

[38] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *NeurIPS* (2020).

[39] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2018. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 63–71.

[40] Xinghua Wang, Zhaohui Peng, Senzhang Wang, S Yu Philip, Wenjing Fu, and Xiaoguang Hong. 2018. Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In *International Conference on Database Systems for Advanced Applications*. Springer, 158–165.

[41] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. 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, 285–294. https://doi.org/10.1145/3331184.3331203

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[43] Dayin Zhang, Xiaojun Chen, Dakui Wang, and Jinqiao Shi. 2018. A Survey on Collaborative Deep Learning and Privacy-Preserving. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 652–658.